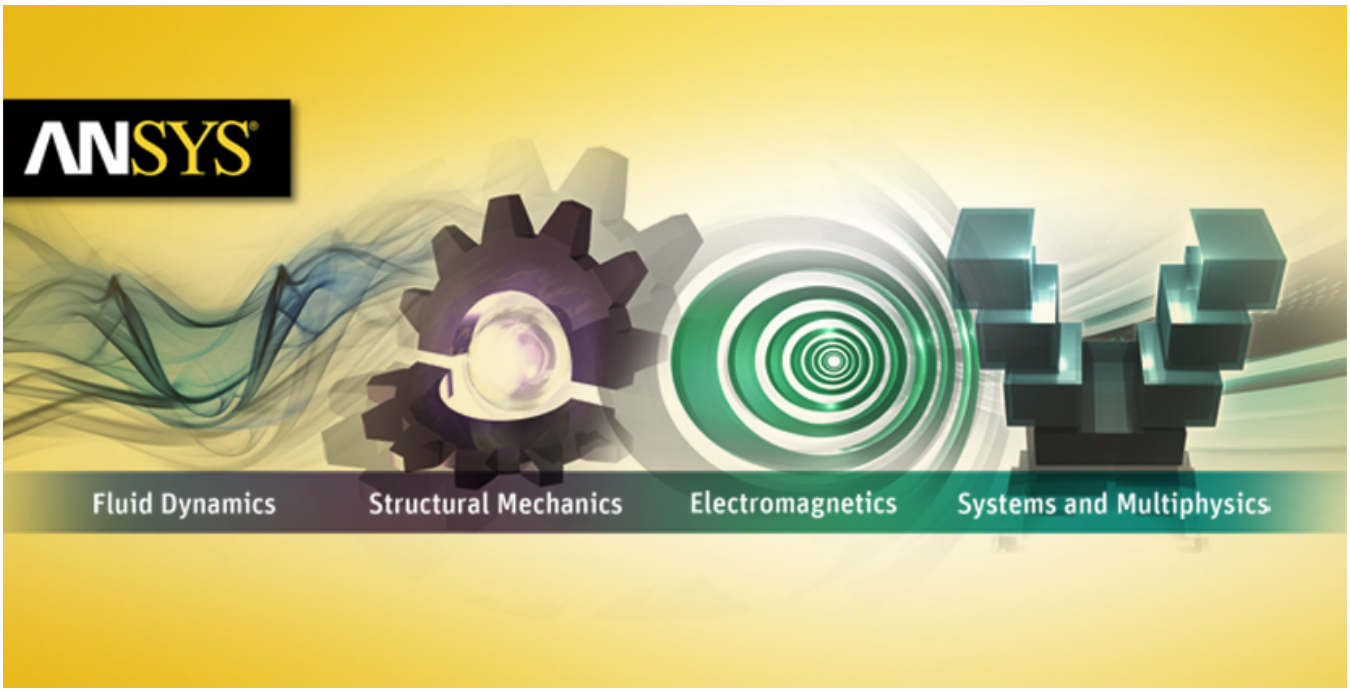


ANSYS Mechanical APDL Theory Reference

ANSYS, Inc.
Southpointe
275 Technology Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
http://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 15.0
November 2013

ANSYS, Inc. is certified to ISO 9001:2008.

Published in the U.S.A.

Edited by: Peter Kohnke, Ph.D.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Welcome to the *Mechanical APDL Theory Reference*. The reference presents theoretical descriptions of all elements, as well as of many procedures and commands used in these products. It is available to any of our product users who need to understand how the program uses input data to calculate the output, and is an indispensable tool to help you interpret various element and command results. The *Mechanical APDL Theory Reference* describes the relationship between input data and output results produced by the programs, and is essential for a thorough understanding of how the programs function.

The following introductory topics are available:

1.1. Purpose of the Theory Reference

The purpose of the *Mechanical APDL Theory Reference* is to inform you of the theoretical basis of these products. By understanding the underlying theory, you can use these products more intelligently and with greater confidence, making better use of their capabilities while being aware of their limitations. Of course, you are not expected to study the entire volume; you need only to refer to sections of it as required for specific elements and procedures. This manual does not, and cannot, present all theory relating to finite element analysis. If you need the theory behind the basic finite element method, you should obtain one of the many references available on the topic. If you need theory or information that goes beyond that presented here, you should (as applicable) consult the indicated reference, run a simple test problem to try the feature of interest, or contact your ANSYS Support Distributor for more information.

The theory behind the basic analysis disciplines is presented in Structures (p. 5) through Coupling (p. 293). Structures (p. 5) covers structural theory, with Structures with Geometric Nonlinearities (p. 29) and Structures with Material Nonlinearities (p. 63) adding geometric and structural material nonlinearities. Electromagnetics (p. 177) discusses electromagnetics, Heat Flow (p. 227) deals with heat flow, Fluid Flow (p. 245) handles fluid flow, and Acoustics (p. 253) deals with acoustics. Coupled effects are treated in Coupling (p. 293).

Element theory is examined in Shape Functions (p. 327), Element Tools (p. 365), and Element Library (p. 411). Shape functions are presented in Shape Functions (p. 327), information about element tools (integration point locations, matrix information, and other topics) is discussed in Element Tools (p. 365), and theoretical details of each ANSYS element are presented in Element Library (p. 411).

Analysis Tools (p. 665) examines a number of analysis tools (acceleration effect, damping, element reordering, and many other features). Analysis Procedures (p. 761) discusses the theory behind the different analysis types used in the ANSYS program.

Numerical processors used in preprocessing and postprocessing are covered in Preprocessing and Postprocessing Tools (p. 831). Postprocessing (p. 843) goes into a number of features from the general postprocessor (POST1) and the time-history postprocessor (POST26).

An index of keywords and commands has been compiled to give you handy access to the topic or command of interest.

1.2. Understanding Theory Reference Notation

The notation defined below is a partial list of the notation used throughout the manual. There are also some tables of definitions given in following sections:

- Coupling (p. 293)
- Rate-Independent Plasticity (p. 64)

Due to the wide variety of topics covered in this manual, some exceptions will exist.

Table 1.1: General Terms

Term	Meaning
[B]	strain-displacement matrix
[C]	damping matrix
[Ct]	specific heat matrix
[D]	elasticity matrix
E	Young's modulus
{F}	force vector
[I]	identity matrix
{I}	current vector, associated with electrical potential degrees of freedom
{J}	current vector, associated with magnetic potential degrees of freedom
[K]	stiffness matrix
$[K^t]$	conductivity matrix
[M]	mass matrix
[O]	null matrix
P, {P}	pressure (vector)
{Q}	heat flow vector
[S]	stress stiffness matrix
{T}	temperature vector
t	time, thickness
$[T_R]$	local to global conversion matrix
u, v, w, {u}	displacement, displacement vector
{V}	electric potential vector
δU	virtual internal work
δV	virtual external work
{W}	fluid flow vector
x, y, z	element coordinate
X, Y, Z	nodal coordinates (usually global Cartesian)

Term	Meaning
α	coefficient of thermal expansion
ε	strain
ν	Poisson's ratio
σ	stress

Below is a partial list of superscripts and subscripts used on [K], [M], [C], [S], {u}, {T}, and/or {F}. See also Coupling (p. 293). The absence of a subscript on the above terms implies the total matrix in final form, ready for solution.

Table 1.2: Superscripts and Subscripts

Term	Meaning
ac	nodal effects caused by an acceleration field
c	convection surface
cr	creep
e	based on element in global coordinates
el	elastic
g	internal heat generation
i	equilibrium iteration number
ℓ	based on element in element coordinates
m	master
n	substep number (time step)
nd	effects applied directly to node
pl	plasticity
pr	pressure
s	slave
sw	swelling
t, th	thermal
^	(flex over term) reduced matrices and vectors
.	(dot over term) time derivative

1.3. Applicable Products

This manual applies to the following ANSYS and ANSYS Workbench products:

ANSYS Multiphysics
ANSYS Mechanical
ANSYS Structural
ANSYS Mechanical EMAG
ANSYS Professional - Nonlinear Structural
ANSYS Professional - Nonlinear Thermal
ANSYS Mechanical PrepPost
ANSYS DesignSpace (ANSYS Workbench only)

Some command arguments and element KEYOPT settings have defaults in the derived products that are different from those in the full ANSYS product. These cases are clearly documented under the "Product Restrictions" section of the affected commands and elements. If you plan to use your derived product input file in the ANSYS Multiphysics product, you should explicitly input these settings in the derived product, rather than letting them default; otherwise, behavior in the full ANSYS product will be different.

Chapter 2: Structures

The following topics are available for structures:

2.1. Structural Fundamentals

The following topics concerning structural fundamentals are available:

2.1.1. Stress-Strain Relationships

This section discusses material relationships for linear materials. Nonlinear materials are discussed in Structures with Material Nonlinearities (p. 63). The stress is related to the strains by:

$$\{\sigma\} = [D]\{\varepsilon^{el}\} \tag{2.1}$$

where:

$\{\sigma\}$ = stress vector = $\left\lfloor \sigma_x \ \sigma_y \ \sigma_z \ \sigma_{xy} \ \sigma_{yz} \ \sigma_{xz} \right\rfloor^T$ (output as S)
[D] = elasticity or elastic stiffness matrix or stress-strain matrix (defined in Equation 2.14 (p. 8) through Equation 2.19 (p. 8)) or inverse defined in Equation 2.4 (p. 6) or, for a few anisotropic elements, defined by full matrix definition (input with **TB**,ANEL.)
$\{\varepsilon^{el}\} = \{\varepsilon\} - \{\varepsilon^{th}\}$ = elastic strain vector (output as EPEL)
$\{\varepsilon\}$ = total strain vector = $\left\lfloor \varepsilon_x \ \varepsilon_y \ \varepsilon_z \ \varepsilon_{xy} \ \varepsilon_{yz} \ \varepsilon_{xz} \right\rfloor^T$
$\{\varepsilon^{th}\}$ = thermal strain vector (defined in Equation 2.3 (p. 6)) (output as EPTH)

Note

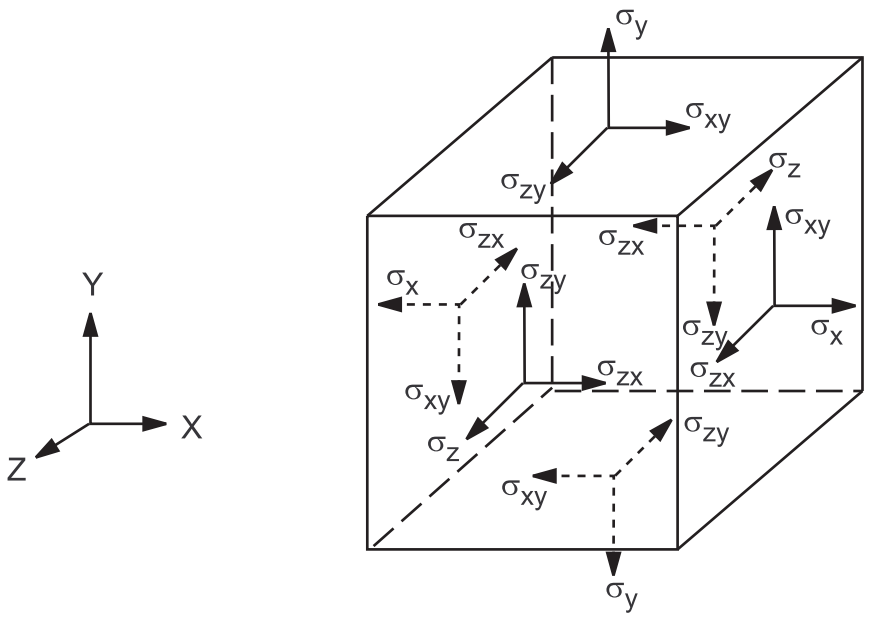
$\{\varepsilon^{el}\}$ (output as EPEL) are the strains that cause stresses.

The shear strains (ε_{xy}, ε_{yz}, and ε_{xz}) are the engineering shear strains, which are twice the tensor shear strains. The ε notation is commonly used for tensor shear strains, but is used here as engineering shear strains for simplicity of output.

A related quantity used in POST1 labeled "component total strain" (output as EPTO) is described in Structures with Material Nonlinearities (p. 63).

The stress vector is shown in the figure below. The sign convention for direct stresses and strains used throughout the ANSYS program is that tension is positive and compression is negative. For shears, positive is when the two applicable positive axes rotate toward each other.

Figure 2.1: Stress Vector Definition



Equation 2.1 (p. 5) may also be inverted to:

$$\{\varepsilon\} = \{\varepsilon^{th}\} + [D]^{-1}\{\sigma\} \tag{2.2}$$

For the 3-D case, the thermal strain vector is:

$$\{\varepsilon^{th}\} = \Delta T \begin{bmatrix} \alpha_x^{se} & \alpha_y^{se} & \alpha_z^{se} & 0 & 0 & 0 \end{bmatrix}^T \tag{2.3}$$

where:

α_x^{se} = secant coefficient of thermal expansion in the x direction (see Temperature-Dependent Coefficient of Thermal Expansion (p. 10))
$\Delta T = T - T_{ref}$
T = current temperature at the point in question
T_{ref} = reference (strain-free) temperature (input on **TREF** command or as REFT on **MP** command)

The flexibility or compliance matrix, $[D]^{-1}$ is:

$$[D]^{-1} = \begin{bmatrix} 1/E_x & -\nu_{xy}/E_x & -\nu_{xz}/E_x & 0 & 0 & 0 \\ -\nu_{yx}/E_y & 1/E_y & -\nu_{yz}/E_y & 0 & 0 & 0 \\ -\nu_{zx}/E_z & -\nu_{zy}/E_z & 1/E_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{xy} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{yz} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{xz} \end{bmatrix} \tag{2.4}$$

where typical terms are:

E_x = Young's modulus in the x direction (input as EX on **MP** command)
ν_{xy} = major Poisson's ratio (input as PRXY on **MP** command)
ν_{yx} = minor Poisson's ratio (input as NUXY on **MP** command)
G_{xy} = shear modulus in the xy plane (input as GXY on **MP** command)

The difference between ν_{xy} and ν_{yx} is described below.

Also, the $[D]^{-1}$ matrix is presumed to be symmetric, so that:

$$\frac{\nu_{yx}}{E_y} = \frac{\nu_{xy}}{E_x} \tag{2.5}$$

$$\frac{\nu_{zx}}{E_z} = \frac{\nu_{xz}}{E_x} \tag{2.6}$$

$$\frac{\nu_{zy}}{E_z} = \frac{\nu_{yz}}{E_y} \tag{2.7}$$

Because of the above three relationships, ν_{xy}, ν_{yz}, ν_{xz}, ν_{yx}, ν_{zy}, and ν_{zx} are not independent quantities and therefore the user should input either ν_{xy}, ν_{yz}, and ν_{xz} (input as PRXY, PRYZ, and PRXZ), or ν_{yx}, ν_{zy}, and ν_{zx} (input as NUXY, NUYZ, and NUXZ). The use of Poisson's ratios for orthotropic materials sometimes causes confusion, so that care should be taken in their use. Assuming that E_x is larger than E_y, ν_{xy} (PRXY) is larger than ν_{yx} (NUXY). Hence, ν_{xy} is commonly referred to as the "major Poisson's ratio", because it is larger than ν_{yx}, which is commonly referred to as the "minor" Poisson's ratio. For orthotropic materials, the user needs to inquire of the source of the material property data as to which type of input is appropriate. In practice, orthotropic material data are most often supplied in the major (PR-notation) form. For isotropic materials ($E_x = E_y = E_z$ and $\nu_{xy} = \nu_{yz} = \nu_{xz}$), so it makes no difference which type of input is used.

Expanding Equation 2.2 (p. 6) with Equation 2.3 (p. 6) through Equation 2.7 (p. 7) and writing out the six equations explicitly,

$$\varepsilon_x = \alpha_x \Delta T + \frac{\sigma_x}{E_x} - \frac{\nu_{xy}\sigma_y}{E_x} - \frac{\nu_{xz}\sigma_z}{E_x} \tag{2.8}$$

$$\varepsilon_y = \alpha_y \Delta T - \frac{\nu_{xy}\sigma_x}{E_x} + \frac{\sigma_y}{E_y} - \frac{\nu_{yz}\sigma_z}{E_y} \tag{2.9}$$

$$\varepsilon_z = \alpha_z \Delta T - \frac{\nu_{xz}\sigma_x}{E_x} - \frac{\nu_{yz}\sigma_y}{E_y} + \frac{\sigma_z}{E_z} \tag{2.10}$$

$$\varepsilon_{xy} = \frac{\sigma_{xy}}{G_{xy}} \tag{2.11}$$

$$\varepsilon_{yz} = \frac{\sigma_{yz}}{G_{yz}} \tag{2.12}$$

$$\varepsilon_{xz} = \frac{\sigma_{xz}}{G_{xz}} \tag{2.13}$$

where typical terms are:

ε_x = direct strain in the x direction
σ_x = direct stress in the x direction
ε_{xy} = shear strain in the x-y plane
σ_{xy} = shear stress on the x-y plane

Alternatively, Equation 2.1 (p. 5) may be expanded by first inverting Equation 2.4 (p. 6) and then combining that result with Equation 2.3 (p. 6) and Equation 2.5 (p. 7) through Equation 2.7 (p. 7) to give six explicit equations:

$$\sigma_x = \frac{E_x}{h}\left(1-(\nu_{yz})^2\frac{E_z}{E_y}\right)(\varepsilon_x-\alpha_x\Delta T)+\frac{E_y}{h}\left(\nu_{xy}+\nu_{xz}\nu_{yz}\frac{E_z}{E_y}\right)(\varepsilon_y-\alpha_y\Delta T)+ \frac{E_z}{h}(\nu_{xz}+\nu_{yz}\nu_{xy})(\varepsilon_z-\alpha_z\Delta T) \tag{2.14}$$

$$\sigma_y = \frac{E_y}{h}\left(\nu_{xy}+\nu_{xz}\nu_{yz}\frac{E_z}{E_y}\right)(\varepsilon_x-\alpha_x\Delta T)+\frac{E_y}{h}\left(1-(\nu_{xz})^2\frac{E_z}{E_x}\right)(\varepsilon_y-\alpha_y\Delta T)+ \frac{E_z}{h}\left(\nu_{yz}+\nu_{xz}\nu_{xy}\frac{E_y}{E_x}\right)(\varepsilon_z-\alpha_z\Delta T) \tag{2.15}$$

$$\sigma_z = \frac{E_z}{h}(\nu_{xz}+\nu_{yz}\nu_{xy})(\varepsilon_x-\alpha_x\Delta T)+\frac{E_z}{h}\left(\nu_{yz}+\nu_{xz}\nu_{xy}\frac{E_y}{E_x}\right)(\varepsilon_y-\alpha_y\Delta T)+ \frac{E_z}{h}\left(1-(\nu_{xy})^2\frac{E_y}{E_x}\right)(\varepsilon_z-\alpha_z\Delta T) \tag{2.16}$$

$$\sigma_{xy} = G_{xy}\varepsilon_{xy} \tag{2.17}$$

$$\sigma_{yz} = G_{yz}\varepsilon_{yz} \tag{2.18}$$

$$\sigma_{xz} = G_{xz}\varepsilon_{xz} \tag{2.19}$$

where:

$$h = 1-(\nu_{xy})^2\frac{E_y}{E_x}-(\nu_{yz})^2\frac{E_z}{E_y}-(\nu_{xz})^2\frac{E_z}{E_x}-2\nu_{xy}\nu_{yz}\nu_{xz}\frac{E_z}{E_x} \tag{2.20}$$

If the shear moduli G_{xy}, G_{yz}, and G_{xz} are not input for isotropic materials, they are computed as:

$$G_{xy} = G_{yz} = G_{xz} = \frac{E_x}{2(1+\nu_{xy})} \tag{2.21}$$

For orthotropic materials, the user needs to inquire of the source of the material property data as to the correct values of the shear moduli, as there are no defaults provided by the program.

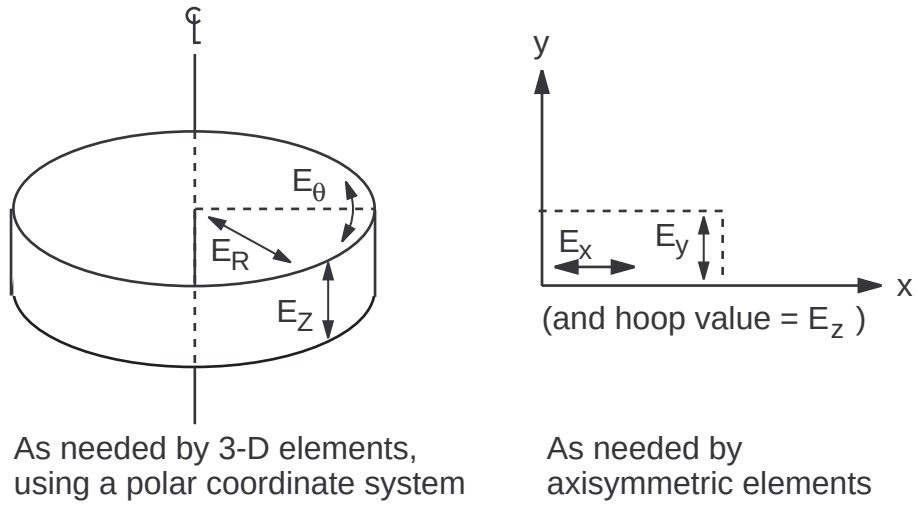
The [D] matrix must be positive definite. The program checks each material property as used by each active element type to ensure that [D] is indeed positive definite. Positive definite matrices are defined in Positive Definite Matrices (p. 390). In the case of temperature dependent material properties, the evaluation is done at the uniform temperature (input as **BFUNIF**,TEMP) for the first load step. The material is always positive definite if the material is isotropic or if ν_{xy}, ν_{yz}, and ν_{xz} are all zero. When using

the major Poisson's ratios (PRXY, PRYZ, PRXZ), h as defined in Equation 2.20 (p. 8) must be positive for the material to be positive definite.

2.1.2. Orthotropic Material Transformation for Axisymmetric Models

The transformation of material property data from the R-θ-Z cylindrical system to the x-y-z system used for the input requires special care. The conversion of the Young's moduli is fairly direct, whereas the correct method of conversion of the Poisson's ratios is not obvious. Consider first how the Young's moduli transform from the global cylindrical system to the global Cartesian as used by the axisymmetric elements for a disc:

Figure 2.2: Material Coordinate Systems



Thus, $E_R \rightarrow E_x$, $E_\theta \rightarrow E_z$, $E_Z \rightarrow E_y$. Starting with the global Cartesian system, the input for x-y-z coordinates gives the following stress-strain matrix for the non-shear terms (from Equation 2.4 (p. 6)):

$$\left[D_{x-y-z}\right]^{-1} = \begin{bmatrix} 1/E_x & -\nu_{xy}/E_x & -\nu_{xz}/E_x \\ -\nu_{yx}/E_y & 1/E_y & -\nu_{yz}/E_y \\ -\nu_{zx}/E_z & -\nu_{zy}/E_z & 1/E_z \end{bmatrix} \tag{2.22}$$

Rearranging so that the R-θ-Z axes match the x-y-z axes (i.e., x → R, y → Z, z → θ):

$$[D_{R-\theta-Z}]^{-1} = \begin{bmatrix} 1/E_R & -\nu_{RZ}/E_R & -\nu_{R\theta}/E_R \\ -\nu_{ZR}/E_Z & 1/E_Z & -\nu_{Z\theta}/E_Z \\ -\nu_{\theta R}/E_\theta & -\nu_{\theta Z}/E_\theta & 1/E_\theta \end{bmatrix} \tag{2.23}$$

If one coordinate system uses the major Poisson's ratios, and the other uses the minor Poisson's ratio, an additional adjustment will need to be made.

Comparing Equation 2.22 (p. 9) and Equation 2.23 (p. 9) gives:

$$E_x = E_R \tag{2.24}$$

$$E_y = E_Z \tag{2.25}$$

$$E_z = E_\theta \tag{2.26}$$

$$\nu_{xy} = \nu_{RZ} \tag{2.27}$$

$$\nu_{yz} = \nu_{Z\theta} \tag{2.28}$$

$$\nu_{xz} = \nu_{R\theta} \tag{2.29}$$

This assumes that: ν_{xy}, ν_{yz}, ν_{xz} and ν_{RZ}, $\nu_{R\theta}$, $\nu_{Z\theta}$ are all major Poisson's ratios (i.e., $E_x \geq E_Y \geq E_z$ and $E_R \geq E_Z \geq E_\theta$).

If this is not the case (e.g., $E_\theta > E_Z$):

$$\nu_{\theta z} = \nu_{z\theta}\frac{E_\theta}{E_z} = \text{largest Poisson ratio (input as NUYZ)} \tag{2.30}$$

2.1.3. Temperature-Dependent Coefficient of Thermal Expansion

Considering a typical component, the thermal strain from Equation 2.3 (p. 6) is:

$$\varepsilon^{th} = \alpha^{se}(T)(T - T_{ref}) \tag{2.31}$$

where:

$\alpha^{se}(T)$ = temperature-dependent secant coefficient of thermal expansion (SCTE)

$\alpha^{se}(T)$ is input in one of three ways:

1. Input $\alpha^{se}(T)$ directly (input as ALPX, ALPY, or ALPZ on **MP** command)
2. Computed using Equation 2.34 (p. 11) from $\alpha^{in}(T)$, the instantaneous coefficients of thermal expansion (input as CTEX, CTEY, or CTEZ on **MP** command)
3. Computed using Equation 2.32 (p. 10) from $\varepsilon^{ith}(T)$, the input thermal strains (input as THSX, THSY, or THSZ on **MP** command)

$\alpha^{se}(T)$ is computed from $\varepsilon^{ith}(T)$ by rearranging Equation 2.31 (p. 10):

$$\alpha^{se}(T) = \frac{\varepsilon^{ith}(T)}{T - T_{ref}} \tag{2.32}$$

Equation 2.32 (p. 10) assumes that when $T = T_{ref}$, $\varepsilon^{ith} = 0$. If this is not the case, the ε^{ith} data is shifted automatically by a constant so that it is true. α^{se} at T_{ref} is calculated based on the slopes from the adjacent user-defined data points. Hence, if the slopes of ε^{ith} above and below T_{ref} are not identical, a step change in α^{se} at T_{ref} will be computed.

$\varepsilon^{th}(T)$ (thermal strain) is related to $\alpha^{in}(T)$ by:

$$\varepsilon^{th}(T) = \int_{T_{ref}}^{T} \alpha^{in}(T)T \tag{2.33}$$

Combining this with equation Equation 2.32 (p. 10),

$$\alpha^{se}(T) = \frac{\int_{T_{ref}}^{T} \alpha^{in}(T)dT}{T - T_{ref}} \tag{2.34}$$

No adjustment is needed for $\alpha^{in}(T)$ as $\alpha^{se}(T)$ is defined to be $\alpha^{in}(T)$ when $T = T_{ref}$.

As seen above, $\alpha^{se}(T)$ is dependent on what was used for T_{ref}. If $\alpha^{se}(T)$ was defined using T_{ref} as one value but then the thermal strain was zero at another value, an adjustment needs to be made (using the **MPAMOD** command). Consider:

$$\varepsilon_o^{th} = \alpha_o^{se}(T)(T - T_o) = \int_{T_o}^{T} \alpha^{in} dT \tag{2.35}$$

$$\varepsilon_r^{th} = \alpha_r^{se}(T)(T - T_{ref}) = \int_{T_{ref}}^{T} \alpha^{in} dT \tag{2.36}$$

Equation 2.35 (p. 11) and Equation 2.36 (p. 11) represent the thermal strain at a temperature T for two different starting points, T_o and T_{ref}. Now let T_o be the temperature about which the data has been generated (definition temperature), and T_{ref} be the temperature at which all strains are zero (reference temperature). Thus, α_o^{se} is the supplied data, and α_r^{se} is what is needed as program input.

The right-hand side of Equation 2.35 (p. 11) may be expanded as:

$$\int_{T_o}^{T} \alpha^{in} dT = \int_{T_o}^{T_{ref}} \alpha^{in} dT + \int_{T_{ref}}^{T} \alpha^{in} dT \tag{2.37}$$

also,

$$\int_{T_o}^{T_{ref}} \alpha^{in} dT = \alpha_o^{se}(T_{ref})(T_{ref} - T_o) \tag{2.38}$$

or

$$\int_{T_o}^{T_{ref}} \alpha^{in} dT = \alpha_r^{se}(T_o)(T_{ref} - T_o) \tag{2.39}$$

Combining Equation 2.35 (p. 11) through Equation 2.38 (p. 11),

$$\alpha_r^{se}(T) = \alpha_o^{se}(T) + \frac{T_{ref} - T_o}{T - T_{ref}}(\alpha_o^{se}(T) - \alpha_o^{se}(T_{ref})) \tag{2.40}$$

Thus, Equation 2.40 (p. 11) must be accounted for when making an adjustment for the definition temperature being different from the strain-free temperature. This adjustment may be made (using the **MPAMOD** command).

Note that:

Equation 2.40 (p. 11) is nonlinear. Segments that were straight before may be no longer straight. Hence, extra temperatures may need to be specified initially (using the **MPTEMP** command).

If $T_{ref} = T_o$, Equation 2.40 (p. 11) is trivial.
If $T = T_{ref}$, Equation 2.40 (p. 11) is undefined.

The values of T as used here are the temperatures used to define α^{se} (input on **MPTEMP** command). Thus, when using the α^{se} adjustment procedure, it is recommended to avoid defining a T value to be the same as $T = T_{ref}$ (to a tolerance of one degree). If a T value is the same as T_{ref}, and:

- the T value is at either end of the input range, then the new α^{se} value is simply the same as the new α value of the nearest adjacent point.
- the T value is not at either end of the input range, then the new α^{se} value is the average of the two adjacent new α values.

2.2. Derivation of Structural Matrices

The principle of virtual work states that a virtual (very small) change of the internal strain energy must be offset by an identical change in external work due to the applied loads, or:

$$\delta U = \delta V \tag{2.41}$$

where:

U = strain energy (internal work) = $U_1 + U_2$
V = external work = $V_1 + V_2 + V_3$
δ = virtual operator

The virtual strain energy is:

$$\delta U_1 = \int_{vol} \{\delta\varepsilon\}\{\sigma\} d(vol)^T \tag{2.42}$$

where:

$\{\varepsilon\}$ = strain vector
$\{\sigma\}$ = stress vector
vol = volume of element

Continuing the derivation assuming linear materials and geometry, Equation 2.41 (p. 12) and Equation 2.42 (p. 12) are combined to give:

$$\delta U_1 = \int_{vol} (\{\delta\varepsilon\}^T [D]\{\varepsilon\} - \{\delta\varepsilon\}^T [D]\{\varepsilon^{th}\}) d(vol) \tag{2.43}$$

The strains may be related to the nodal displacements by:

$$\{\varepsilon\} = [B]\{u\} \tag{2.44}$$

where:

[B] = strain-displacement matrix, based on the element shape functions
{u} = nodal displacement vector

It will be assumed that all effects are in the global Cartesian system. Combining Equation 2.44 (p. 12) with Equation 2.43 (p. 12), and noting that {u} does not vary over the volume:

$$\delta U_1 = \{\delta u\}^T \int_{vol} [B]^T [D][B] d(vol)\{u\} - \{\delta u\}^T \int_{vol} [B]^T [D]\{\varepsilon^{th}\} d(vol) \tag{2.45}$$

Another form of virtual strain energy is when a surface moves against a distributed resistance, as in a foundation stiffness. This may be written as:

$$\delta U_2 = \int_{area_f} \{\delta w_n\}^T \{\sigma\} d(area_f) \tag{2.46}$$

where:

$\{w_n\}$ = motion normal to the surface
$\{\sigma\}$ = stress (or pressure) carried by the surface
$area_f$ = area of the distributed resistance

Both $\{w_n\}$ and $\{\sigma\}$ will usually have only one nonzero component. The point-wise normal displacement is related to the nodal displacements by:

$$\{w_n\} = [N_n]\{u\} \tag{2.47}$$

where:

$[N_n]$ = matrix of shape functions for normal motions at the surface

The stress, $\{\sigma\}$, is

$$\{\sigma\} = k\{w_n\} \tag{2.48}$$

where:

k = the foundation stiffness in units of force per length per unit area

Combining Equation 2.46 (p. 13) through Equation 2.48 (p. 13), and assuming that k is constant over the area,

$$\delta U_2 = \{\delta u\}^T k \int_{area_f} [N_n]^T [N_n] d(area_f)\{u\} \tag{2.49}$$

Next, the external virtual work will be considered. The inertial effects will be studied first:

$$\delta V_1 = -\int_{vol} \{\delta w\}^T \frac{\{F^a\}}{vol} d(vol) \tag{2.50}$$

where:

$\{w\}$ = vector of displacements of a general point
$\{F^a\}$ = acceleration (D'Alembert) force vector

According to Newton's second law:

$$\frac{\{F^a\}}{vol} = \rho \frac{\partial^2}{\partial t^2}\{w\} \tag{2.51}$$

where:

ρ = density (input as DENS on **MP** command)
t = time

The displacements within the element are related to the nodal displacements by:

$$\{w\} = [N]\{u\} \tag{2.52}$$

where [N] = matrix of shape functions. Combining Equation 2.50 (p. 13), Equation 2.51 (p. 13), and Equation 2.52 (p. 14), and assuming that ρ is constant over the volume,

$$\delta V_1 = -\{\delta u\}^T \rho \int_{vol} [N]^T [N] d(vol) \frac{\delta^2}{\delta t^2} \{u\} \tag{2.53}$$

The pressure force vector formulation starts with:

$$\delta V_2 = \int_{area_p} \{\delta w_n\}^T \{P\} d(area_p) \tag{2.54}$$

where:

{P} = the applied pressure vector (normally contains only one nonzero component)
area $_p$ = area over which pressure acts

Combining equations Equation 2.52 (p. 14) and Equation 2.54 (p. 14),

$$\delta V_2 = \{\delta u\}^T \int_{area_p} [N_n]\{P\} d(area_p) \tag{2.55}$$

Unless otherwise noted, pressures are applied to the outside surface of each element and are normal to curved surfaces, if applicable.

Nodal forces applied to the element can be accounted for by:

$$\delta V_3 = \{\delta u\}^T \{F_e^{nd}\} \tag{2.56}$$

where:

$\{F_e^{nd}\}$ = nodal forces applied to the element

Finally, Equation 2.41 (p. 12), Equation 2.45 (p. 13), Equation 2.49 (p. 13), Equation 2.53 (p. 14), Equation 2.55 (p. 14), and Equation 2.56 (p. 14) may be combined to give:

$$\begin{aligned} &\{\delta u\}^T \int_{vol} [B]^T [D][B] d(vol)\{u\} - \{\delta u\}^T \int_{vol} [B]^T [D]\{\varepsilon^{th}\} d(vol) \\ &\quad + \{\delta u\}^{T_k} \int_{area_f} [N_n]^T [N_n] d(area_f)\{u\} \\ &= -\{\delta u\}^T \rho \int_{vol} [N]^T [N] d(vol) \frac{\delta^2}{\delta t^2} \{u\} + \{\delta u\}^T \int_{area_p} [N_n]^T \{P\} d(area_p) + \{\delta u\}^T \{F_e^{nd}\} \end{aligned} \tag{2.57}$$

Noting that the $\{\delta u\}^T$ vector is a set of arbitrary virtual displacements common in all of the above terms, the condition required to satisfy equation Equation 2.57 (p. 14) reduces to:

$$([K_e] + [K_e^f])\{u\} - \{F_e^{th}\} = [M_e]\{\ddot{u}\} + \{F_e^{pr}\} + \{F_e^{nd}\} \tag{2.58}$$

where:

$$[K_e] = \int_{vol}[B]^T[D][B]d(vol) = \text{element stiffness matrix}$$

$$[K_e^f] = k\int_{area_f}[N_n]^T[N_n]d(area_f) = \text{element foundation stiffness matrix}$$

$$\{F_e^{th}\} = \int_{vol}[B]^T[D]\{\varepsilon^{th}\}d(vol) = \text{element thermal load vector}$$

$$[M_e] = \rho\int_{vol}[N]^T[N]d(vol) = \text{element mass matrix}$$

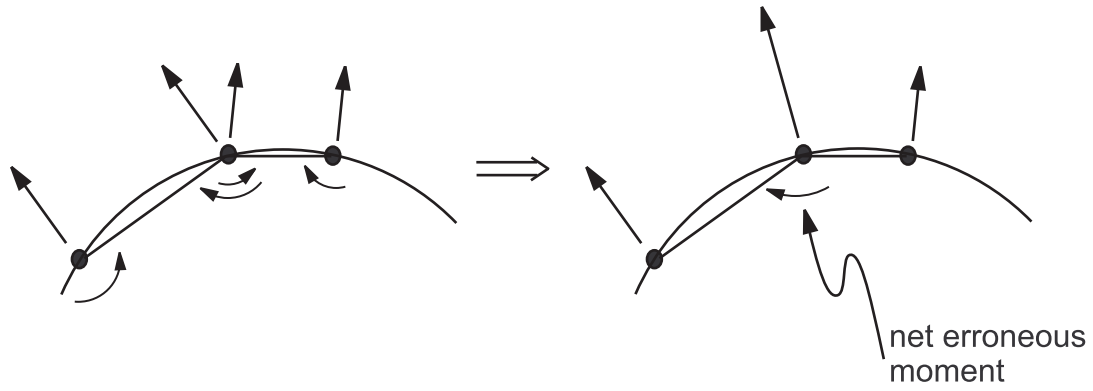
$$\{\ddot{u}\} = \frac{\partial^2}{\partial t^2}\{u\} = \text{acceleration vector (such as gravity effects)}$$

$$\{F_e^{pr}\} = \int_{area_p}[N_n]^T\{P\}d(area_p) = \text{element pressure vector}$$

Equation 2.58 (p. 14) represents the equilibrium equation on a one element basis.

The above matrices and load vectors were developed as "consistent". Other formulations are possible. For example, if only diagonal terms for the mass matrix are requested (**LUMPM**,ON), the matrix is called "lumped" (see Lumped Matrices (p. 391)). For most lumped mass matrices, the rotational degrees of freedom (DOFs) are removed. If the rotational DOFs are requested to be removed (KEYOPT commands with certain elements), the matrix or load vector is called "reduced". Thus, use of the reduced pressure load vector does not generate moments as part of the pressure load vector. Use of the consistent pressure load vector can cause erroneous internal moments in a structure. An example of this would be a thin circular cylinder under internal pressure modelled with irregular shaped shell elements. As suggested by Figure 2.3: Effects of Consistent Pressure Loading (p. 15), the consistent pressure loading generates an erroneous moment for two adjacent elements of dissimilar size.

Figure 2.3: Effects of Consistent Pressure Loading



2.3. Structural Strain and Stress Evaluations

2.3.1. Integration Point Strains and Stresses

The element integration point strains and stresses are computed by combining equations Equation 2.1 (p. 5) and Equation 2.44 (p. 12) to get:

$$\{\varepsilon^{el}\} = [B]\{u\} - \{\varepsilon^{th}\} \tag{2.59}$$

$$\{\sigma\} = [D]\{\varepsilon^{el}\} \tag{2.60}$$

where:

$\{\varepsilon^{el}\}$ = strains that cause stresses (output as EPEL)
[B] = strain-displacement matrix evaluated at integration point
{u} = nodal displacement vector
$\{\varepsilon^{th}\}$ = thermal strain vector
$\{\sigma\}$ = stress vector (output as S)
[D] = elasticity matrix

Nodal and centroidal stresses are computed from the integration point stresses as described in Nodal and Centroidal Data Evaluation (p. 409).

2.3.2. Surface Stresses

Surface stress output may be requested on "free" faces of 2-D and 3-D elements. "Free" means not connected to other elements as well as not having any imposed displacements or nodal forces normal to the surface. The following steps are executed at each surface Gauss point to evaluate the surface stresses. The integration points used are the same as for an applied pressure to that surface.

1. Compute the in-plane strains of the surface at an integration point using:

$$\{\varepsilon'\} = [B']\{u'\} - \{(\varepsilon^{th'})\} \tag{2.61}$$

Hence, ε'_x, ε'_y and ε'_{xy} are known. The prime (') represents the surface coordinate system, with z being normal to the surface.

2. A each point, set:

$$\sigma'_z = -P \tag{2.62}$$

$$\sigma'_{xz} = 0 \tag{2.63}$$

$$\sigma'_{yz} = 0 \tag{2.64}$$

where P is the applied pressure. Equation 2.63 (p. 16) and Equation 2.64 (p. 16) are valid, as the surface for which stresses are computed is presumed to be a free surface.

3. At each point, use the six material property equations represented by:

$$\{\sigma'\} = [D']\{\varepsilon'\} \tag{2.65}$$

to compute the remaining strain and stress components (ε'_z, ε'_{xz}, ε'_{yz}, σ'_x, σ'_y and σ'_{xy}).

4. Repeat and average the results across all integration points.

2.3.3. Shell Element Output

For elastic shell elements, the forces and moments per unit length (using shell nomenclature) are computed as:

$$T_x = \int_{-t/2}^{t/2} \sigma_x dz \tag{2.66}$$

$$T_y = \int_{-t/2}^{t/2} \sigma_y dz \tag{2.67}$$

$$T_{xy} = \int_{-t/2}^{t/2} \sigma_{xy} dz \tag{2.68}$$

$$M_x = \int_{-t/2}^{t/2} z\sigma_x dz \tag{2.69}$$

$$M_y = \int_{-t/2}^{t/2} z\sigma_y dz \tag{2.70}$$

$$M_{xy} = \int_{-t/2}^{t/2} z\sigma_{xy} dz \tag{2.71}$$

$$N_x = \int_{-t/2}^{t/2} \sigma_{xz} dz \tag{2.72}$$

$$N_y = \int_{-t/2}^{t/2} \sigma_{yz} dz \tag{2.73}$$

where:

T_x, T_y, T_{xy} = in-plane forces per unit length (output as TX, TY, and TXY)
M_x, M_y, M_{xy} = bending moments per unit length (output as MX, MY, and MXY)
N_x, N_y = transverse shear forces per unit length (output as NX and NY)
t = thickness at midpoint of element, computed normal to center plane
σ_x, etc. = direct stress (output as SX, etc.)
σ_{xy}, etc. = shear stress (output as SXY, etc.)

For shell elements with linearly elastic material, Equation 2.66 (p. 16) to Equation 2.73 (p. 17) reduce to:

$$T_x = \frac{t(\sigma_{x,top} + 4\sigma_{x,mid} + \sigma_{x,bot})}{6} \tag{2.74}$$

$$T_y = \frac{t(\sigma_{y,top} + 4\sigma_{y,mid} + \sigma_{y,bot})}{6} \tag{2.75}$$

$$T_{xy} = \frac{t(\sigma_{xy,top} + 4\sigma_{xy,mid} + \sigma_{xy,bot})}{6} \tag{2.76}$$

$$M_x = \frac{t^2(\sigma_{x,top} - \sigma_{x,bot})}{12} \tag{2.77}$$

$$M_y = \frac{t^2(\sigma_{y,top} - \sigma_{y,bot})}{12} \tag{2.78}$$

$$M_{xy} = \frac{t^2(\sigma_{xy,top} - \sigma_{xy,bot})}{12} \tag{2.79}$$

$$N_x = \frac{t(\sigma_{xz,top} + 4\sigma_{xz,mid} + \sigma_{xz,bot})}{6} \tag{2.80}$$

$$N_y = \frac{t(\sigma_{yz,top} + 4\sigma_{yz,mid} + \sigma_{yz,bot})}{6} \tag{2.81}$$

For shell elements with nonlinear materials, Equation 2.66 (p. 16) to Equation 2.73 (p. 17) are numerically integrated.

It should be noted that the shell nomenclature and the nodal moment conventions are in apparent conflict with each other. For example, a cantilever beam located along the x axis and consisting of shell elements in the x-y plane that deforms in the z direction under a pure bending load with coupled nodes at the free end, has the following relationship:

$$M_x b = F_{MY} \tag{2.82}$$

where:

b = width of beam
F_{MY} = nodal moment applied to the free end (input as *VALUE* on **F** command with *Lab* = MY (not MX))

The shape functions of the shell element result in constant transverse strains and stresses through the thickness. Some shell elements adjust these values so that they will peak at the mid-surface with 3/2 of the constant value and be zero at both surfaces, as noted in the element discussions in Element Library (p. 411).

The through-thickness stress (σ_z) is set equal to the negative of the applied pressure at the surfaces of the shell elements, and linearly interpolated in between.

2.4. Combined Stresses and Strains

When a model has only one functional direction of strains and stress, comparison with an allowable value is straightforward. But when there is more than one component, the components are normally combined into one number to allow a comparison with an allowable. This section discusses different ways of doing that combination, representing different materials and/or technologies.

2.4.1. Combined Strains

The principal strains are calculated from the strain components by the cubic equation:

$$\begin{vmatrix} \varepsilon_x - \varepsilon_o & \frac{1}{2}\varepsilon_{xy} & \frac{1}{2}\varepsilon_{xz} \\ \frac{1}{2}\varepsilon_{xy} & \varepsilon_y - \varepsilon_o & \frac{1}{2}\varepsilon_{yz} \\ \frac{1}{2}\varepsilon_{xz} & \frac{1}{2}\varepsilon_{yz} & \varepsilon_z - \varepsilon_o \end{vmatrix} = 0 \tag{2.83}$$

where:

ε_o = principal strain (3 values)

The three principal strains are labeled ε_1, ε_2, and ε_3 (output as 1, 2, and 3 with strain items such as EPEL). The principal strains are ordered so that ε_1 is the most positive and ε_3 is the most negative.

The strain intensity ε_I (output as INT with strain items such as EPEL) is the largest of the absolute values of ε_1 - ε_2, ε_2 - ε_3, or ε_3 - ε_1. That is:

$$\varepsilon_I = \text{MAX}(|\varepsilon_1 - \varepsilon_2|, |\varepsilon_2 - \varepsilon_3|, |\varepsilon_3 - \varepsilon_1|) \tag{2.84}$$

The von Mises or equivalent strain ε_e (output as EQV with strain items such as EPEL) is computed as:

$$\varepsilon_e = \frac{1}{1+\nu'}\left(\frac{1}{2}\left[(\varepsilon_1 - \varepsilon_2)^2 + (\varepsilon_2 - \varepsilon_3)^2 + (\varepsilon_3 - \varepsilon_1)^2\right]\right)^{\frac{1}{2}} \tag{2.85}$$

where:

ν' = effective Poisson's ratio =
- user input value (EFFNU on **AVPRIN** command)
- default
  - Poisson's ratio (as defined on the **MP** commands) for EPEL and EPTH
  - 0.5 for EPPL and EPCR
  - 0.5 if referenced material is hyperelastic and usermat (accessed with **TB**,HYPER, **TB**,AHYPER, and/or **TB**,USER)
  - 0.0 for line elements (includes beam, link, and pipe elements, as well as discrete elements), cyclic symmetry analysis, and load case operations (**LOPER**)

2.4.2. Combined Stresses

The principal stresses (σ_1, σ_2, σ_3) are calculated from the stress components by the cubic equation:

$$\begin{vmatrix} \sigma_x - \sigma_o & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y - \sigma_o & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z - \sigma_o \end{vmatrix} = 0 \tag{2.86}$$

where:

σ_o = principal stress (3 values)

The three principal stresses are labeled σ_1, σ_2, and σ_3 (output quantities S1, S2, and S3). The principal stresses are ordered so that σ_1 is the most positive (tensile) and σ_3 is the most negative (compressive).

The stress intensity σ_I (output as SINT) is the largest of the absolute values of σ_1 - σ_2, σ_2 - σ_3, or σ_3 - σ_1. That is:

$$\sigma_I = \text{MAX}(|\sigma_1 - \sigma_2|\,|\sigma_2 - \sigma_3|\,|\sigma_3 - \sigma_1|) \tag{2.87}$$

The von Mises or equivalent stress σ_e (output as SEQV) is computed as:

$$\sigma_e = \left(\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]\right)^{\frac{1}{2}} \tag{2.88}$$

or

$$\sigma_e = \left(\frac{1}{2} \left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{xz}^2) \right] \right)^{\frac{1}{2}} \quad (2.89)$$

When $\nu' = \nu$ (input as PRXY or NUXY on **MP** command), the equivalent stress is related to the equivalent strain through

$$\sigma_e = E\varepsilon_e \quad (2.90)$$

where:

E = Young's modulus (input as EX on **MP** command)

2.4.3. Failure Criteria

Use failure criteria to assess the possibility of failure of a material. Doing so allows the consideration of orthotropic materials, which might be much weaker in one direction than another. Failure criteria are available in POST1 for all plane, shell, and solid structural elements (using the **TB** or **FC** family of commands).

Possible failure of a material can be evaluated by up to 20 different criteria, of which 11 are predefined. They are evaluated at the top and bottom (or middle) of each layer at each of the in-plane integration points.

The following topics related to the predefined failure criteria are available:

2.4.3.1. Maximum Strain Failure Criteria

$$\xi_1 = \text{maximum of} \left\{ \begin{array}{l} \frac{\varepsilon_{xt}}{\varepsilon_{xt}^f} \text{ or } \frac{\varepsilon_{xc}}{\varepsilon_{xc}^f} \text{ whichever is applicable} \\ \frac{\varepsilon_{yt}}{\varepsilon_{yt}^f} \text{ or } \frac{\varepsilon_{yc}}{\varepsilon_{yc}^f} \text{ whichever is applicable} \\ \frac{\varepsilon_{zt}}{\varepsilon_{zt}^f} \text{ or } \frac{\varepsilon_{zc}}{\varepsilon_{zc}^f} \text{ whichever is applicable} \\ \frac{|\varepsilon_{xy}|}{\varepsilon_{xy}^f} \\ \frac{|\varepsilon_{yz}|}{\varepsilon_{yz}^f} \\ \frac{|\varepsilon_{xz}|}{\varepsilon_{xz}^f} \end{array} \right. \quad (2.91)$$

where:

$\xi_1 =$ value of maximum strain failure criterion

$\varepsilon_{xt} = \begin{cases} 0 \\ \varepsilon_x \end{cases}$ whichever is greater

$\varepsilon_x =$ strain in layer x-direction

$\varepsilon_{xc} = \begin{cases} \varepsilon_x \\ 0 \end{cases}$ whichever is lesser

$\varepsilon_{xt}^f =$ failure strain in layer x-direction in tension

$\varepsilon_{xc}^f =$ failure strain in layer x-direction in compression

$\varepsilon_{yt} = \begin{cases} 0 \\ \varepsilon_y \end{cases}$ whichever is greater

$\varepsilon_y =$ strain in layer y-direction

$\varepsilon_{yc} = \begin{cases} \varepsilon_y \\ 0 \end{cases}$ whichever is lesser

$\varepsilon_{yt}^f =$ failure strain in layer y-direction in tension

$\varepsilon_{yc}^f =$ failure strain in layer y-direction in compression

$\varepsilon_{zt} = \begin{cases} 0 \\ \varepsilon_z \end{cases}$ whichever is greater

$\varepsilon_z =$ strain in layer z-direction

$\varepsilon_{zc} = \begin{cases} \varepsilon_z \\ 0 \end{cases}$ whichever is lesser

$\varepsilon_{zt}^f =$ failure strain in layer z-direction in tension

$\varepsilon_{zc}^f =$ failure strain in layer z-direction in compression

$\varepsilon_{xy} =$ shear strain in layer x-y direction

$\varepsilon_{xy}^f =$ failure shear strain in layer x-y direction

$\varepsilon_{yz} =$ shear strain in layer y-z direction

$\varepsilon_{yz}^f =$ failure shear strain in layer y-z direction

$\varepsilon_{xz} =$ shear strain in layer x-z direction

$\varepsilon_{xz}^f =$ failure shear strain in layer x-z direction

2.4.3.2. Maximum Stress Failure Criteria

$$\xi_2 = \text{maximum of} \begin{cases} \dfrac{\sigma_{xt}}{\sigma_{xt}^f} \text{ or } \dfrac{\sigma_{xc}}{\sigma_{xc}^f} \text{ whichever is applicable} \\ \dfrac{\sigma_{yt}}{\sigma_{yt}^f} \text{ or } \dfrac{\sigma_{yc}}{\sigma_{yc}^f} \text{ whichever is applicable} \\ \dfrac{\sigma_{zt}}{\sigma_{zt}^f} \text{ or } \dfrac{\sigma_{zc}}{\sigma_{zc}^f} \text{ whichever is applicable} \\ \dfrac{|\sigma_{xy}|}{\sigma_{xy}^f} \\ \dfrac{|\sigma_{yz}|}{\sigma_{yz}^f} \\ \dfrac{|\sigma_{xz}|}{\sigma_{xz}^f} \end{cases} \tag{2.92}$$

where:

$\xi_2 =$ value of maximum stress failure criterion

$\sigma_{xt} = \begin{cases} 0 \\ \sigma_x \end{cases}$ whichever is greater

$\sigma_x =$ stress in layer x-direction

$\sigma_{xc} = \begin{cases} \sigma_x \\ 0 \end{cases}$ whichever is lesser

$\sigma_{xt}^f =$ failure stress in layer x-direction in tension

$\sigma_{xc}^f =$ failure stress in layer x-direction in compression

$\sigma_{yt} = \begin{cases} 0 \\ \sigma_y \end{cases}$ whichever is greater

$\sigma_y =$ stress in layer y-direction

$\sigma_{yc} = \begin{cases} \sigma_y \\ 0 \end{cases}$ whichever is lesser

$\sigma_{yt}^f =$ failure stress in layer y-direction in tension

$\sigma_{yc}^f =$ failure stress in layer y-direction in compression

$\sigma_{zt} = \begin{cases} 0 \\ \sigma_z \end{cases}$ whichever is greater

$\sigma_z =$ stress in layer z-direction

$\sigma_{zc} = \begin{cases} \sigma_z \\ 0 \end{cases}$ whichever is lesser

σ_{zt}^f = failure stress in layer z-direction in tension

σ_{zc}^f = failure stress in layer z-direction in compression

σ_{xy} = shear stress in layer x-y direction

σ_{xy}^f = failure shear stress in layer x-y direction

σ_{yz} = shear stress in layer y-z direction

σ_{yz}^f = failure shear stress in layer y-z direction

σ_{xz} = shear stress in layer x-z direction

σ_{xz}^f = failure shear stress in layer x-z direction

2.4.3.3. Tsai-Wu Failure Criteria

If the criterion used is the “strength index”:

$$\xi_3 = A + B \tag{2.93}$$

and if the criterion used is the inverse of the "strength ratio":

$$\xi_3 = 1.0 / \left(-\frac{B}{2A} + \sqrt{(B/2A)^2 + 1.0/A} \right) \tag{2.94}$$

where:

ξ_3 = value of Tsai-Wu failure criterion

$$A = -\frac{(\sigma_x)^2}{\sigma_{xt}^f \sigma_{xc}^f} - \frac{(\sigma_y)^2}{\sigma_{yt}^f \sigma_{yc}^f} - \frac{(\sigma_z)^2}{\sigma_{zt}^f \sigma_{zc}^f} + \frac{(\sigma_{xy})^2}{(\sigma_{xy}^f)^2} + \frac{(\sigma_{yz})^2}{(\sigma_{yz}^f)^2} + \frac{(\sigma_{xz})^2}{(\sigma_{xz}^f)^2}$$
$$+ \frac{C_{xy}\sigma_x\sigma_y}{\sqrt{\sigma_{xt}^f \sigma_{xc}^f \sigma_{yt}^f \sigma_{tc}^f}} + \frac{C_{yz}\sigma_y\sigma_z}{\sqrt{\sigma_{yt}^f \sigma_{yc}^f \sigma_{zt}^f \sigma_{zc}^f}} + \frac{C_{xz}\sigma_x\sigma_z}{\sqrt{\sigma_{xt}^f \sigma_{xc}^f \sigma_{zt}^f \sigma_{zc}^f}}$$

$$B = \left(\frac{1}{\sigma_{xt}^f} + \frac{1}{\sigma_{xc}^f} \right) \sigma_x + \left(\frac{1}{\sigma_{yt}^f} + \frac{1}{\sigma_{yc}^f} \right) \sigma_y + \left(\frac{1}{\sigma_{zt}^f} + \frac{1}{\sigma_{zc}^f} \right) \sigma_z$$

C_{xy}, C_{yz}, C_{xz} = x-y, y-z, x-z, respectively, coupling coefficient for Tsai-Wu theory

The Tsai-Wu failure criteria used here are 3-D versions of the failure criterion reported in Tsai and Hahn([190] (p. 931)) for the 'strength index' and in Tsai([93] (p. 925)) for the 'strength ratio'. Apparent differences are:

1. The program input uses negative values for compression limits, whereas Tsai uses positive values for all limits.

2. The program uses C_{xy} instead of the F_{xy}^{*} used by Tsai and Hahn with C_{xy} being twice the value of F_{xy}^{*}.

2.4.3.4. Physical Failure Criteria

The physical failure criteria are specially formulated to account for different damage mechanisms (fiber and matrix failure) in fiber-reinforced composite materials.

Predefined physical failure criteria include the following:

2.4.3.4.1. Hashin Fiber Failure Criterion

$$\xi_4 = \begin{cases} \left(\frac{\sigma_x}{\sigma_{xt}^f}\right)^2 + \frac{\sigma_{xy}{}^2 + \sigma_{xz}{}^2}{(\sigma_{xy}^f)^2} & \text{if } \sigma_x > 0 \\ \left(\frac{\sigma_x}{\sigma_{xc}^f}\right)^2 & \text{if } \sigma_x \le 0 \end{cases} \tag{2.95}$$

2.4.3.4.2. Hashin Matrix Failure Criterion

$$\xi_5 = \begin{cases} \left(\frac{\sigma_{y+}\sigma_z}{\sigma_{yt}^f}\right)^2 + \frac{\sigma_{yz}{}^2 - \sigma_y\sigma_z}{(\sigma_{yz}^f)^2} + \frac{\sigma_{xy}{}^2 + \sigma_{xz}{}^2}{(\sigma_{xy}^f)^2} & \text{if } \sigma_y + \sigma_z > 0 \\ \frac{1}{\sigma_{yc}^f}\left(\left(\frac{\sigma_{yc}^f}{2\sigma_{yz}^f}\right)^2 - 1\right)(\sigma_y + \sigma_z) + \left(\frac{\sigma_y + \sigma_z}{2\sigma_{yz}^f}\right)^2 + \frac{\sigma_{yz}{}^2 - \sigma_y\sigma_z}{(\sigma_{yz}^f)^2} + \frac{\sigma_{xy}{}^2 + \sigma_{xz}{}^2}{(\sigma_{xy}^f)^2} & \text{if } \sigma_y + \sigma_z \le 0 \end{cases} \tag{2.96}$$

2.4.3.4.3. Puck Fiber Failure Criterion ([401] (p. 943))

$$\xi_6 = \begin{cases} \frac{\sigma_x}{\sigma_{xt}^f} & \text{if } \sigma_x > 0 \\ \frac{\sigma_x}{\sigma_{xc}^f} & \text{if } \sigma_x \le 0 \end{cases} \tag{2.97}$$

2.4.3.4.4. Puck Matrix Failure Criterion ([401] (p. 943))

$$\xi_7 = \begin{cases} \sqrt{\left(\frac{1}{\sigma_{yt}^f} - P_t\right)^2 \sigma_n{}^2 + \left(\frac{\sigma_{n2}}{R}\right)^2 + \left(\frac{\sigma_{n1}}{\sigma_{xy}^f}\right)^2} + K_t\sigma_n & \text{if } \sigma_n > 0 \\ \sqrt{\left(\frac{\sigma_{n2}}{R}\right)^2 + \left(\frac{\sigma_{n1}}{\sigma_{xy}^f}\right)^2 + (K_c\sigma_n)^2} + K_c\sigma_n & \text{if } \sigma_n \le 0 \end{cases} \tag{2.98}$$

where:

$\sigma_{n,}\sigma_{n1},\sigma_{n2}$ = stresses on an action plane parallel to the fibers

$$R = \begin{cases} \sigma_{yz}^{f} & \text{if } \sigma_{yz}^{f} \text{ is provided} \\ \dfrac{\left|\sigma_{yc}^{f}\right|}{2\left(1+P_{yz}^{c}\right)} & \text{default} \end{cases}$$

$$K_t = \begin{cases} \dfrac{1}{(\sigma_{n1}{}^2 + \sigma_{n2}{}^2)}\left(\dfrac{P_{yz}^{t}}{R}\sigma_{n2}{}^2 + \dfrac{P_{xz}^{t}}{\sigma_{xy}^{f}}\sigma_{n1}{}^2\right) & \text{if } (\sigma_{n1}{}^2 + \sigma_{n2}{}^2) > 0 \\ 0 & \text{if } \left(\sigma_{n1}{}^2 + \sigma_{n2}{}^2\right) = 0 \end{cases}$$

$$K_c = \begin{cases} \dfrac{1}{(\sigma_{n1}{}^2 + \sigma_{n2}{}^2)}\left(\dfrac{P_{yz}^{c}}{R}\sigma_{n2}{}^2 + \dfrac{P_{xz}^{c}}{\sigma_{xy}^{f}}\sigma_{n1}{}^2\right) & \text{if } (\sigma_{n1}{}^2 + \sigma_{n2}{}^2) > 0 \\ 0 & \text{if } \left(\sigma_{n1}{}^2 + \sigma_{n2}{}^2\right) = 0 \end{cases}$$

$P_{xz}^{t}, P_{xz}^{t}, P_{yz}^{t}, P_{yz}^{c}$ = four inclination parameters

2.4.3.4.5. LaRc03 Fiber Failure Criterion ([402] (p. 943))

$$\xi_8 = \begin{cases} \dfrac{\varepsilon_x}{\varepsilon_{xt}^{f}} & \text{if } \sigma_x \geq 0 \\ \dfrac{\left|\sigma_{12}^{m}\right| + \eta^{l}\sigma_{22}^{m}}{S^{l}} & \text{if } \sigma_x < 0 \text{ and } \sigma_{22}^{m} < 0 \\ g\left(\dfrac{\sigma_{22}^{m}}{Y^{t}}\right)^2 + \left(\dfrac{\sigma_{12}^{m}}{S^{l}}\right)^2 + (1-g)\left(\dfrac{\sigma_{22}^{m}}{Y^{t}}\right) & \text{if } \sigma_x < 0 \text{ and } \sigma_{22}^{m} \geq 0 \end{cases} \tag{2.99}$$

where:

g = fracture toughness ratio between mode I and mode II

$\sigma_{11}^{m}, \sigma_{12}^{m}, \sigma_{22}^{m}$ = 2-D stresses on the fiber misalignment (kinking) plane

η^{l} = longitudinal friction coefficient

$S^{l} = \sqrt{2}\sigma_{xy}^{f}$ = in-plane shear strength

$Y^{t} = 1.12\sqrt{2}\sigma_{yt}^{f}$ = tensile transverse strength

2.4.3.4.6. LaRc03 Matrix Failure Criterion ([402] (p. 943))

$$\xi_9 = \begin{cases} g\left(\frac{\sigma_y}{Y^t}\right)^2 + \left(\frac{\sigma_{xy}}{S^l}\right)^2 + (1-g)\left(\frac{\sigma_y}{Y^t}\right) & \text{if } \sigma_y \geq 0 \\ \left(\frac{\sigma^t}{S^t}\right)^2 + \left(\frac{\sigma^l}{S^l}\right)^2 & \text{if } \sigma_y < 0 \text{ and } \sigma_x > \left|\sigma_{yc}^f\right| \\ \left(\frac{\sigma_m^t}{S^t}\right)^2 + \left(\frac{\sigma_m^l}{S^l}\right)^2 & \text{if } \sigma_y < 0 \text{ and } \sigma_x \leq \left|\sigma_{yc}^f\right| \end{cases} \quad (2.100)$$

where:

g = fracture toughness ratio between mode I and mode II

σ^t, σ^l = effective transverse and longitudinal stresses on an action plane parallel to fibers

σ_m^t, σ_m^l = effective stresses on an action plane parallel to the misaligned fibers

$S^t = \left|\sigma_{yc}^f\right| \cos\alpha_0 \left(\sin\alpha_0 + \frac{\cos\alpha_0}{\tan 2\alpha_0}\right)$ = transverse shear strength

α_0 = initial fracture angle under pure transverse compression

2.4.3.4.7. LaRc04 Fiber Failure Criterion ([403] (p. 943))

$$\xi_{10} = \begin{cases} \frac{\sigma_x}{\sigma_{xt}^f} & \text{if } \sigma_x \geq 0 \\ \left(\frac{\sigma_{12}^m}{S^l - \eta^l \sigma_{22}^m}\right)^2 & \text{if } \sigma_x < 0 \text{ and } \sigma_{22}^m < 0 \\ (1-g)\frac{\sigma_{22}^m}{Y^t} + g\left(\frac{\sigma_{22}^m}{Y^t}\right)^2 + \left(\frac{\sigma_{12}^m}{S^l}\right)^2 + 0.5\Lambda_{23}^0\left(\frac{\sigma_{23}^m}{S^l}\right)^2 & \text{if } \sigma_x < 0 \text{ and } \sigma_{22}^m \geq 0 \end{cases} \quad (2.101)$$

$\sigma_{11}^m, \sigma_{22}^m, \sigma_{12}^m, \sigma_{23}^m, \sigma_{23}^m$ = 3-D stresses on the fiber misalignment (kinking) plane

$\Lambda_{23}^0 = 2\left(\frac{1}{E_y} - \frac{\upsilon_{xy}\upsilon_{yx}}{E_x}\right)$ = derived from orthotropic material constants

2.4.3.4.8. LaRc04 Matrix Failure Criterion ([403] (p. 943))

$$\xi_{11} = \begin{cases} (1-g)\dfrac{\sigma_y}{Y^t} + g\left(\dfrac{\sigma_y}{Y^t}\right)^2 + \left(\dfrac{\sigma_{xy}}{S^l}\right)^2 + 0.5\Lambda_{23}^0\left(\dfrac{\sigma_{yz}}{S^l}\right)^2 & \text{if } \sigma_y \geq 0 \\ \left(\dfrac{\sigma^t}{S^t - \eta^t\sigma^n}\right)^2 + \left(\dfrac{\sigma^l}{S^l - \eta^l\sigma^n}\right)^2 & \text{if } \sigma_y < 0 \text{ and } \sigma_x > \left|\sigma_{yc}^f\right| \\ \left(\dfrac{\sigma_m^t}{S^t - \eta^t\sigma_m^n}\right)^2 + \left(\dfrac{\sigma_m^l}{S^l - \eta^l\sigma_m^n}\right)^2 & \text{if } \sigma_y < 0 \text{ and } \sigma_x \leq \left|\sigma_{yc}^f\right| \end{cases} \quad (2.102)$$

where:

η^t = transverse friction coefficient

$\sigma^n, \sigma^t, \sigma^l$ = effective normal, transverse, and longitudinal stresses on an action plane parallel to the fibers

$\sigma_m^n, \sigma_m^t, \sigma_m^l$ = effective stresses on the action plane parallel to the misaligned fibers

Chapter 3: Structures with Geometric Nonlinearities

This chapter discusses the various geometrically nonlinear options available, including large strain, large deflection, stress stiffening, pressure load stiffness, and spin softening. Only elements with displacements degrees of freedom (DOFs) are applicable. Not included in this section are multi-status elements (such as COMBIN40, discussed in Element Library (p. 411)) and the eigenvalue buckling capability (discussed in Buckling Analysis (p. 792)).

The following topics are available:

3.1. Understanding Geometric Nonlinearities

Geometric nonlinearities refer to the nonlinearities in the structure or component due to the changing geometry as it deflects. That is, the stiffness [K] is a function of the displacements {u}. The stiffness changes because the shape changes and/or the material rotates. The program can account for four types of geometric nonlinearities:

1. *Large strain* assumes that the strains are no longer infinitesimal (they are finite). Shape changes (e.g. area, thickness, etc.) are also accounted for. Deflections and rotations may be arbitrarily large.

2. *Large rotation* assumes that the rotations are large but the mechanical strains (those that cause stresses) are evaluated using linearized expressions. The structure is assumed not to change shape except for rigid body motions. The elements of this class refer to the original configuration.

3. *Stress stiffening* assumes that both strains and rotations are small. A 1st order approximation to the rotations is used to capture some nonlinear rotation effects.

4. *Spin softening* also assumes that both strains and rotations are small. This option accounts for the coupling between the transverse vibrational motion and the centrifugal force due to an angular velocity.

All elements support the spin softening capability, while only some of the elements support the other options. Please refer to the *Element Reference* for details.

3.2. Large Strain

When the strains in a material exceed more than a few percent, the changing geometry due to this deformation can no longer be neglected. Analyses which include this effect are called large strain, or finite strain, analyses. A large strain analysis is performed in a static (**ANTYPE**,STATIC) or transient (**ANTYPE**,TRANS) analysis while flagging large deformations (**NLGEOM**,ON) when the appropriate element type(s) is used.

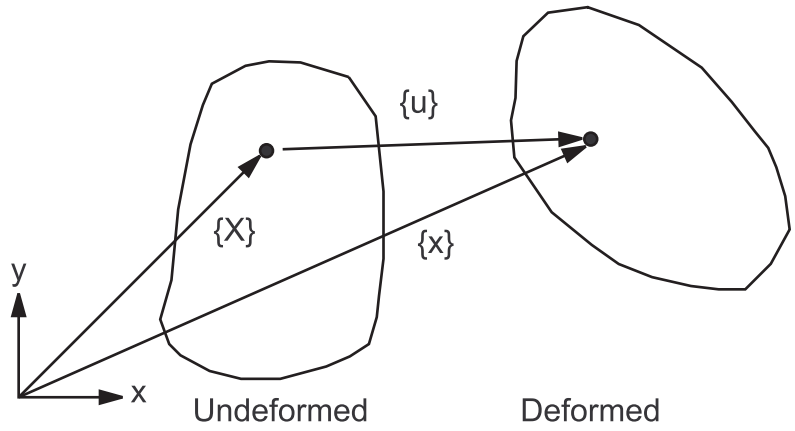
The remainder of this section addresses the large strain formulation for elastic-plastic elements. These elements use a hypoelastic formulation so that they are restricted to small elastic strains (but allow for arbitrarily large plastic strains). Hyperelasticity (p. 124) addresses the large strain formulation for hyperelastic elements, which allow arbitrarily large elastic strains.

3.2.1. Theory

The theory of large strain computations can be addressed by defining a few basic physical quantities (motion and deformation) and the corresponding mathematical relationship. The applied loads acting on a body make it move from one position to another. This motion can be defined by studying a position vector in the "deformed" and "undeformed" configuration. Say the position vectors in the "deformed" and "undeformed" state are represented by {x} and {X} respectively, then the motion (displacement) vector {u} is computed by (see Figure 3.1: Position Vectors and Motion of a Deforming Body (p. 30)):

$$\{u\} = \{x\} - \{X\} \tag{3.1}$$

Figure 3.1: Position Vectors and Motion of a Deforming Body



The deformation gradient is defined as:

$$[F] = \frac{\partial\{x\}}{\partial\{X\}} \tag{3.2}$$

which can be written in terms of the displacement of the point via Equation 3.1 (p. 30) as:

$$[F] = [I] + \frac{\partial\{u\}}{\partial\{X\}} \tag{3.3}$$

where:

[I] = identity matrix

The information contained in the deformation gradient [F] includes the volume change, the rotation and the shape change of the deforming body. The volume change at a point is

$$\frac{dV}{dV_o} = \det[F] \tag{3.4}$$

where:

V_o = original volume
V = current volume
det [•] = determinant of the matrix

The deformation gradient can be separated into a rotation and a shape change using the right polar decomposition theorem:

$$[F] = [R][U] \tag{3.5}$$

where:

[R] = rotation matrix ($[R]^T[R] = [I]$)
[U] = right stretch (shape change) matrix

Once the stretch matrix is known, a logarithmic or Hencky strain measure is defined as:

$$[\varepsilon] = \ell n[U] \tag{3.6}$$

([ε] is in tensor (matrix) form here, as opposed to the usual vector form {ε}). Since [U] is a 2nd order tensor (matrix), Equation 3.6 (p. 31) is determined through the spectral decomposition of [U]:

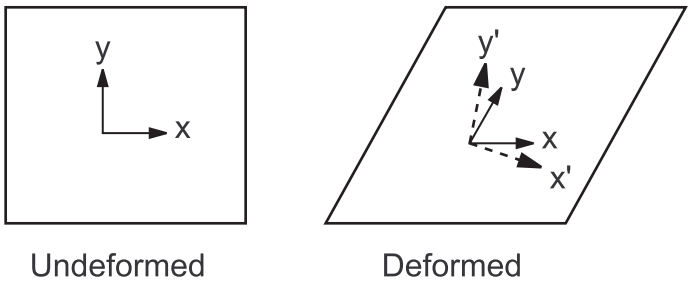
$$[\varepsilon] = \sum_{i=1}^{3} \ell n \lambda_i \{e_i\}\{e_i\}^T \tag{3.7}$$

where:

λ_i = eigenvalues of [U] (principal stretches)
$\{e_i\}$ = eigenvectors of [U] (principal directions)

The polar decomposition theorem (Equation 3.5 (p. 31)) extracts a rotation [R] that represents the average rotation of the material at a point. Material lines initially orthogonal will not, in general, be orthogonal after deformation (because of shearing), see Figure 3.2: Polar Decomposition of a Shearing Deformation (p. 31). The polar decomposition of this deformation, however, will indicate that they will remain orthogonal (lines x-y' in Figure 3.2: Polar Decomposition of a Shearing Deformation (p. 31)). For this reason, non-isotropic behavior (e.g. orthotropic elasticity or kinematic hardening plasticity) should be used with care with large strains, especially if large shearing deformation occurs.

Figure 3.2: Polar Decomposition of a Shearing Deformation



3.2.2. Implementation

Computationally, the evaluation of Equation 3.6 (p. 31) is performed by one of two methods using the incremental approximation (since, in an elastic-plastic analysis, we are using an incremental solution procedure):

$$[\varepsilon] = \int d[e] \approx \sum [D\varepsilon_n] \tag{3.8}$$

with

$$[\Delta\varepsilon_n] = \ell n[\Delta U_n] \tag{3.9}$$

where $[\Delta U_n]$ is the increment of the stretch matrix computed from the incremental deformation gradient:

$$[\Delta F_n] = [\Delta R_n][\Delta U_n] \tag{3.10}$$

where $[\Delta F_n]$ is:

$$[\Delta F_n] = [F_n][F_{n-1}]^{-1} \tag{3.11}$$

$[F_n]$ is the deformation gradient at the current time step and $[F_{n-1}]$ is at the previous time step.

(Hughes([156] (p. 929))) uses the approximate 2nd order accurate calculation for evaluating Equation 3.9 (p. 32):

$$[\Delta\varepsilon_n] = [R_{1/2}]^T [\Delta\varepsilon_n][R_{1/2}] \tag{3.12}$$

where $[R_{1/2}]$ is the rotation matrix computed from the polar decomposition of the deformation gradient evaluated at the midpoint configuration:

$$[F_{1/2}] = [R_{1/2}][U_{1/2}] \tag{3.13}$$

where $[F_{1/2}]$ is (using Equation 3.3 (p. 30)):

$$[F_{1/2}] = [I] + \frac{\partial\{u_{1/2}\}}{\partial\{X\}} \tag{3.14}$$

and the midpoint displacement is:

$$\{u_{1/2}\} = \frac{1}{2}(\{u_n\} + \{u_{n-1}\}) \tag{3.15}$$

$\{u_n\}$ is the current displacement and $\{u_{n-1}\}$ is the displacement at the previous time step. $[\Delta\varepsilon_n]$ is the "rotation-neutralized" strain increment over the time step. The strain increment $[\Delta\tilde{\varepsilon}_n]$ is also computed from the midpoint configuration:

$$\{\Delta\tilde{\varepsilon}_n\} = [B_{1/2}]\{\Delta u_n\} \tag{3.16}$$

$\{\Delta u_n\}$ is the displacement increment over the time step and $[B_{1/2}]$ is the strain-displacement relationship evaluated at the midpoint geometry:

$$\{X_{1/2}\} = \frac{1}{2}(\{X_n\} + \{X_{n-1}\}) \tag{3.17}$$

This method is an excellent approximation to the logarithmic strain if the strain steps are less than ~10%. This method is used by the standard 2-D and 3-D solid and shell elements.

The computed strain increment $[\Delta\varepsilon_n]$ (or equivalently $\{\Delta\varepsilon_n\}$) can then be added to the previous strain $\{\varepsilon_{n-1}\}$ to obtain the current total Hencky strain:

$$\{\varepsilon_n\} = \{\varepsilon_{n-1}\} + \{\Delta\varepsilon_n\} \tag{3.18}$$

This strain can then be used in the stress updating procedures, see Rate-Independent Plasticity (p. 64) and Rate-Dependent Plasticity (Including Creep and Viscoplasticity) (p. 105) for discussions of the rate-independent and rate-dependent procedures respectively.

3.2.3. Definition of Thermal Strains

According to Callen([243] (p. 934)), the coefficient of thermal expansion is defined as the fractional increase in the length per unit increase in the temperature. Mathematically,

$$\alpha = \frac{1}{\ell}\frac{d\ell}{dT} \tag{3.19}$$

where:

α = coefficient of thermal expansion
ℓ = current length
T = temperature

Rearranging Equation 3.19 (p. 33) gives:

$$\frac{d\ell}{\ell} = \alpha dT \tag{3.20}$$

On the other hand, the logarithmic strain is defined as:

$$\varepsilon^{\ell} = \ell n\left(\frac{\ell}{\ell_o}\right) \tag{3.21}$$

where:

ε^{ℓ} = logarithmic strain
ℓ_o = initial length

Differential of Equation 3.21 (p. 33) yields:

$$d\varepsilon^{\ell} = \frac{d\ell}{\ell} \tag{3.22}$$

Comparison of Equation 3.20 (p. 33) and Equation 3.22 (p. 33) gives:

$$d\varepsilon^{\ell} = \alpha dT \tag{3.23}$$

Integration of Equation 3.23 (p. 33) yields:

$$\varepsilon^{\ell} - \varepsilon_o^{\ell} = \alpha(T - T_o) \tag{3.24}$$

where:

ε_o^{ℓ} = initial (reference) strain at temperature T_o
T_o = reference temperature

In the absence of initial strain ($\varepsilon_o^\ell = 0$), then Equation 3.24 (p. 33) reduces to:

$$\varepsilon^\ell = \alpha(T - T_o) \tag{3.25}$$

The thermal strain corresponds to the logarithmic strain. As an example problem, consider a line element of a material with a constant coefficient of thermal expansion α. If the length of the line is ℓ_o at temperature T_o, then the length after the temperature increases to T is:

$$\ell = \ell_o \exp\varepsilon^\ell = \ell_o \exp[\alpha(T - T_o)] \tag{3.26}$$

Now if one interpreted the thermal strain as the engineering (or nominal) strain, then the final length would be different.

$$\varepsilon^e = \alpha(T - T_o) \tag{3.27}$$

where:

ε^e = engineering strain

The final length is then:

$$\ell = \ell_o(1 + \varepsilon^e) = \ell_o[1 + \alpha(T - T_o)] \tag{3.28}$$

However, the difference should be very small as long as:

$$\alpha|T - T_o| \ll 1 \tag{3.29}$$

because

$$\exp[\alpha(T - T_o)] \approx 1 + \alpha(T - T_o) \tag{3.30}$$

3.2.4. Element Formulation

The element matrices and load vectors are derived using an updated Lagrangian formulation. This produces equations of the form:

$$[\bar{K}_i]\Delta u_i = \{F^{app}\} - \{F_i^{nr}\} \tag{3.31}$$

where the tangent matrix $[\bar{K}_i]$ has the form:

$$[\bar{K}_i] = [K_i] + [S_i] \tag{3.32}$$

$[K_i]$ is the usual stiffness matrix:

$$[K_i] = \int [B_i]^T [D_i][B_i] d(vol) \tag{3.33}$$

$[B_i]$ is the strain-displacement matrix in terms of the current geometry $\{X_n\}$ and $[D_i]$ is the current stress-strain matrix.

$[S_i]$ is the stress stiffness (or geometric stiffness) contribution, written symbolically as:

$$[S_i] = \int [G_i]^T [\tau_i][G_i] d(vol) \tag{3.34}$$

where $[G_i]$ is a matrix of shape function derivatives and $[\tau_i]$ is a matrix of the current Cauchy (true) stresses $\{\sigma_i\}$ in the global Cartesian system. The Newton-Raphson restoring force is:

$$[F_i^{nr}] = \int [B_i]^T \{\sigma_i\} d(vol) \tag{3.35}$$

Some of the plane stress and shell elements account for the thickness changes due to the out-of-plane strain ε_z using an approach similar to that of Hughes and Carnoy ([157] (p. 929)). Shells, however, do not update their reference plane (as might be required in a large strain out-of-plane bending deformation); the thickness change is assumed to be constant through the thickness. General element formulations using finite deformation are developed in General Element Formulations (p. 50) and apply to current-technology elements only.

3.2.5. Applicable Input

NLGEOM,ON activates large strain computations in those elements which support it. **NLGEOM**,ON also activates the stress-stiffening contribution to the tangent matrix.

3.2.6. Applicable Output

For elements which have large strain capability, stresses (output as S) are true (Cauchy) stresses in the rotated element coordinate system (the element coordinate system follows the material as it rotates). Strains (output as EPEL, EPPL, etc.) are the logarithmic or Hencky strains, also in the rotated element coordinate system.

An exception is for the hyperelastic elements. For these elements, stress and strain components maintain their original orientations and some of these elements use other strain measures.

3.3. Large Rotation

If the rotations are large but the mechanical strains (those that cause stresses) are small, then a large-rotation procedure can be used.

A large-rotation analysis is performed in a static (**ANTYPE**,STATIC) or transient (**ANTYPE**,TRANS) analysis while flagging large deformations (**NLGEOM**,ON) when the appropriate element type is used.

All large-strain elements also support this capability, as both options account for the large rotations and for small strains, the logarithmic strain measure and the engineering strain measure coincide.

3.3.1. Theory

Large Strain (p. 29) presented the theory for general motion of a material point. Large-rotation theory follows a similar development, except that the logarithmic strain measure (Equation 3.6 (p. 31)) is replaced by the Biot, or small (engineering) strain measure:

$$[\varepsilon] = [U] - [I] \tag{3.36}$$

where:

[U] = stretch matrix
[I] = 3 x 3 identity matrix

3.3.2. Implementation

A corotational (or convected coordinate) approach is used in solving large-rotation/small-strain problems (Rankin and Brogan([66] (p. 924))). "Corotational" may be thought of as "rotated with". The nonlinearities are contained in the strain-displacement relationship which for this algorithm takes on the special form:

$$[B_n] = [B_v][T_n] \tag{3.37}$$

where:

$[B_v]$ = usual small strain-displacement relationship in the original (virgin) element coordinate system
$[T_n]$ = orthogonal transformation relating the original element coordinates to the convected (or rotated) element coordinates

The convected element coordinate frame differs from the original element coordinate frame by the amount of rigid body rotation. Hence $[T_n]$ is computed by separating the rigid body rotation from the total deformation $\{u_n\}$ using the polar decomposition theorem, Equation 3.5 (p. 31). From Equation 3.37 (p. 36), the element tangent stiffness matrix has the form:

$$[K_e] = \int_{vol} [T_n]^T [B_v]^T [D][B_v][T_n] d(vol) \tag{3.38}$$

and the element restoring force is:

$$\{F_e^{nr}\} = \int_{vol} [T_n]^T [B_v]^T [D]\{\varepsilon_n^{el}\} d(vol) \tag{3.39}$$

where the elastic strain is computed from:

$$\{\varepsilon_n^{el}\} = [B_v]\{u_d^n\} \tag{3.40}$$

$\{u_n^d\}$ is the element deformation which causes straining as described in a subsequent subsection.

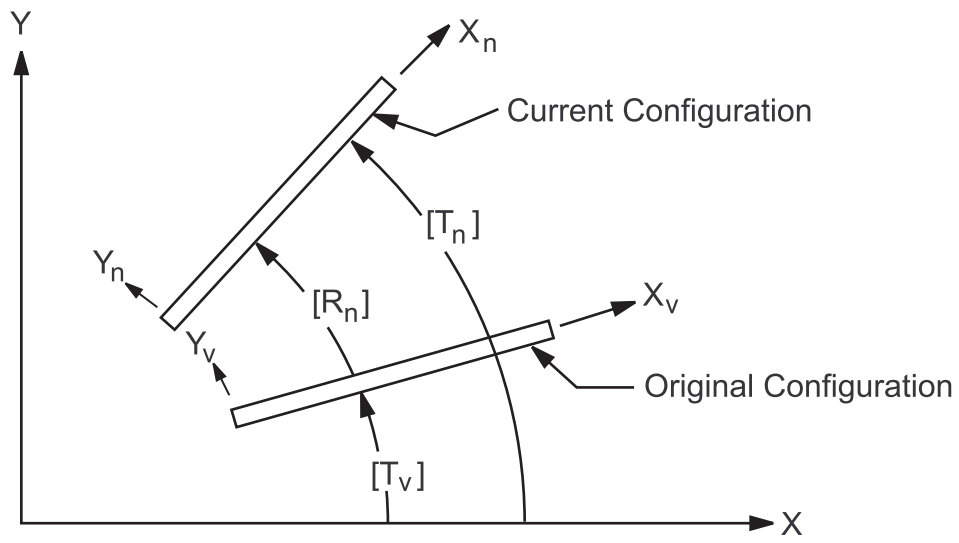
The large-rotation process can be summarized as a three step process for each element:

1. Determine the updated transformation matrix $[T_n]$ for the element.
2. Extract the deformational displacement $\{u_n^d\}$ from the total element displacement $\{u_n\}$ for computing the stresses as well as the restoring force $\{F_e^{nr}\}$.
3. After the rotational increments in $\{\Delta u\}$ are computed, update the node rotations appropriately. All three steps require the concept of a rotational pseudovector in order to be efficiently implemented (Rankin and Brogan([66] (p. 924)), Argyris([67] (p. 924))).

3.3.3. Element Transformation

The updated transformation matrix $[T_n]$ relates the current element coordinate system to the global Cartesian coordinate system as shown in Figure 3.3: Element Transformation Definitions (p. 37).

Figure 3.3: Element Transformation Definitions



$[T_n]$ can be computed directly or the rotation of the element coordinate system $[R_n]$ can be computed and related to $[T_n]$ by

$$[T_n] = [T_v][R_n] \tag{3.41}$$

where $[T_v]$ is the original transformation matrix. The determination of $[T_n]$ is unique to the type of element involved, whether it is a solid element, shell element, beam element, or spar element.

Solid Elements. The rotation matrix $[R_n]$ for these elements is extracted from the displacement field using the deformation gradient coupled with the polar decomposition theorem (see Malvern([87] (p. 925))).

Shell Elements. The updated normal direction (element z direction) is computed directly from the updated coordinates. The computation of the element normal is given in Element Library (p. 411) for each particular shell element. The extraction procedure outlined for solid elements is used coupled with the information on the normal direction to compute the rotation matrix $[R_n]$.

Beam Elements. The nodal rotation increments from $\{\Delta u\}$ are averaged to determine the average rotation of the element. The updated average element rotation and then the rotation matrix $[R_n]$ is computed using Rankin and Brogan([66] (p. 924)). In special cases where the average rotation of the element computed in the above way differs significantly from the average rotation of the element computed from nodal translations, the quality of the results will be degraded.

Link Elements. The updated transformation $[T_n]$ is computed directly from the updated coordinates.

Generalized Mass Element (MASS21). The nodal rotation increment from $\{\Delta u\}$ is used to update the element rotation which then yields the rotation matrix $[R_n]$.

3.3.4. Deformational Displacements

The displacement field can be decomposed into a rigid body translation, a rigid body rotation, and a component which causes strains:

$$\{u\} = \{u^r\} + \{u^d\} \tag{3.42}$$

where:

$\{u^r\}$ = rigid body motion
$\{u^d\}$ = deformational displacements which cause strains

$\{u^d\}$ contains both translational as well as rotational DOF.

The translational component of the deformational displacement can be extracted from the displacement field by

$$\{u_t^d\} = [R_n](\{x_V\} + \{u\}) - \{x_V\} \tag{3.43}$$

where:

$\{u_t^d\}$ = translational component of the deformational displacement
$[R_n]$ = current element rotation matrix
$\{x_v\}$ = original element coordinates in the global coordinate system
$\{u\}$ = element displacement vector in global coordinates

$\{u^d\}$ is in the global coordinate system.

For elements with rotational DOFs, the rotational components of the deformational displacement must be computed. The rotational components are extracted by essentially "subtracting" the nodal rotations $\{u\}$ from the element rotation given by $\{u^r\}$. In terms of the pseudovectors this operation is performed as follows for each node:

1. Compute a transformation matrix from the nodal pseudovector $\{\theta^n\}$ yielding $[T_n]$.

2. Compute the relative rotation $[T^d]$ between $[R_n]$ and $[T_n]$:

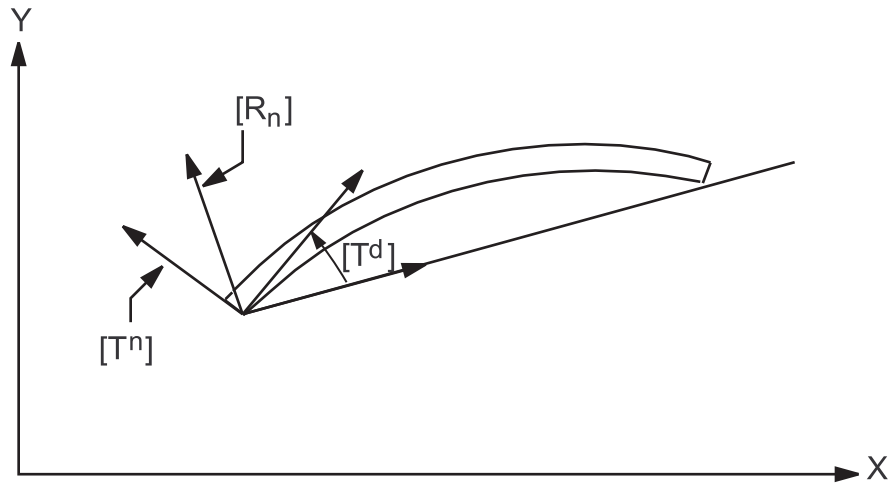
   $$[T^d] = [R_n][T_n]^T \tag{3.44}$$

   This relative rotation contains the rotational deformations of that node as shown in Figure 3.4: Definition of Deformational Rotations (p. 39).

3. Extract the nodal rotational deformations $\{u^d\}$ from $[T^d]$.

Because of the definition of the pseudovector, the deformational rotations extracted in step 3 are limited to less than 30°, since $2\sin(\theta/2)$ no longer approximates θ itself above 30°. This limitation only applies to the rotational distortion (i.e., bending) within a single element.

Figure 3.4: Definition of Deformational Rotations



3.3.5. Updating Rotations

Once the transformation [T] and deformational displacements $\{u^d\}$ are determined, the element matrices Equation 3.38 (p. 36) and restoring force Equation 3.39 (p. 36) can be determined. The solution of the system of equations yields a displacement increment $\{\Delta u\}$. The nodal rotations at the element level are updated with the rotational components of $\{\Delta u\}$. The global rotations (in the output and on the results file) are not updated with the pseudovector approach, but are simply added to the previous rotation in $\{u_{n-1}\}$.

3.3.6. Applicable Input

The large-rotation computations in those elements which support it are activated by the large-deformation key (**NLGEOM**,ON). Stress stiffening is also be included and contributes to the tangent stiffness matrix (which may be required for structures weak in bending resistance).

3.3.7. Applicable Output

Stresses (output as S) are engineering stresses in the rotated element coordinate system (the element coordinate system follows the material as it rotates). Strains (output as EPEL, EPPL, etc.) are engineering strains, also in the rotated element coordinate system. This applies to element types that do not have large-strain capability. For element types that have large-strain capability, see Large Strain (p. 29).

3.3.8. Consistent Tangent Stiffness Matrix and Finite Rotation

It has been found in many situations that the use of consistent tangent stiffness in a nonlinear analysis can speed up the rate of convergence greatly. It normally results in a quadratic rate of convergence. A consistent tangent stiffness matrix is derived from the discretized finite element equilibrium equations without the introduction of various approximations. The terminology of finite rotation in the context of geometrical nonlinearity implies that rotations can be arbitrarily large and can be updated accurately. A consistent tangent stiffness accounting for finite rotations derived by Nour-Omid and Rankin([175] (p. 930)) for beam/shell elements is used. The technology of consistent tangent matrix and finite rotation makes the buckling and postbuckling analysis a relatively easy task. The theory of finite rotation representation and update has been described in Large Rotation (p. 35) using a pseudovector

representation. The following will outline the derivations of a consistent tangent stiffness matrix used for the corotational approach.

The nonlinear static finite element equations solved can be characterized by at the element level by:

$$\sum_{e=1}^{N} ([T_n]^T \{F_e^{int}\} - \{F_e^a\}) = 0 \tag{3.45}$$

where:

N = number of total elements

$\{F_e^{int}\}$ = element internal force vector in the element coordinate system, generally see Equation 3.46 (p. 40)

$[T_n]^T$ = transform matrix transferring the local internal force vector into the global coordinate system

$\{F_e^a\}$ = applied load vector at the element level in the global coordinate system

$$\{F_e^{int}\} = \int [B_v]^T \{\sigma_e\} d(vol) \tag{3.46}$$

Hereafter, we shall focus on the derivation of the consistent tangent matrix at the element level without introducing an approximation. The consistent tangent matrix is obtained by differentiating Equation 3.45 (p. 40) with respect to displacement variables $\{u_e\}$:

$$\begin{aligned}[K_e^T]_{consistent} &= [T_n]^T \frac{\partial \{F_e^{int}\}}{\partial \{u_e\}} + \frac{\partial [T_n]^T}{\partial \{u_e\}} \{F_e^{int}\} \\ &= [T_n]^T \int_e [B_v]^T \frac{\partial \{\sigma_e\}}{\partial \{u_e\}} d(vol) + [T_n]^T \int_e \frac{\partial [B_v]^T}{\partial \{u_e\}} \{\sigma_e\} d(vol) \\ &\qquad \text{I} \qquad\qquad\qquad\qquad\qquad\qquad \text{II} \\ &\quad + \frac{\partial [T_v]^T}{\partial \{u_e\}} \{F_e^{int}\} \\ &\qquad \text{III}\end{aligned} \tag{3.47}$$

It can be seen that Part I is the main tangent matrix Equation 3.38 (p. 36) and Part II is the stress stiffening matrix (Equation 3.34 (p. 34), Equation 3.61 (p. 45) or Equation 3.64 (p. 45)). Part III is another part of the stress stiffening matrix (see Nour-Omid and Rankin([175] (p. 930))) traditionally neglected in the past. However, many numerical experiments have shown that Part III of $[K_e^T]$ is essential to the faster rate of convergence. In some cases, Part III of $[K_e^T]$ is unsymmetric; when this occurs, a procedure of symmetrizing $[K_e^T]$ is invoked.

As Part III of the consistent tangent matrix utilizes the internal force vector $\{F_e^{int}\}$ to form the matrix, it is required that the internal vector $\{F_e^{int}\}$ not be so large as to dominate the main tangent matrix

(Part I). This can normally be guaranteed if the realistic material and geometry are used, that is, the element is not used as a rigid link and the actual thicknesses are input.

It is also noted that the consistent tangent matrix Equation 3.47 (p. 40) is very suitable for use with the arc-length solution method.

3.4. Stress Stiffening

3.4.1. Overview and Usage

Stress stiffening (also called geometric stiffening, incremental stiffening, initial stress stiffening, or differential stiffening by other authors) is the stiffening (or weakening) of a structure due to its stress state. This stiffening effect normally needs to be considered for thin structures with bending stiffness very small compared to axial stiffness, such as cables, thin beams, and shells and couples the in-plane and transverse displacements. This effect also augments the regular nonlinear stiffness matrix produced by large-strain or large-deflection effects (**NLGEOM**,ON). The effect of stress stiffening is accounted for by generating and then using an additional stiffness matrix, hereinafter called the "stress stiffness matrix". The stress stiffness matrix is added to the regular stiffness matrix in order to give the total stiffness. Stress stiffening may be used for static (**ANTYPE**,STATIC) or transient (**ANTYPE**,TRANS) analyses. Working with the stress stiffness matrix is the pressure load stiffness, discussed in Pressure Load Stiffness (p. 46).

The stress stiffness matrix is computed based on the stress state of the previous equilibrium iteration. Thus, to generate a valid stress-stiffened problem, at least two iterations are normally required, with the first iteration being used to determine the stress state that will be used to generate the stress stiffness matrix of the second iteration. If this additional stiffness affects the stresses, more iterations need to be done to obtain a converged solution.

In some linear analyses, the static (or initial) stress state may be large enough that the additional stiffness effects must be included for accuracy. Modal (**ANTYPE**,MODAL) and substructure (**ANTYPE**,SUBSTR) analyses are linear analyses for which the prestressing effects can be requested to be included (**PSTRES**,ON command). Note that in these cases the stress stiffness matrix is constant, so that the stresses computed in the analysis are assumed small compared to the prestress stress.

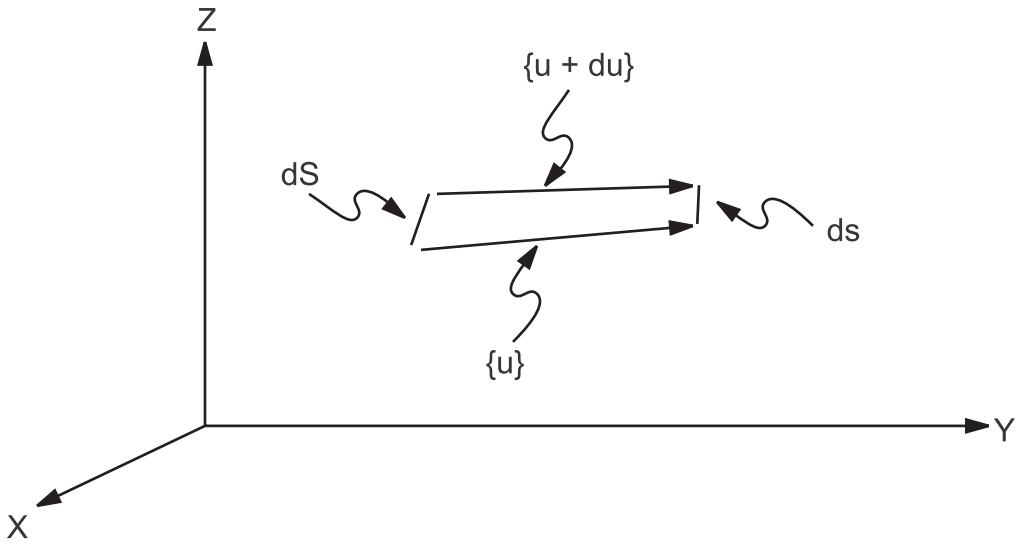
If membrane stresses should become compressive rather than tensile, then terms in the stress stiffness matrix may "cancel" the positive terms in the regular stiffness matrix and therefore yield a nonpositive-definite total stiffness matrix, which indicates the onset of buckling. If this happens, it is indicated with the message: *"Large negative pivot value ___, at node ___ may be because buckling load has been exceeded"*. It must be noted that a stress stiffened model with insufficient boundary conditions to prevent rigid body motion may yield the same message.

The linear buckling load can be calculated directly by adding an unknown multiplier of the stress stiffness matrix to the regular stiffness matrix and performing an eigenvalue buckling problem (**ANTYPE**,BUCKLE) to calculate the value of the unknown multiplier. This is discussed in more detail in Buckling Analysis (p. 792).

3.4.2. Theory

The strain-displacement equations for the general motion of a differential length fiber are derived below. Two different results have been obtained and these are both discussed below. Consider the motion of a differential fiber, originally at dS, and then at ds after deformation.

Figure 3.5: General Motion of a Fiber



One end moves *{u}*, and the other end moves *{u + du}*, as shown in Figure 3.5: General Motion of a Fiber (p. 42). The motion of one end with the rigid body translation removed is *{u + du} - {u} = {du}*. *{du}* may be expanded as

$$\{du\} = \begin{Bmatrix} du \\ dv \\ dw \end{Bmatrix} \tag{3.48}$$

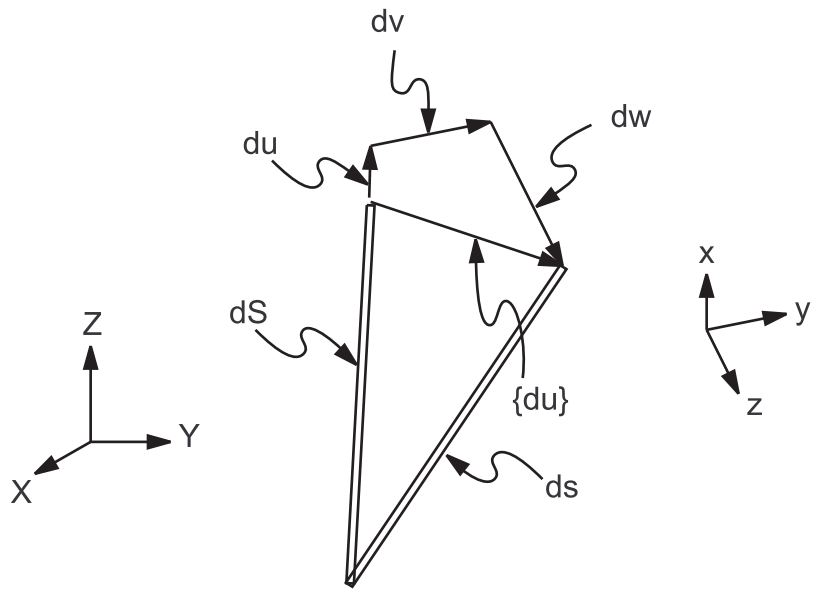
where u is the displacement parallel to the original orientation of the fiber. This is shown in Figure 3.6: Motion of a Fiber with Rigid Body Motion Removed (p. 43). Note that X, Y, and Z represent global Cartesian axes, and x, y, and z represent axes based on the original orientation of the fiber. By the Pythagorean theorem,

$$ds = \sqrt{(dS + du)^2 + (dv)^2 + (dw)^2} \tag{3.49}$$

The stretch, Λ, is given by dividing ds by the original length dS:

$$\Lambda = \frac{ds}{dS} = \sqrt{\left(1 + \frac{du}{dS}\right)^2 + \left(\frac{dv}{dS}\right)^2 + \left(\frac{dw}{dS}\right)^2} \tag{3.50}$$

Figure 3.6: Motion of a Fiber with Rigid Body Motion Removed



As dS is along the local x axis,

$$\Lambda = \sqrt{\left(1+\frac{du}{dx}\right)^2 + \left(\frac{dv}{dx}\right)^2 + \left(\frac{dw}{dx}\right)^2} \tag{3.51}$$

Next, Λ is expanded and converted to partial notation:

$$\Lambda = \sqrt{1+2\frac{\partial u}{\partial x} + \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2} \tag{3.52}$$

The binominal theorem states that:

$$\sqrt{1+A} = 1+\frac{A}{2} - \frac{A^2}{8} + \frac{A^3}{16}... \tag{3.53}$$

when $A^2 < 1$. One should be aware that using a limited number of terms of this series may restrict its applicability to small rotations and small strains. If the first *two* terms of the series in Equation 3.53 (p. 43) are used to expand Equation 3.52 (p. 43),

$$\Lambda = 1+\frac{\partial u}{\partial x} + \frac{1}{2}\left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right) \tag{3.54}$$

The resultant strain (same as extension since strains are assumed to be small) is then

$$\varepsilon_x = \Lambda - 1 = \frac{\partial u}{\partial x} + \frac{1}{2}\left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right) \tag{3.55}$$

If, more accurately, the first *three* terms of Equation 3.53 (p. 43) are used and displacement derivatives of the third order and above are dropped, Equation 3.53 (p. 43) reduces to:

$$\Lambda = 1 + \frac{\partial u}{\partial x} + \frac{1}{2}\left(\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right) \tag{3.56}$$

The resultant strain is:

$$\varepsilon_x = \Lambda - 1 = \frac{\partial u}{\partial x} + \frac{1}{2}\left(\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right) \tag{3.57}$$

For most 2-D and 3-D elements, Equation 3.55 (p. 43) is more convenient to use as no account of the loaded direction has to be considered. The error associated with this is small as the strains were assumed to be small. For 1-D structures, and some 2-D elements, Equation 3.57 (p. 44) is used for its greater accuracy and causes no difficulty in its implementation.

3.4.3. Implementation

The stress-stiffness matrices are derived based on Equation 3.34 (p. 34), but using the nonlinear strain-displacement relationships given in Equation 3.55 (p. 43) or Equation 3.57 (p. 44) (Cook([5] (p. 921))).

For a spar, the stress-stiffness matrix is given as:

$$[S_\ell] = \frac{F}{L}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \tag{3.58}$$

The stress stiffness matrix for a 2-D beam is given in Equation 3.59 (p. 44), which is the same as reported by Przemieniecki([28] (p. 922)). All beam and straight pipe elements use the same type of matrix. Forces used by straight pipe elements are based on not only the effect of axial stress with pipe wall, but also internal and external pressures on the "end-caps" of each element. This force is sometimes referred to as effective tension.

$$[S_\ell] = \frac{F}{L}\begin{bmatrix} 0 & & & & & \\ 0 & \frac{6}{5} & & & \text{Symmetric} & \\ 0 & \frac{1}{10}L & \frac{2}{15}L^2 & & & \\ 0 & 0 & 0 & 0 & & \\ 0 & -\frac{6}{5} & -\frac{1}{10}L & 0 & \frac{6}{5} & \\ 0 & \frac{1}{10}L & -\frac{1}{30}L^2 & 0 & -\frac{1}{10}L & \frac{2}{15}L^2 \end{bmatrix} \tag{3.59}$$

where:

F = force in member
L = length of member

The stress stiffness matrix for 2-D and 3-D solid elements is generated by the use of numerical integration. A 3-D solid element (SOLID185) is used here as an example:

$$[S_\ell] = \begin{bmatrix} [S_o] & 0 & 0 \\ 0 & [S_o] & 0 \\ 0 & 0 & [S_o] \end{bmatrix} \tag{3.60}$$

where the matrices shown in Equation 3.60 (p. 45) have been reordered so that first all x-direction DOF are given, then y, and then z. $[S_o]$ is an 8 by 8 matrix given by:

$$[S_o] = \int_{vol} [S_g]^T [S_m][S_g] d(vol) \tag{3.61}$$

The matrices used by this equation are:

$$[S_m] = \begin{bmatrix} \sigma_x & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z \end{bmatrix} \tag{3.62}$$

where σ_x, σ_{xy} etc. are stress based on the displacements of the previous iteration, and,

$$[S_g] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & .... & \frac{\partial N_8}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & .... & \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & .... & \frac{\partial N_8}{\partial z} \end{bmatrix} \tag{3.63}$$

where N_i represents the ith shape function. This is the stress stiffness matrix for small strain analyses. For large-strain elements in a large-strain analysis (**NLGEOM**,ON), the stress stiffening contribution is computed using the actual strain-displacement relationship (Equation 3.6 (p. 31)).

One further case requires some explanation: axisymmetric structures with nonaxisymmetric deformations. As any stiffening effects may only be axisymmetric, only axisymmetric cases are used for the prestress case. Axisymmetric cases are defined as ℓ (input as MODE on **MODE** command) = 0. Then, any subsequent load steps with any value of ℓ (including 0 itself) uses that same stress state, until another, more recent, ℓ = 0 case is available. Also, torsional stresses are not incorporated into any stress stiffening effects.

Specializing this to SHELL61 (Axisymmetric-Harmonic Structural Shell), only two stresses are used for prestressing: σ_s, σ_θ, the meridional and hoop stresses, respectively. The element stress stiffness matrix is:

$$[S_\ell] = \int_{vol} [S_g]^T [S_m][S_g] d(vol) \tag{3.64}$$

$$[S_m] = \begin{bmatrix} \sigma_s & 0 & 0 & 0 \\ 0 & \sigma_s & 0 & 0 \\ 0 & 0 & \sigma_\theta & 0 \\ 0 & 0 & 0 & \sigma_\theta \end{bmatrix} \tag{3.65}$$

$$[S_g] = [A_s][N]$$

where $[A_s]$ is defined below and [N] is defined by the element shape functions. $[A_s]$ is an operator matrix and its terms are:

$$[A_s] = \begin{bmatrix} 0 & 0 & \frac{\partial}{\partial s} \\ 0 & C\left(-\frac{\partial}{\partial s} - \frac{\sin\theta}{R}\right) & 0 \\ 0 & C\frac{\cos\theta}{R} & \frac{\partial}{R\partial\theta} \\ -\frac{\partial}{R\partial\theta} & 0 & 0 \end{bmatrix} \tag{3.66}$$

where:

$$C = \begin{cases} 0.0 \text{ if } \ell = 0 \\ 1.0 \text{ if } \ell > 0 \end{cases}$$

The three columns of the $[A_s]$ matrix refer to u, v, and w motions, respectively. As suggested by the definition for $[S_m]$, the first two rows of $[A_s]$ relate to σ_s and the second two rows relate to σ_θ. The first row of $[A_s]$ is for motion normal to the shell varying in the s direction and the second row is for hoop motions varying in the s direction. Similarly, the third row is for normal motions varying in the hoop direction. Thus Equation 3.57 (p. 44), rather than Equation 3.55 (p. 43), is the type of nonlinear strain-displacement expression that has been used to develop Equation 3.66 (p. 46).

3.4.4. Pressure Load Stiffness

Quite often concentrated forces are treated numerically by equivalent pressure over a known area. This is especially common in the context of a linear static analysis. However, it is possible that different buckling loads may be predicted from seemingly equivalent pressure and force loads in an eigenvalue buckling analysis. The difference can be attributed to the fact that pressure is considered as a "follower" load. The force on the surface depends on the prescribed pressure magnitude and also on the surface orientation. Concentrated loads are not considered as follower loads. The follower effects is a preload stiffness and plays a significant role in nonlinear and eigenvalue buckling analysis. The follower effects manifest in the form of a "load stiffness matrix" in addition to the normal stress stiffening effects. As with any numerical analysis, it is recommended to use the type of loading which best models the in-service component.

The effect of change of direction and/or area of an applied pressure is responsible for the pressure load stiffness matrix ($[S^{pr}]$) (see section 6.5.2 of Bonet and Wood([236] (p. 934))). It is used either for a large-deflection analysis (**NLGEOM**,ON), for an eigenvalue buckling analysis, or for a modal, linear transient, or harmonic analysis that has prestressing flagged (**PSTRES**,ON command).

The need of $[S^{pr}]$ is most dramatically seen when modelling the collapse of a ring due to external pressure using eigenvalue buckling. The expected answer is:

$$P_{cr} = \frac{CEI}{R^3} \tag{3.67}$$

where:

P_{cr} = critical buckling load
E = Young's modulus
I = moment of inertia
R = radius of the ring
C = 3.0

This value of C = 3.0 is achieved when using $[S^{pr}]$, but when it is missing, C = 4.0, a 33% error.

For eigenvalue buckling analyses, all elements with pressure load stiffness capability use that capability. Otherwise, its use is controlled by *KEY3* on the **SOLCONTROL** command.

$[S^{pr}]$ is derived as an unsymmetric matrix. Symmetricizing is done, unless the command **NROPT**,UNSYM is used. Processing unsymmetric matrices takes more running time and storage, but may be more convergent.

3.4.5. Applicable Input

In a nonlinear analysis (**ANTYPE**,STATIC or **ANTYPE**,TRANS), the stress stiffness contribution is activated and then added to the stiffness matrix. When not using large deformations (**NLGEOM**,OFF), the rotations are presumed to be small and the additional stiffness induced by the stress state is included. When using large deformations (**NLGEOM**,ON), the stress stiffness augments the tangent matrix, affecting the rate of convergence but not the final converged solution.

The stress stiffness contribution in the prestressed analysis is activated by the prestress flag (**PSTRES**,ON) and directs the preceding analysis to save the stress state.

3.4.6. Applicable Output

In a small deflection/small strain analysis (**NLGEOM**,OFF), the 2-D and 3-D elements compute their strains using Equation 3.55 (p. 43). The strains (output as EPEL, EPPL, etc.) therefore include the higher-order terms (e.g. $\frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2$ in the strain computation. Also, nodal and reaction loads (output quantities F and M) will reflect the stress stiffness contribution, so that moment and force equilibrium include the higher order (small rotation) effects.

3.5. Spin Softening

The vibration of a spinning body will cause relative circumferential motions, which will change the direction of the centrifugal load which, in turn, will tend to destabilize the structure. As a small deflection analysis cannot directly account for changes in geometry, the effect can be accounted for by an adjustment of the stiffness matrix, called spin softening. When large deformations are active (**NLGEOM**,ON), or pres-stress effects are active (**PSTRES**,ON), the spin softening contribution is automatically included as an additional contribution to the tangent matrix (Equation 3.32 (p. 34)).

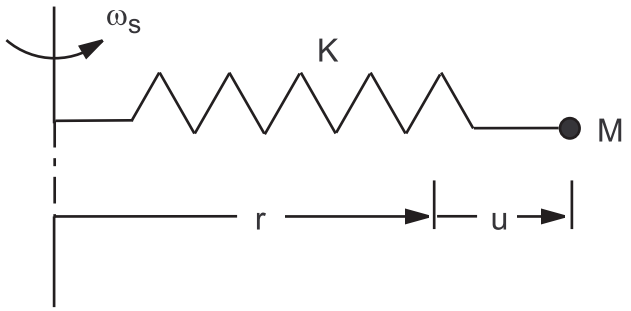
Consider a simple spring-mass system, with the spring oriented radially with respect to the axis of rotation, as shown in Figure 3.7: Spinning Spring-Mass System (p. 48). Equilibrium of the spring and centrifugal forces on the mass using small deflection logic requires:

$$Ku = \omega_s^2 Mr \tag{3.68}$$

where:

u = radial displacement of the mass from the rest position
r = radial rest position of the mass with respect to the axis of rotation
ω_s = angular velocity of rotation

Figure 3.7: Spinning Spring-Mass System



However, to account for large-deflection effects, Equation 3.68 (p. 47) must be expanded to:

$$Ku = \omega_s^2 M(r + u) \tag{3.69}$$

Rearranging terms,

$$(K - \omega_s^2 M)u = \omega_s^2 Mr \tag{3.70}$$

Defining:

$$\bar{K} = K - \omega_s^2 M \tag{3.71}$$

and

$$\bar{F} = \omega_s^2 Mr \tag{3.72}$$

Equation 3.70 (p. 48) becomes simply,

$$\bar{K}u = \bar{F} \tag{3.73}$$

$\bar{K}$ is the stiffness needed in a small deflection solution to account for large-deflection effects. $\bar{F}$ is the same as that derived from small deflection logic. This decrease in the effective stiffness matrix is called spin (or centrifugal) softening. See also Carnegie([104] (p. 926)) for additional development.

Extension of Equation 3.71 (p. 48) into three dimensions is illustrated for a single noded element here:

$$\bar{K} = K + \Omega^2 M \tag{3.74}$$

with

$$\Omega^2 = \begin{bmatrix} -(\omega_y^2 + \omega_z^2) & \omega_x\omega_y & \omega_x\omega_z \\ \omega_x\omega_y & -(\omega_x^2 + \omega_z^2) & \omega_y\omega_z \\ \omega_x\omega_z & \omega_y\omega_z & -(\omega_x^2 + \omega_y^2) \end{bmatrix} \tag{3.75}$$

where:

ω_x, ω_y, ω_z = x, y, and z components of the angular velocity (input with **OMEGA** or **CMOMEGA** command)

It can be seen from Equation 3.74 (p. 48)and Equation 3.75 (p. 48) that if there are more than one non-zero component of angular velocity of rotation, the stiffness matrix may become unsymmetric. For example, for a diagonal mass matrix with a different mass in each direction, the $\bar{K}$ matrix becomes non-symmetric with the expression in Equation 3.74 (p. 48) expanded as:

$$\bar{K}_{xx} = K_{xx} - (\omega_y^2 + \omega_z^2)M_{xx} \tag{3.76}$$

$$\bar{K}_{yy} = K_{yy} - (\omega_x^2 + \omega_z^2)M_{yy} \tag{3.77}$$

$$\bar{K}_{zz} = K_{zz} - (\omega_x^2 + \omega_y^2)M_{zz} \tag{3.78}$$

$$\bar{K}_{xy} = K_{xy} + \omega_x\omega_y M_{yy} \tag{3.79}$$

$$\bar{K}_{yx} = K_{yx} + \omega_x\omega_y M_{xx} \tag{3.80}$$

$$\bar{K}_{xz} = K_{xz} + \omega_x\omega_z M_{zz} \tag{3.81}$$

$$\bar{K}_{zx} = K_{zx} + \omega_x\omega_z M_{xx} \tag{3.82}$$

$$\bar{K}_{yz} = K_{yz} + \omega_y\omega_z M_{zz} \tag{3.83}$$

$$\bar{K}_{zy} = K_{zy} + \omega_y\omega_z M_{yy} \tag{3.84}$$

where:

K_{xx}, K_{yy}, K_{zz} = x, y, and z components of stiffness matrix as computed by the element

Kxy, Kyx, Kxz, Kzx, Kyz, Kzy = off-diagonal components of stiffness matrix as computed by the element

$\bar{K}_{xx}, \bar{K}_{yy}, \bar{K}_{zz}$ = x, y, and z components of stiffness matrix adjusted for spin softening

M_{xx}, M_{yy}, M_{zz} = x, y, and z components of mass matrix

$\bar{K}_{xy}, \bar{K}_{yx}, \bar{K}_{xz}, \bar{K}_{zx}, \bar{K}_{yz}, \bar{K}_{zy}$ = off-diagonal components of stiffness matrix adjusted for spin softening

From Equation 3.76 (p. 49) thru Equation 3.84 (p. 49), it may be seen that there are spin softening effects only in the plane of rotation, not normal to the plane of rotation. Using the example of a modal analysis, Equation 3.71 (p. 48) can be combined with Equation 15.49 (p. 779) to give:

$$\left| [\bar{K}] - \omega^2 [M] \right| = 0 \tag{3.85}$$

or

$$\left| ([K] - \omega_s^2 [M]) - \omega^2 [M] \right| = 0 \tag{3.86}$$

where:

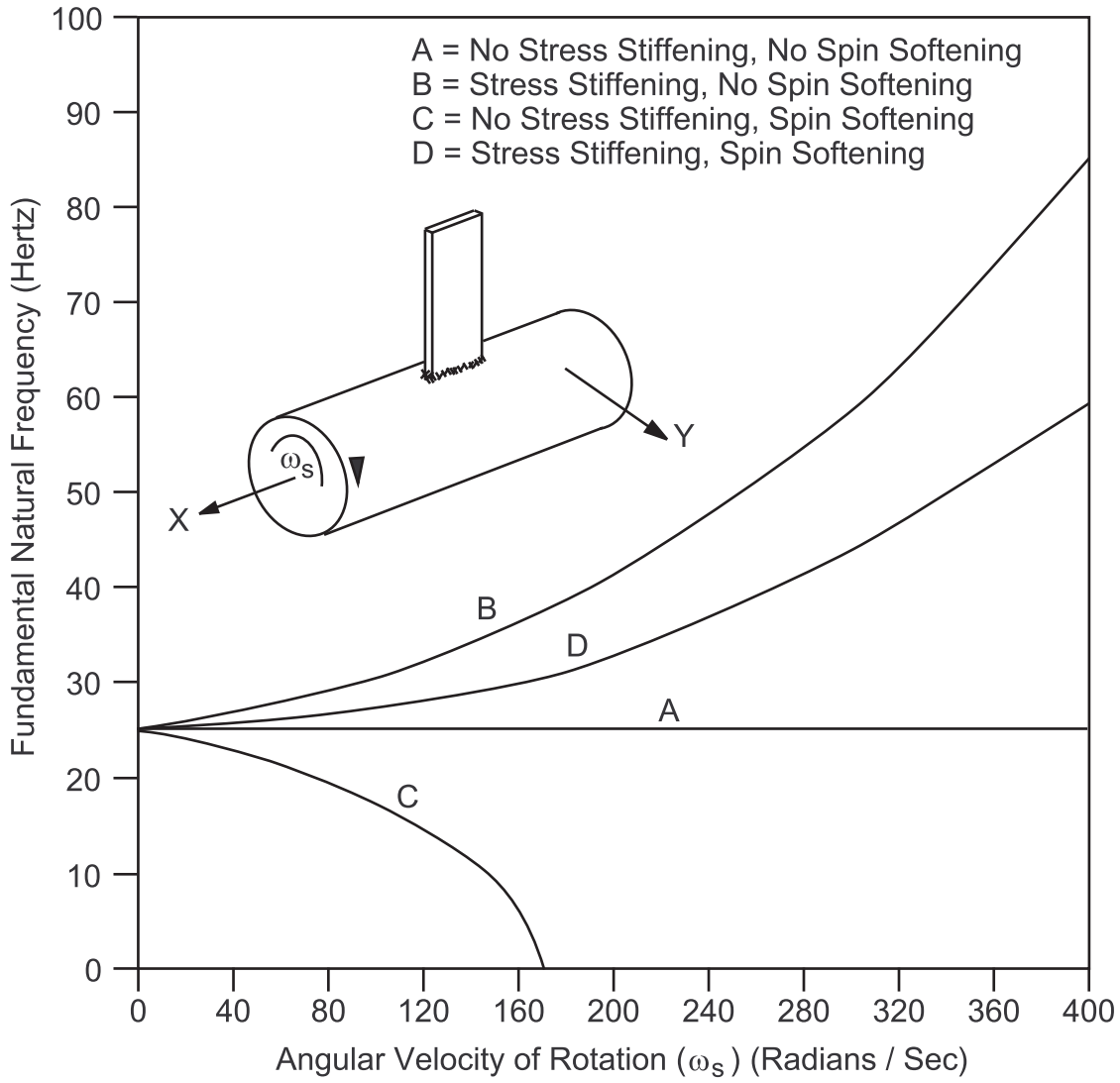
ω = the natural circular frequencies of the rotating body.

If stress stiffening is added to Equation 3.86 (p. 49), the resulting equation is:

$$\left| ([K] + [S] - \omega_s^2 [M]) - \omega^2 [M] \right| = 0 \tag{3.87}$$

Stress stiffening is normally applied whenever spin softening is activated, even though they are independent theoretically. The modal analysis of a thin fan blade is shown in Figure 3.8: Effects of Spin Softening and Stress Stiffening (p. 50).

Figure 3.8: Effects of Spin Softening and Stress Stiffening



On Fan Blade Natural Frequencies

3.6. General Element Formulations

Element formulations developed in this section are applicable for general finite strain deformation. Naturally, they are applicable to small deformations, small deformation with large rotations, and stress stiffening as particular cases. The formulations are based on the principle of virtual work. Minimal assumptions are used in arriving at the slope of nonlinear force-displacement relationship, i.e., the element tangent stiffness. Hence, they are also called consistent formulations. These formulations have been implemented in PLANE182, PLANE183, SOLID185, and SOLID186. SOLID187, SOLID272, SOLID273, SOL-

ID285, SOLSH190, LINK180, SHELL181, BEAM188, BEAM189, SHELL208, SHELL209, REINF264, REINF265, SHELL281, PIPE288, PIPE289, and ELBOW290 are further specializations of the general theory.

In this section, the convention of index notation will be used. For example, repeated subscripts imply summation on the possible range of the subscript, usually the space dimension, so that $\sigma_{ii} = \sigma_{11} + \sigma_{22} + \sigma_{33}$, where 1, 2, and 3 refer to the three coordinate axes x_1, x_2, and x_3, otherwise called x, y, and z.

3.6.1. Fundamental Equations

General finite strain deformation has the following characteristics:

- Geometry changes during deformation. The deformed domain at a particular time is generally different from the undeformed domain and the domain at any other time.
- Strain is no longer infinitesimal so that a large-strain definition has to be employed.
- Cauchy stress cannot be updated simply by adding its increment. It has to be updated by a particular algorithm in order to take into account the finite deformation.
- Incremental analysis is necessary to simulate the nonlinear behaviors.

The updated Lagrangian method is applied to simulate geometric nonlinearities (accessed with **NLGEOM**,ON). Assuming all variables, such as coordinates x_i, displacements u_i, strains ε_{ij}, stresses σ_{ij}, velocities v_i, volume V and other material variables have been solved for and are known at time t; one solves for a set of linearized simultaneous equations having displacements (and hydrostatic pressures in the mixed u-P formulation) as primary unknowns to obtain the solution at time $t + \Delta t$. These simultaneous equations are derived from the element formulations which are based on the principle of virtual work:

$$\int_V \sigma_{ij}\delta e_{ij} dV = \int_V f_i^B \delta u_i dV + \int_S f_i^s \delta u_i ds \tag{3.88}$$

where:

σ_{ij} = Cauchy stress component

$$e_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) = \text{deformation tensor (Bathe(2))}$$

u_i = displacement

x_i = current coordinate

f_i^B = component of body force

f_i^S = component of surface traction

V = volume of deformed body

S = surface of deformed body on which tractions are prescribed

The internal virtual work can be indicated by:

$$\delta W = \int_V \sigma_{ij}\delta e_{ij} dV \tag{3.89}$$

where:

W = internal virtual work

Element formulations are obtained by differentiating this virtual work expression (Bonet and Wood([236] (p. 934)) and Gadala and Wang([292] (p. 937))). In derivation, only linear differential terms are kept and all higher order terms are ignored so that finally a linear set of equations can be obtained.

In element formulation, material constitutive law is used to create the relation between stress increment and strain increment. The constitutive law only reflects the stress increment due to straining. However, the Cauchy stress is affected by the rigid body rotation and is not objective (not frame invariant). An objective stress is needed, therefore, to be able to be applied in constitutive law. One of these is Jaumann rate of Cauchy stress expressed by McMeeking and Rice([293] (p. 937))

$$\dot{\sigma}_{ij}^{J} = \dot{\sigma}_{ij} - \sigma_{ik}\dot{\omega}_{jk} - \sigma_{jk}\dot{\omega}_{ik} \tag{3.90}$$

where:

$\dot{\sigma}_{ij}^{J}$ = Jaumann rate of Cauchy stress

$\dot{\omega}_{ij} = \frac{1}{2}\left(\frac{\partial \upsilon_i}{\partial x_j} - \frac{\partial \upsilon_j}{\partial x_i}\right)$ = spin tensor

$\dot{\sigma}_{ij}$ = time rate of Cauchy stress

Therefore, the Cauchy stress rate is:

$$\dot{\sigma}_{ij} = \dot{\sigma}_{ij}^{J} + \sigma_{ik}\dot{\omega}_{jk} + \sigma_{jk}\dot{\omega}_{ik} \tag{3.91}$$

Using the constitutive law, the stress change due to straining can be expressed as:

$$\dot{\sigma}_{ij}^{J} = c_{ijkl}d_{kl} \tag{3.92}$$

where:

c_{ijkl} = material constitutive tensor

$d_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)$ = rate of deformation tensor

v_i = velocity

The Cauchy stress rate can be written as:

$$\dot{\sigma}_{ij} = c_{ijkl}d_{kl} + \sigma_{ik}\dot{\omega}_{jk} + \sigma_{jk}\dot{\omega}_{ik} \tag{3.93}$$

3.6.2. Classical Pure Displacement Formulation

Pure displacement formulation only takes displacements or velocities as primary unknown variables. All other quantities such as strains, stresses and state variables in history-dependent material models are derived from the displacements. It is the most widely used formulation and is able to handle most nonlinear deformation problems.

The differentiation of δW:

$$D\delta W = \int_V (D\sigma_{ij}\delta e_{ij}dV + \sigma_{ij}D\delta e_{ij}dV + \sigma_{ij}\delta e_{ij}D(dV)) \tag{3.94}$$

From Equation 3.93 (p. 52), the stress differentiation can be derived as:

$$D\sigma_{ij} = C_{ijkl}De_{kl} + \sigma_{ik}D\omega_{jk} + \sigma_{jk}D\omega_{ik} \tag{3.95}$$

where:

$$D\omega_{ij} = \frac{1}{2}D\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$$

The differentiation of dV is:

$$D(dV) = \frac{\partial Du_k}{\partial x_k}dV = De_v dV \tag{3.96}$$

where:

$$e_v = e_{ii}$$

Substitution of Equation 3.95 (p. 53) and Equation 3.96 (p. 53) into Equation 3.94 (p. 53) yields:

$$\begin{aligned} D\delta W = &\int_V \delta e_{ij}C_{ijkl}De_{kl}dV \\ &+\int_V \sigma_{ij}\left(\frac{\partial \delta u_k}{\partial x_i}\frac{\partial Du_k}{\partial x_j} - 2\delta e_{ik}De_{kj}\right)dV \\ &+\int_V \delta e_{ij}\sigma_{ij}\frac{\partial Du_k}{\partial x_k}dV \end{aligned} \tag{3.97}$$

The third term is unsymmetric and is usually insignificant in most of deformation cases. Hence, it is ignored. The final pure displacement formulation is:

$$\begin{aligned} D\delta W = &\int_V \delta e_{ij}C_{ijkl}De_{kl}dV \\ &+\int_V \sigma_{ij}\left(\frac{\partial \delta u_k}{\partial x_i}\frac{\partial Du_k}{\partial x_j} - \delta e_{ik}De_{kj}\right)dV \end{aligned} \tag{3.98}$$

The above equation is a set of linear equations in terms of Du_i or displacement change, which can be solved by standard linear solvers. This formulation is exactly the same as the one published by McMeeking and Rice([293] (p. 937)). The stiffness has two terms: the first one is material stiffness due to straining; the second one is stiffness due to geometric nonlinearity (stress stiffness).

Since no other assumption is made on the size or type of deformation, the formulation can be applied to any deformation problems (small deformation, finite deformation, small deformation with large rotation, stress stiffening, etc.) so it is called a general element formulation.

To achieve higher efficiency, the second term or stress stiffness is included only if requested for analyses with geometric nonlinearities (**NLGEOM**,ON, or **PSTRES**,ON) or buckling analysis (**ANTYPE**,BUCKLE).

3.6.3. Mixed u-P Formulations

The above pure displacement formulation is computationally efficient. However, the accuracy of any displacement formulation is dependent on Poisson's ratio or the bulk modulus. In such formulations, volumetric strain is determined from derivatives of displacements, which are not as accurately predicted as the displacements themselves. Under nearly incompressible conditions (Poisson's ratio is close to 0.5 or bulk modulus approaches infinity), any small error in the predicted volumetric strain will appear as a large error in the hydrostatic pressure and subsequently in the stresses. This error will, in turn, also affect the displacement prediction since external loads are balanced by the stresses, and may result in displacements very much smaller than they should be for a given mesh--this is called *locking*-- or, in some cases, in no convergence at all.

Another disadvantage of pure displacement formulation is that it is not to be able to handle fully incompressible deformation, such as fully incompressible hyperelastic materials.

To overcome these difficulties, mixed u-P formulations were developed. In these u-P formulations of the current-technology elements, the hydrostatic pressure $\bar{P}$ or volume change rate is interpolated on the element level and solved on the global level independently in the same way as displacements. The final stiffness matrix has the format of:

$$\begin{bmatrix} K_{uu} & K_{uP} \\ K_{Pu} & K_{PP} \end{bmatrix} \begin{Bmatrix} \Delta u \\ \Delta \bar{P} \end{Bmatrix} = \begin{Bmatrix} \Delta F \\ 0 \end{Bmatrix} \tag{3.99}$$

where:

Δu = displacement increment

$\Delta \bar{P}$ = hydrostatic pressure increment

Since hydrostatic pressure is obtained on a global level instead of being calculated from volumetric strain, the solution accuracy is independent of Poisson's ratio and bulk modulus. Hence, it is more robust for nearly incompressible material. For fully incompressible material, mixed u-P formulation has to be employed in order to get solutions.

The pressure DOFs are brought to global level by using internal or external nodes. The internal nodes are different from the regular (external) nodes in the following aspects:

- Each internal node is associated with only *one* element.
- The location of internal nodes is not important. They are used only to bring the pressure DOFs into the global equations.
- Internal nodes are created automatically and are not accessible by users.

The interpolation function of pressure is determined according to the order of elements. To remedy the locking problem, they are one order less than the interpolation function of strains or stresses. For most current-technology elements, the hydrostatic pressure degrees of freedom are introduced by the internal nodes. The number of pressure degrees of freedom, number of internal nodes, and interpolation

functions are shown in Table 3.1: Interpolation Functions of Hydrostatic Pressure of Current-Technology Elements (p. 55).

Table 3.1: Interpolation Functions of Hydrostatic Pressure of Current-Technology Elements

Element	KEY-OPT(6)	Internal nodes	$\bar{P}$	Functions
PLANE182 $\bar{B}$ selective reduced integration and uniform reduced integration	1	1	1	$\bar{P} = \bar{P}_1$
PLANE182 Enhanced strain formulation PLANE183	1	2	3	$\bar{P} = \bar{P}_1 + s\bar{P}_2 + t\bar{P}_3$
SOLID185 $\bar{B}$ selective reduced integration and uniform reduced integration	1	1	1	$\bar{P} = \bar{P}_1$
SOLID185 Enhanced strain formulation SOLID186 Uniform reduced integration and full integration	1	2	4	$\bar{P} = \bar{P}_1 + s\bar{P}_2 + t\bar{P}_3 + r\bar{P}_4$
SOLID187	1	1	1	$\bar{P} = \bar{P}_1$
SOLID187	2	2	4	$\bar{P} = \bar{P}_1 + s\bar{P}_2 + t\bar{P}_3 + r\bar{P}_4$
SOLID272	1	KEY-OPT(2) / 3	KEY-OPT(2)	$P = \bar{P}_1$ on r-z plane and Fourier interpolation in the circumferential (θ) direction
SOLID273	1	KEY-OPT(2)	KEY-OPT(2) x 3	$P = \bar{P}_1 + s\bar{P}_2 + t\bar{P}_3$ on r-z plane and Fourier interpolation in the circumferential (θ) direction

In Table 3.1: Interpolation Functions of Hydrostatic Pressure of Current-Technology Elements (p. 55), $\bar{P}$, $\bar{P}_1$, $\bar{P}_2$, $\bar{P}_3$, and $\bar{P}_4$ are the pressure degrees of freedom at internal node i. s, t, and r are the natural coordinates.

For SOLID285, the hydrostatic pressure degrees of freedom are introduced by extra degrees of freedom (HDSP) at each node. The total number of pressures and interpolation function of hydrostatic pressure are shown in Table 3.2: Interpolation Functions of Hydrostatic Pressure for SOLID285 (p. 56).

Table 3.2: Interpolation Functions of Hydrostatic Pressure for SOLID285

Element	$\bar{P}$	Functions
285	4	$\bar{P} = \bar{P}_1 + s\bar{P}_2 + t\bar{P}_3 + r\bar{P}_4$

$\bar{P}_1$, $\bar{P}_2$, $\bar{P}_3$, and $\bar{P}_4$ are the pressure degrees of freedom at each element node i. s, t, and r are the natural coordinates.

3.6.4. u-P Formulation I

This formulation is for nearly incompressible materials other than hyperelastic materials. For these materials, the volumetric constraint equations or volumetric compatibility can be defined as (see Bathe([2] (p. 921)) for details):

$$\frac{P - \bar{P}}{K} = 0 \tag{3.100}$$

where:

$P = -\sigma_m = -\frac{1}{3}\sigma_{ii}$ = hydrostatic pressure from material constitutive law

K = bulk modulus

P can also be defined as:

$$DP = -KDe_v \tag{3.101}$$

In mixed formulation, stress is updated and reported by:

$$\sigma_{ij} = \sigma'_{ij} - \delta_{ij}\bar{P} = \sigma_{ij} + \delta_{ij}P - \delta_{ij}\bar{P} \tag{3.102}$$

where:

δ_{ij} = Kronecker delta

σ_{ij} = Cauchy stress from constitutive law

so that the internal virtual work Equation 3.89 (p. 51) can be expressed as:

$$\delta W_a = \int_V \bar{\sigma}_{ij}\delta e_{ij} dV \tag{3.103}$$

Introduce the constraint Equation 3.100 (p. 56) by Lagrangian multiplier $\bar{P}$, the augmented internal virtual work is:

$$\delta W_a = \int_v \bar{\sigma}_{ij}\delta e_{ij} dV + \int_v \left(\frac{P-\bar{P}}{K}\right)\delta\bar{P} dV \tag{3.104}$$

Substitute Equation 3.102 (p. 56) into above; it is obtained:

$$\delta W_a = \int_v \sigma_{ij}\delta e_{ij} dV + \int_v (P-\bar{P})\delta e_v dV + \int_v \left(\frac{P-\bar{P}}{k}\right)\delta\bar{P} dV \tag{3.105}$$

where:

$e_v = \delta_{ij}\, e_{ij} = e_{ii}$

Take differentiation of Equation 3.104 (p. 57), ignore all higher terms of Du_i and $D\bar{P}$ than linear term, the final formulation can be expressed as:

$$\begin{aligned} D\delta W_a = & \int_v \delta e_{ij} C_{ijkl} De_{kl} dV - \int_v K De_v \delta e_v dV \\ & + \int_v \bar{\sigma}_{ij}\left(\frac{\partial \delta u_k}{\partial x_i}\frac{\partial Du_k}{\partial x_j} - 2\delta e_{ik} De_{kj}\right) dV \\ & - \int_v (D\bar{P}\delta e_v + De_v \delta\bar{P}) dV - \frac{1}{K}\int D\bar{P}\delta\bar{P} dV \end{aligned} \tag{3.106}$$

This is a linear set of equations of Du_i and $D\bar{P}$ (displacement and hydrostatic pressure changes). In the final mixed u-P formulation, the third term is the stress stiffness and is included only if requested (**NLGEOM**,ON or **PSTRES**,ON). The rest of the terms are based on the material stiffness. The first term is from material constitutive law directly or from straining; the second term is because of the stress modification (Equation 3.102 (p. 56)); the fourth and fifth terms are the extra rows and columns in stiffness matrix due to the introduction of the extra DOF: pressure, i.e., K_{uP}, K_{Pu} and K_{PP} as in Equation 3.99 (p. 54).

The stress stiffness in the above formulation is the same as the one in pure displacement formulation. All other terms exist even for small deformation and are the same as the one derived by Bathe([2] (p. 921)) for small deformation problems.

It is worthwhile to indicate that in the mixed formulation of the higher order elements (PLANE183 , SOLID186 and SOLID187 with KEYOPT(6) = 1), elastic strain only relates to the stress in the element on an averaged basis, rather than pointwise. The reason is that the stress is updated by Equation 3.102 (p. 56) and pressure $\bar{P}$ is interpolated independently in an element with a function which is one order lower than the function for volumetric strain. For lower order elements (PLANE182, SOLID185), this problem is eliminated since either B-bar technology or uniform reduced integration is used; volumetric strain is constant within an element, which is consistent with the constant pressure $\bar{P}$ interpolation functions (see Table 3.1: Interpolation Functions of Hydrostatic Pressure of Current-Technology Elements (p. 55)). In addition, this problem will not arise in element SOLID187 with linear interpolation function of $\bar{P}$ (KEYOPT(6) = 2). This is because the order of interpolation function of $\bar{P}$ is the same as the one for volumetric strain. In other words, the number of DOF $\bar{P}$ in one element is large enough to make $\bar{P}$ consistent with the volumetric strain at each integration point. Therefore, when mixed formulation of

element SOLID187 is used with nearly incompressible material, the linear interpolation function of $\bar{P}$ or KEYOPT(6) = 2 is recommended.

3.6.5. u-P Formulation II

A special formulation is necessary for fully incompressible hyperelastic material since the volume constraint equation is different and hydrostatic pressure can not be obtained from material constitutive law. Instead, it has to be calculated separately. For these kinds of materials, the stress has to be updated by:

$$\sigma_{ij} = \sigma'_{ij} - \delta_{ij}\bar{P} \tag{3.107}$$

where:

σ'_{ij} = deviatoric component of Cauchy stress tensor

The deviatoric component of deformation tensor defined by the e_{ij} term of Equation 3.88 (p. 51) can be expressed as:

$$e'_{ij} = e_{ij} - \frac{1}{3}\delta_{ij}e_v \tag{3.108}$$

The internal virtual work (Equation 3.89 (p. 51)) can be shown using σ'_{ij} and e'_{ij}:

$$\delta W = \int_v (\sigma'_{ij}\delta e'_{ij} - \bar{P}\delta e_v)dV \tag{3.109}$$

The volume constraint is the incompressible condition. For a fully incompressible hyperelastic material, it can be as defined by Sussman and Bathe([124] (p. 927)), Bonet and Wood([236] (p. 934)), Crisfield([294] (p. 937)):

$$1 - J = 0 \tag{3.110}$$

where:

$$J = |F_{ij}| = \left|\frac{\partial x_i}{\partial X_j}\right| = \frac{dV}{dV_o}$$

$|F_{ij}|$ = determinant of deformation gradient tensor
X_i = original coordinate
V_o = original volume

As in the mixed u-P formulation I (u-P Formulation I (p. 56)), the constraint Equation 3.110 (p. 58) was introduced to the internal virtual work by the Lagrangian multiplier $\bar{P}$. Then, differentiating the augmented internal virtual work, the final formulation is obtained.

This formulation is similar to the formulation for nearly incompressible materials, i.e. Equation 3.106 (p. 57). The only major difference is that $[K_{PP}] = [0]$ in this formulation. This is because material in this formulation is fully incompressible.

3.6.6. u-P Formulation III

When material behavior is almost incompressible, the pure displacement formulation may be applicable. The bulk modulus of material, however, is usually very large and thus often results in an ill-conditioned matrix. To avoid this problem, a special mixed u-P formulation is therefore introduced. The almost incompressible material usually has small volume changes at all material integration points. A new variable $\bar{J}$ is introduced to quantify this small volume change, and the constraint equation

$$J - \bar{J} = 0 \tag{3.111}$$

is enforced by introduction of the modified potential:

$$W + Q = W - \frac{\partial W}{\partial \bar{J}}(J - \bar{J}) \tag{3.112}$$

where:

W = hyperelastic strain energy potential
Q = energy augmentation due to volume constraint condition

3.6.7. Volumetric Constraint Equations in u-P Formulations

The final set of linear equations of mixed formulations (see Equation 3.99 (p. 54)) can be grouped into two:

$$[K_{uu}]\{\Delta u\} + [K_{uP}]\{\Delta \bar{P}\} = \{\Delta F\} \tag{3.113}$$

$$[K_{Pu}]\{\Delta u\} + [K_{PP}]\{\Delta \bar{P}\} = \{0\} \tag{3.114}$$

Equation 3.113 (p. 59) are the equilibrium equations and Equation 3.114 (p. 59) are the volumetric constraint equations. The total number of active equilibrium equations on a global level (indicated by N_d) is the total number of displacement DOFs without any prescribed displacement boundary condition. The total number of volumetric constraint equations (indicated by N_p) is the total number of pressure DOFs in all mixed u-P elements. The optimal ratio of N_d/N_p is 2 for 2-D elements and 3 for 3-D elements. When N_d/N_p is too small, the system may have too many constraint equations which may result in a severe locking problem. On the other hand, when N_d/N_p is too large, the system may have too few constraint equations which may result in too much deformation and loss of accuracy.

When $N_d/N_p < 1$, the system has more volumetric constraint equations than equilibrium equations, thus the system is over-constrained. In this case, if the u-P formulation I is used, the system equations will be very ill-conditioned so that it is hard to keep accuracy of solution and may cause divergence. If the u-P formulation II is used, the system equation will be singular because $[K_{PP}] = [0]$ in this formulation so that the system is not solvable. Therefore, over-constrained models should be avoided as described in the *Element Reference*.

Volumetric constraint is incorporated into the final equations as extra conditions. A check is made at the element level for elements with internal nodes for pressure degrees of freedom and at degrees of freedom (HDSP) at global level for SOLID285 to see if the constraint equations are satisfied. The number of elements in which constraint equations have not been satisfied is reported for current-technology elements if the check is done at element level.

For u-P formulation I, the volumetric constraint is met if:

$$\left| \frac{\int_V \frac{P - \bar{P}}{K} dV}{V} \right| \leq tol_V \tag{3.115}$$

and for u-P formulation II, the volumetric constraint is met if:

$$\left| \frac{\int_V \frac{J - 1}{J} dV}{V} \right| \leq tol_V \tag{3.116}$$

and for u-P formulation III, the volumetric constraint is met if:

$$\left| \frac{\int_V \frac{J - \bar{J}}{J} dV}{V} \right| \leq tol_V \tag{3.117}$$

where:

tol_V = tolerance for volumetric compatibility (input as Vtol on **SOLCONTROL** command)

3.7. Constraints and Lagrange Multiplier Method

Constraints are generally implemented using the Lagrange Multiplier Method (See Belytschko([348] (p. 940))). This formulation has been implemented in MPC184 as described in the *Element Reference*. In this method, the internal energy term given by Equation 3.89 (p. 51) is augmented by a set of constraints, imposed by the use of Lagrange multipliers and integrated over the volume leading to an augmented form of the virtual work equation:

$$\delta W' = \delta W + \int \delta \bar{\lambda}^T \bar{\Phi}(\bar{u}) dv + \int \bar{\lambda}^T \delta \bar{\Phi}(\bar{u}) dv \tag{3.118}$$

where:

W' = augmented potential

and

$$\bar{\Phi}(\bar{u}) = \bar{0} \tag{3.119}$$

is the set of constraints to be imposed.

The variation of the augmented potential is zero provided $\bar{\Phi}(\bar{u}) = \bar{0}$ (and, hence $\delta\bar{\Phi} = \bar{0}$) and, simultaneously:

$$\delta W = 0 \tag{3.120}$$

The equation for augmented potential (Equation 3.118 (p. 60)) is a system of n_{tot} equations, where:

$$n_{tot} = n_{dof} + n_c \tag{3.121}$$

where:

n_{dof} = number of degrees of freedom in the model
n_c = number of Lagrange multipliers

The solution vector consists of the displacement degrees of freedom $\bar{u}$ and the Lagrange multipliers.

The stiffness matrix is of the form:

$$\begin{bmatrix} \bar{K} + \bar{\lambda}\bar{H} & \bar{B}^T \\ \bar{B} & \bar{0} \end{bmatrix} \begin{Bmatrix} \Delta\bar{u} \\ \Delta\bar{\lambda} \end{Bmatrix} = \begin{Bmatrix} -\bar{r} - \bar{\lambda}^T\bar{B} \\ \bar{\Phi}(\bar{u}) \end{Bmatrix} \tag{3.122}$$

where:

$$\bar{r} = \bar{f}_{int} - \bar{f}_{ext}$$
$$= \int \sigma_{ij}\delta e_{ij} - \int_v f_i^B \delta u_i dv - \int_s f_i^s \delta u_i ds$$

$$\bar{K} = \delta\bar{r}$$

$$\bar{B} = \frac{\partial\bar{\Phi}(\bar{u})}{\partial\bar{u}}$$

$$\bar{H} = \frac{\partial\bar{B}}{\partial\bar{u}}$$

$\Delta\bar{u}, \Delta\bar{\lambda}$ = increments in displacements and Lagrange multiplier, respectively.

Chapter 4: Structures with Material Nonlinearities

This chapter discusses the structural material nonlinearities of plasticity, creep, nonlinear elasticity, hyperelasticity, viscoelasticity, concrete and swelling. Not included in this section are the slider, frictional, or other nonlinear elements (discussed in Element Library (p. 411)) that can represent other nonlinear material behavior.

The following topics are available:

4.1. Understanding Material Nonlinearities

Material nonlinearities occur because of the nonlinear relationship between stress and strain; that is, the stress is a nonlinear function of the strain. The relationship is also path-dependent (except for the case of nonlinear elasticity and hyperelasticity), so that the stress depends on the strain history as well as the strain itself.

The program can account for many material nonlinearities, as follows:

1. Rate-independent plasticity is characterized by the irreversible instantaneous straining that occurs in a material.

2. Rate-dependent plasticity allows the plastic-strains to develop over a time interval. This is also termed viscoplasticity.

3. Creep is also an irreversible straining that occurs in a material and is rate-dependent so that the strains develop over time. The time frame for creep is usually much larger than that for rate-dependent plasticity.

4. Gasket material may be modelled using special relationships.

5. Nonlinear elasticity allows a nonlinear stress-strain relationship to be specified. All straining is reversible.

6. Hyperelasticity is defined by a strain-energy density potential that characterizes elastomeric and foam-type materials. All straining is reversible.

7. Viscoelasticity is a rate-dependent material characterization that includes a viscous contribution to the elastic straining.

8. Concrete materials include cracking and crushing capability.

9. Swelling allows materials to enlarge in the presence of neutron flux.

Llisted in this table are the number of stress and strain components involved. One component uses X (e.g., SX, EPELX, etc.), four components use X, Y, Z, XY, and six components use X, Y, Z, XY, YZ, XZ.

Strain Definitions

For the case of nonlinear materials, the definition of elastic strain given with Equation 2.1 (p. 5) has the form of:

$$\{\varepsilon^{el}\} = \{\varepsilon\} - \{\varepsilon^{th}\} - \{\varepsilon^{pl}\} - \{\varepsilon^{cr}\} - \{\varepsilon^{sw}\} \tag{4.1}$$

where:

ε^{el} = elastic strain vector (output as EPEL)
ε = total strain vector
ε^{th} = thermal strain vector (output as EPTH)
ε^{pl} = plastic strain vector (output as EPPL)
ε^{cr} = creep strain vector (output as EPCR)
ε^{sw} = swelling strain vector (output as EPSW)

In this case, $\{\varepsilon\}$ is the strain measured by a strain gauge. Equation 4.1 (p. 64) is only intended to show the relationships between the terms. See subsequent sections for more detail).

In POST1, total strain is reported as:

$$\{\varepsilon^{tot}\} = \{\varepsilon^{el}\} + \{\varepsilon^{pl}\} + \{\varepsilon^{cr}\} \tag{4.2}$$

where:

ε^{tot} = component total strain (output as EPTO)

Comparing the last two equations,

$$\{\varepsilon^{tot}\} = \{\varepsilon\} - \{\varepsilon^{th}\} - \{\varepsilon^{sw}\} \tag{4.3}$$

The difference between these two "total" strains stems from the different usages: $\{\varepsilon\}$ can be used to compare strain gauge results and ε^{tot} can be used to plot nonlinear stress-strain curves.

4.2. Rate-Independent Plasticity

Rate-independent plasticity is characterized by the irreversible straining that occurs in a material once a certain level of stress is reached. The plastic strains are assumed to develop instantaneously, that is, independent of time. Several options are available to characterize different types of material behaviors:

- Material Behavior Option
- Bilinear Isotropic Hardening

- Multilinear Isotropic Hardening
- Nonlinear Isotropic Hardening
- Classical Bilinear Kinematic Hardening
- Multilinear Kinematic Hardening
- Nonlinear Kinematic Hardening
- Anisotropic
- Drucker-Prager
- Cast Iron
- User-specified behavior (as described in User Routines and Non-Standard Uses in the *Advanced Analysis Guide* and in the *Guide to ANSYS User Programmable Features*)

Except for user-specified behavior (**TB**,USER), each material-behavior option is explained in greater detail later in this chapter. Figure 4.1: Stress-Strain Behavior of Each of the Plasticity Options (p. 67) represents the stress-strain behavior of each of the options.

4.2.1. Theory

Plasticity theory provides a mathematical relationship that characterizes the elastoplastic response of materials. There are three ingredients in the rate-independent plasticity theory: the yield criterion, flow rule and the hardening rule. These will be discussed in detail subsequently. Table 4.1: Notation (p. 65) summarizes the notation used in the remainder of this chapter.

4.2.2. Yield Criterion

The yield criterion determines the stress level at which yielding is initiated. For multi-component stresses, this is represented as a function of the individual components, $f(\{\sigma\})$, which can be interpreted as an equivalent stress σ_e:

$$\sigma_e = f(\{\sigma\}) \tag{4.4}$$

where:

$\{\sigma\}$ = stress vector

Table 4.1: Notation

Variable	Definition	ANSYS Output Label
$\{\varepsilon^{el}\}$	elastic strains	EPEL
$\{\varepsilon^{pl}\}$	plastic strains	EPPL
$\{\varepsilon^{tr}\}$	trial strain	
$\hat{\varepsilon}^{pl}$	equivalent plastic strain	EPEQ
$\{\sigma\}$	stresses	S

Variable	Definition	ANSYS Output Label
σ_e	equivalent stress	
σ_y	material yield parameter	
σ_m	mean or hydrostatic stress	HPRES
$\hat{\sigma}_e^{pl}$	equivalent stress parameter	SEPL
λ	plastic multiplier	
$\{\alpha\}$	yield surface translation	
κ	plastic work	
C	translation multiplier	
[D]	stress-strain matrix	
E_T	tangent modulus	
F	yield criterion	
N	stress ratio	SRAT
Q	plastic potential	
{S}	deviatoric stress	

When the equivalent stress is equal to a material yield parameter σ_y,

$$f(\{\sigma\}) = \sigma_y \tag{4.5}$$

the material will develop plastic strains. If σ_e is less than σ_y, the material is elastic and the stresses will develop according to the elastic stress-strain relations. Note that the equivalent stress can never exceed the material yield since in this case plastic strains would develop instantaneously, thereby reducing the stress to the material yield. Equation 4.5 (p. 66) can be plotted in stress space as shown in Figure 4.2: Various Yield Surfaces (p. 68) for some of the plasticity options. The surfaces in Figure 4.2: Various Yield Surfaces (p. 68) are known as the yield surfaces and any stress state inside the surface is elastic, that is, they do not cause plastic strains.

Figure 4.1: Stress-Strain Behavior of Each of the Plasticity Options

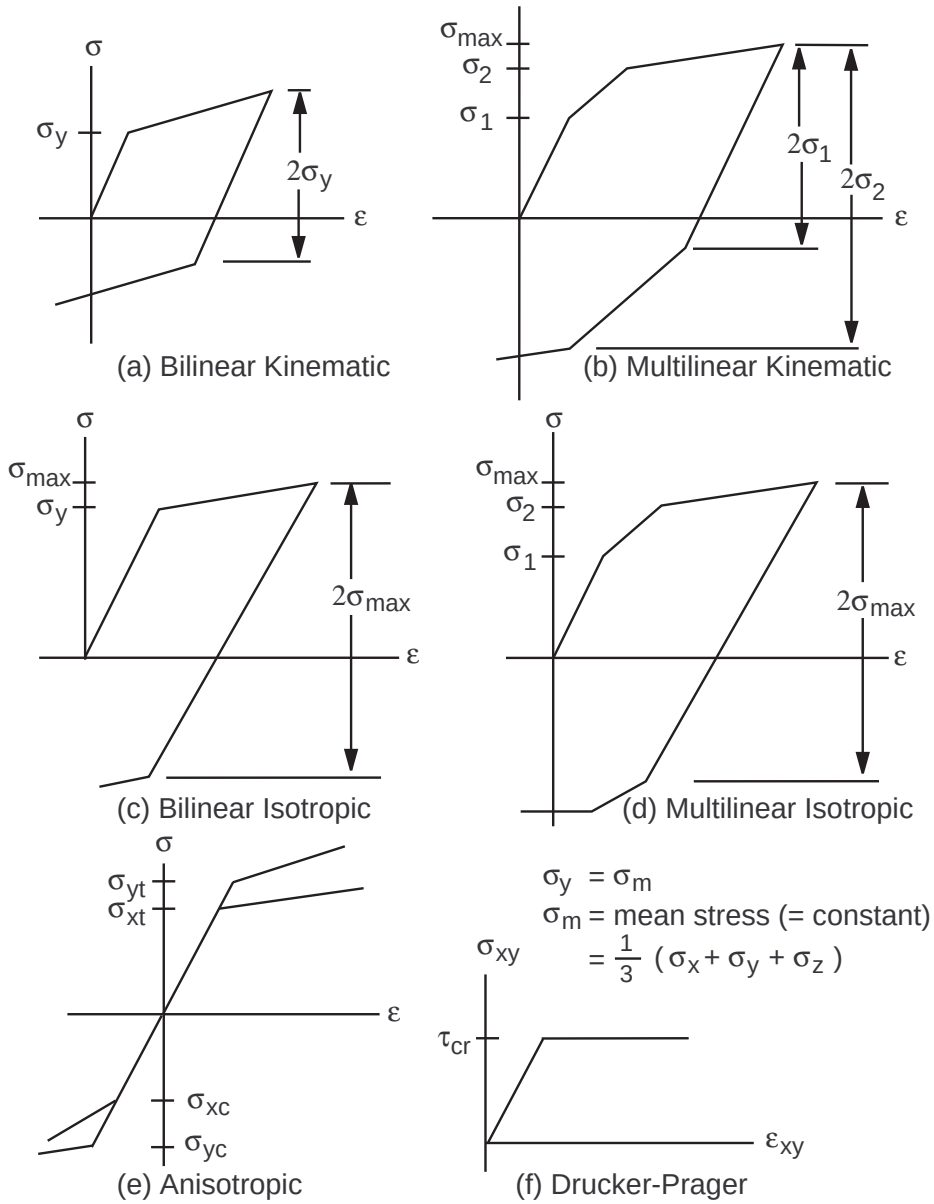
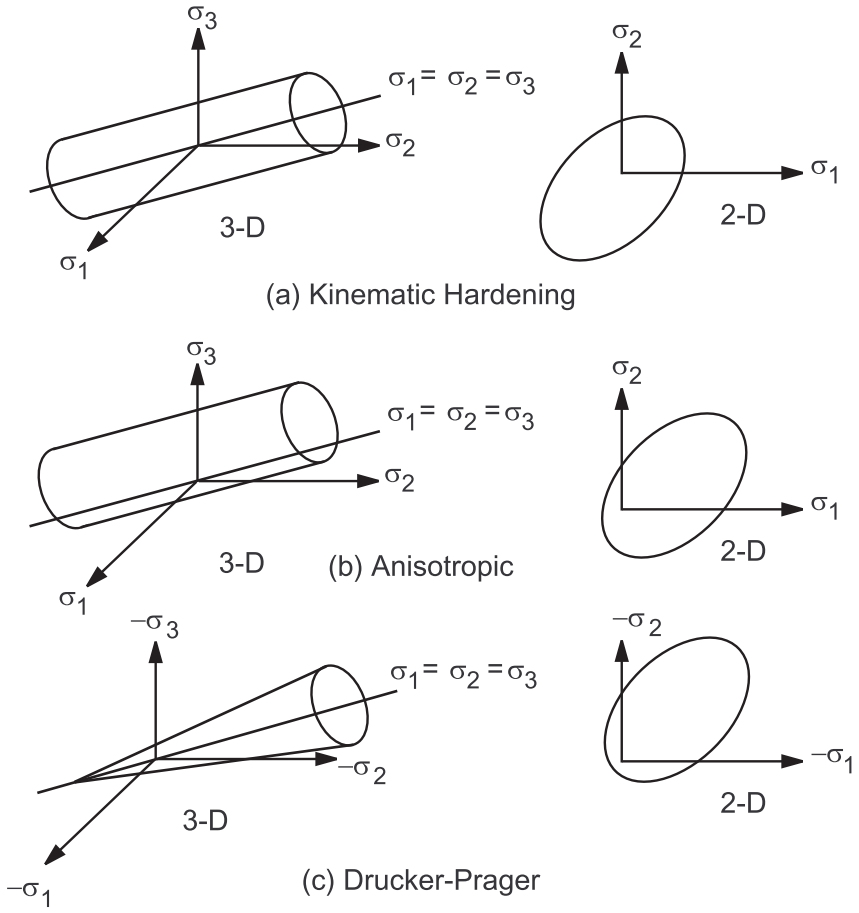


Figure 4.2: Various Yield Surfaces



4.2.3. Flow Rule

The flow rule determines the direction of plastic straining and is given as:

$$\{d\varepsilon^{pl}\} = \lambda \left\{ \frac{\partial Q}{\partial \sigma} \right\} \tag{4.6}$$

where:

λ = plastic multiplier (which determines the amount of plastic straining)
Q = function of stress termed the plastic potential (which determines the direction of plastic straining)

If Q is the yield function (as is normally assumed), the flow rule is termed associative and the plastic strains occur in a direction normal to the yield surface.

4.2.4. Hardening Rule

The hardening rule describes the changing of the yield surface with progressive yielding, so that the conditions (i.e. stress states) for subsequent yielding can be established. Two hardening rules are available: work (or isotropic) hardening and kinematic hardening. In work hardening, the yield surface remains centered about its initial centerline and expands in size as the plastic strains develop. For materials with isotropic plastic behavior this is termed isotropic hardening and is shown in Figure 4.3: Types

of Hardening Rules (p. 69) (a). Kinematic hardening assumes that the yield surface remains constant in size and the surface translates in stress space with progressive yielding, as shown in Figure 4.3: Types of Hardening Rules (p. 69) (b).

The yield criterion, flow rule and hardening rule for each option are summarized in Table 4.2: Summary of Plasticity Options (p. 69) and are discussed in detail later in this chapter.

Figure 4.3: Types of Hardening Rules

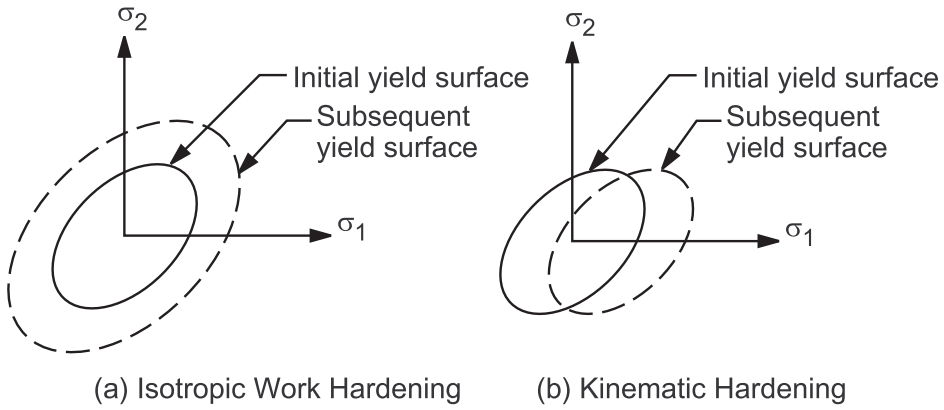


Table 4.2: Summary of Plasticity Options

Name	TB Lab	Yield Criterion	Flow Rule	Hardening Rule	Material Response
Bilinear Isotropic Hardening	BISO	von Mises/Hill	associative	work hardening	bilinear
Multilinear Isotropic Hardening	MISO	von Mises/Hill	associative	work hardening	multilinear
Nonlinear Isotropic Hardening	NLISO	von Mises/Hill	associative	work hardening	nonlinear
Classical Bilinear Kinematic Hardening	BKIN	von Mises/Hill	associative (Prandtl- Reuss equations)	kinematic hardening	bilinear
Multilinear Kinematic Hardening	MKIN/KINH	von Mises/Hill	associative	kinematic hardening	multilinear
Nonlinear Kinematic Hardening	CHAB	von Mises/Hill	associative	kinematic hardening	nonlinear
Anisotropic	ANISO	modified von Mises	associative	work hardening	bilinear, each direction and tension and compression different

Name	TB Lab	Yield Criterion	Flow Rule	Hardening Rule	Material Response
Drucker- Prager	DP	von Mises with dependence on hydrostatic stress	associative or non- associative	none	elastic- perfectly plastic
Extended Drucker-Prager	EDP	von MIses with dependence on hydrostatic stress	associative or non- associative	work hardening	multilinear
Cast Iron	CAST	von Mises with dependence on hydrostatic stress	non- associative	work hardening	multilinear
Gurson	GURS	von Mises with dependence pressure and porosity	associative	work hardening	multilinear

4.2.5. Plastic Strain Increment

If the equivalent stress computed using elastic properties exceeds the material yield, then plastic straining must occur. Plastic strains reduce the stress state so that it satisfies the yield criterion, Equation 4.5 (p. 66). Based on the theory presented in the previous section, the plastic strain increment is readily calculated.

The hardening rule states that the yield criterion changes with work hardening and/or with kinematic hardening. Incorporating these dependencies into Equation 4.5 (p. 66), and recasting it into the following form:

$$F(\{\sigma\}, \kappa, \{\alpha\}) = 0 \tag{4.7}$$

where:

κ = plastic work
$\{\alpha\}$ = translation of yield surface

κ and $\{\alpha\}$ are termed internal or state variables. Specifically, the plastic work is the sum of the plastic work done over the history of loading:

$$\kappa = \int \{\sigma\}^T [M] \{d\varepsilon^{pl}\} \tag{4.8}$$

where:

$$[M] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

and translation (or shift) of the yield surface is also history dependent and is given as:

$$\{\alpha\} = \int C\{d\varepsilon^{pl}\} \tag{4.9}$$

where:

C = material parameter
$\{\alpha\}$ = back stress (location of the center of the yield surface)

Equation 4.7 (p. 70) can be differentiated so that the consistency condition is:

$$dF = \left\{\frac{\partial F}{\partial \sigma}\right\}^T [M]\{d\sigma\} + \frac{\partial F}{\partial \kappa} d\kappa + \left\{\frac{\partial F}{\partial \alpha}\right\}^T [M]\{d\alpha\} = 0 \tag{4.10}$$

Noting from Equation 4.8 (p. 70) that

$$d\kappa = \{\sigma\}^T [M]\{d\varepsilon^{pl}\} \tag{4.11}$$

and from Equation 4.9 (p. 71) that

$$\{d\alpha\} = C\{d\varepsilon^{pl}\} \tag{4.12}$$

Equation 4.10 (p. 71) becomes

$$\left\{\frac{\partial F}{\partial \sigma}\right\}^T [M]\{d\sigma\} + \frac{\partial F}{\partial \kappa}\{\sigma\}^T [M]\{d\varepsilon^{pl}\} + C\left\{\frac{\partial F}{\partial \alpha}\right\}^T [M]\{d\varepsilon^{pl}\} = 0 \tag{4.13}$$

The stress increment can be computed via the elastic stress-strain relations

$$\{d\sigma\} = [D]\{d\varepsilon^{el}\} \tag{4.14}$$

where:

[D] = stress-strain matrix

with

$$\{d\varepsilon^{el}\} = \{d\varepsilon\} - \{d\varepsilon^{pl}\} \tag{4.15}$$

since the total strain increment can be divided into an elastic and plastic part. Substituting Equation 4.6 (p. 68) into Equation 4.13 (p. 71) and Equation 4.15 (p. 71) and combining Equation 4.13 (p. 71), Equation 4.14 (p. 71), and Equation 4.15 (p. 71) yields

$$\lambda = \frac{\left\{\frac{\partial F}{\partial \sigma}\right\}^T [M][D]\{d\varepsilon\}}{-\left\{\frac{\partial F}{\partial \kappa}\right\}\{\sigma\}^T [M]\left\{\frac{\partial Q}{\partial \sigma}\right\} - C\left\{\frac{\partial F}{\partial \alpha}\right\}^T [M]\left\{\frac{\partial Q}{\partial \sigma}\right\} + \left\{\frac{\partial F}{\partial \sigma}\right\}^T [M][D]\left\{\frac{\partial Q}{\partial \sigma}\right\}} \tag{4.16}$$

The size of the plastic strain increment is therefore related to the total increment in strain, the current stress state, and the specific forms of the yield and potential surfaces. The plastic strain increment is then computed using Equation 4.6 (p. 68):

$$\{d\varepsilon^{pl}\} = \lambda\left\{\frac{\partial Q}{\partial \sigma}\right\} \tag{4.17}$$

4.2.6. Implementation

An Euler backward scheme is used to enforce the consistency condition Equation 4.10 (p. 71). This ensures that the updated stress, strains and internal variables are on the yield surface. The algorithm proceeds as follows:

1. The material parameter σ_y Equation 4.5 (p. 66) is determined for this time step (e.g., the yield stress at the current temperature).

2. The stresses are computed based on the trial strain $\{\varepsilon^{tr}\}$, which is the total strain minus the plastic strain from the previous time point (thermal and other effects are ignored):

   $$\{\varepsilon_n^{tr}\} = \{\varepsilon_n\} - \{\varepsilon_{n-1}^{pl}\} \tag{4.18}$$

   where the superscripts are described with Understanding Theory Reference Notation (p. 2) and subscripts refer to the time point. Where all terms refer to the current time point, the subscript is dropped. The trial stress is then

   $$\{\sigma^{tr}\} - [D]\{\varepsilon^{tr}\} \tag{4.19}$$

3. The equivalent stress σ_e is evaluated at this stress level by Equation 4.4 (p. 65). If σ_e is less than σ_y the material is elastic and no plastic strain increment is computed.

4. If the stress exceeds the material yield, the plastic multiplier λ is determined by a local Newton-Raphson iteration procedure (Simo and Taylor([155] (p. 929))).

5. $\{\Delta\varepsilon^{pl}\}$ is computed via Equation 4.17 (p. 72).

6. The current plastic strain is updated

   $$\{\varepsilon_n^{pl}\} = \{\varepsilon_{n-1}^{pl}\} + \{\Delta\varepsilon^{pl}\} \tag{4.20}$$

   where:

   $\{\varepsilon_n^{pl}\}$ = current plastic strains (output as EPPL)

   and the elastic strain computed

   $$\{\varepsilon^{el}\} = \{\varepsilon^{tr}\} - \{\Delta\varepsilon^{pl}\} \tag{4.21}$$

where:

ε^{el} = elastic strains (output as EPEL)

The stress vector is:

$$\{\sigma\} = [D]\{\varepsilon^{el}\} \tag{4.22}$$

where:

$\{\sigma\}$ = stresses (output as S)

7. The increments in the plastic work $\Delta\kappa$ and the center of the yield surface $\{\Delta\alpha\}$ are computed via Equation 4.11 (p. 71) and Equation 4.12 (p. 71) and the current values updated

$$\kappa_n = \kappa_{n-1} + \Delta\kappa \tag{4.23}$$

and

$$\{\alpha_n\} = \{\alpha_{n-1}\} + \{\Delta\alpha\} \tag{4.24}$$

where the subscript n-1 refers to the values at the previous time point.

8. For output purposes, an equivalent plastic strain $\hat{\varepsilon}^{pl}$ (output as EPEQ), equivalent plastic strain increment $\Delta\hat{\varepsilon}^{pl}$ (output with the label "MAX PLASTIC STRAIN STEP"), equivalent stress parameter $\hat{\sigma}_e^{pl}$ (output as SEPL) and stress ratio N (output as SRAT) are computed. The stress ratio is given as

$$N = \frac{\sigma_e}{\sigma_y} \tag{4.25}$$

where σ_e is evaluated using the trial stress . N is therefore greater than or equal to one when yielding is occurring and less than one when the stress state is elastic. The equivalent plastic strain increment is given as:

$$\Delta\hat{\varepsilon}^{pl} = \left(\frac{2}{3}\{\Delta\varepsilon^{pl}\}^T[M]\{\Delta\varepsilon^{pl}\}\right)^{\frac{1}{2}} \tag{4.26}$$

The equivalent plastic strain and equivalent stress parameters are developed for each option in the next sections.

Note that the Euler backward integration scheme in step 4 is the radial return algorithm (Krieg([46] (p. 923))) for the von Mises yield criterion.

4.2.7. Elastoplastic Stress-Strain Matrix

The tangent or elastoplastic stress-strain matrix is derived from the local Newton-Raphson iteration scheme used in step 4 above (Simo and Taylor([155] (p. 929))). It is therefore the consistent (or algorithmic) tangent. If the flow rule is nonassociative ($F \neq Q$), then the tangent is unsymmetric. To preserve the symmetry of the matrix, for analyses with a nonassociative flow rule (Drucker-Prager only), the matrix is evaluated using F only and again with Q only and the two matrices averaged.

4.2.8. Specialization for Hardening

Multilinear Isotropic Hardening and Bilinear Isotropic Hardening

These options use the von Mises yield criterion with the associated flow rule and isotropic (work) hardening (accessed with **TB**,MISO and **TB**,BISO).

The equivalent stress Equation 4.4 (p. 65) is:

$$\sigma_e = \left[\frac{3}{2}\{s\}^T[M]\{s\}\right]^{\frac{1}{2}} \tag{4.27}$$

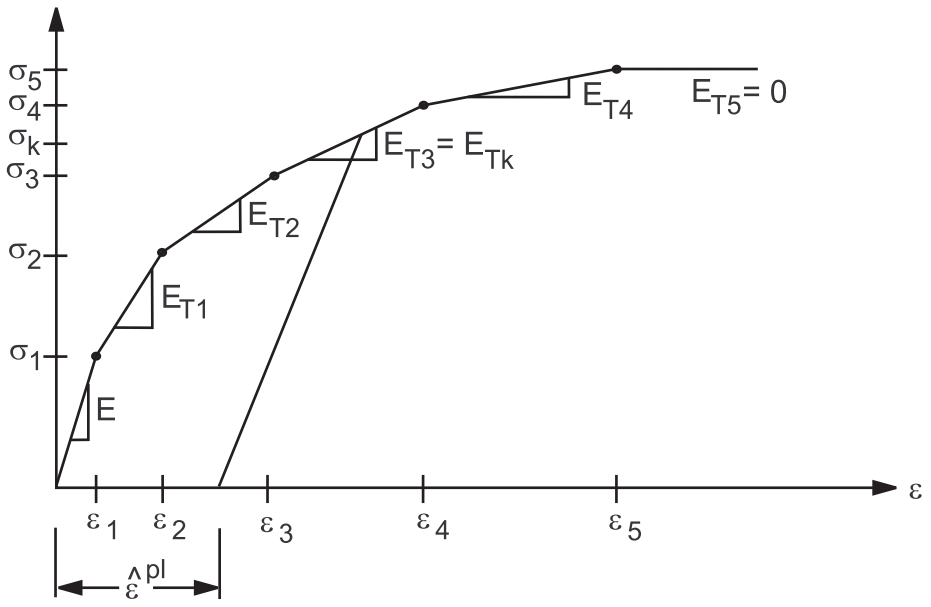
where {s} is the deviatoric stress Equation 4.37 (p. 76). When σ_e is equal to the current yield stress σ_k the material is assumed to yield. The yield criterion is:

$$F = \left[\frac{3}{2}\{s\}^T[M]\{s\}\right]^{\frac{1}{2}} - \sigma_k = 0 \tag{4.28}$$

For work hardening, σ_k is a function of the amount of plastic work done. For the case of isotropic

plasticity assumed here, σ_k can be determined directly from the equivalent plastic strain $\hat{\varepsilon}^{pl}$ of Equation 4.42 (p. 77) (output as EPEQ) and the uniaxial stress-strain curve as depicted in Figure 4.4: Uniaxial Behavior (p. 74). σ_k is output as the equivalent stress parameter (output as SEPL). For temperature-dependent curves with the MISO option, σ_k is determined by temperature interpolation of the input curves after they have been converted to stress-plastic strain format.

Figure 4.4: Uniaxial Behavior



For Multilinear Isotropic Hardening and σ_k Determination

4.2.9. Specification for Nonlinear Isotropic Hardening

Both the Voce([253] (p. 935)) hardening law, and the nonlinear power hardening law can be used to model nonlinear isotropic hardening. The Voce hardening law for nonlinear isotropic hardening behavior (accessed with **TB**,NLISO,,,,VOCE) is specified by the following equation:

$$R = k + R_o \hat{\varepsilon}^{pl} + R_\infty (1 - e^{-b \hat{\varepsilon}^{pl}}) \tag{4.29}$$

where:

k = elastic limit

R_o, R_∞, b = material parameters characterizing the isotropic hardening behavior of materials

$\hat{\varepsilon}^{pl}$ = equivalent plastic strain

The constitutive equations are based on linear isotropic elasticity, the von Mises yield function and the associated flow rule. The yield function is:

$$F = \left[\frac{3}{2}\{s\}^T [M]\{s\}\right]^{\frac{1}{2}} - R = 0 \tag{4.30}$$

The plastic strain increment is:

$$\{\Delta\varepsilon^{pl}\} = \lambda \left\{\frac{\partial Q}{\partial \sigma}\right\} = \lambda \left\{\frac{\partial F}{\partial \sigma}\right\} = \frac{3}{2}\lambda \frac{\{s\}}{\sigma_e} \tag{4.31}$$

where:

λ = plastic multiplier

The equivalent plastic strain increment is then:

$$\Delta\hat{\varepsilon}^{pl} = \sqrt{\frac{2}{3}\{\Delta\varepsilon^{pl}\}^T [M]\{\Delta\varepsilon^{pl}\}} = \lambda \tag{4.32}$$

The accumulated equivalent plastic strain is:

$$\varepsilon^{pl} = \sum \Delta\hat{\varepsilon}^{pl} \tag{4.33}$$

The power hardening law for nonlinear isotropic hardening behavior (accessed with **TB**,NLISO,,,,POWER) which is used primarily for ductile plasticity and damage is developed in the Gurson's Model (p. 96):

$$\frac{\sigma_Y}{\sigma_0} = \left(\frac{\sigma_Y}{\sigma_0} + \frac{3G}{\sigma_0}\bar{\varepsilon}^p\right)^N \tag{4.34}$$

where:

σ_Y = current yield strength

σ_0 = initial yield strength

G = shear modulus

$\bar{\varepsilon}^p$ is the microscopic equivalent plastic strain and is defined by:

$$\dot{\bar{\varepsilon}}^p = \frac{\sigma : \dot{\varepsilon}^p}{(1-f)\sigma_Y} \tag{4.35}$$

where:

ε^p = macroscopic plastic strain tensor
$\dot{}$ = rate change of variables
σ = Cauchy stress tensor
: = inner product operator of two second order tensors
f = porosity

4.2.10. Specialization for Bilinear Kinematic Hardening

This option uses the von Mises yield criterion with the associated flow rule and kinematic hardening (accessed with **TB**,BKIN).

The equivalent stress Equation 4.4 (p. 65) is therefore

$$\sigma_e = \left[\frac{3}{2}(\{s\}-\{\alpha\})^T [M](\{s\}-\{\alpha\})\right]^{\frac{1}{2}} \tag{4.36}$$

where: {s} = deviatoric stress vector

$$\{s\} = \{\sigma\} - \sigma_m \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{4.37}$$

where:

σ_m = mean or hydrostatic stress $= \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z)$

{α} = yield surface translation vector Equation 4.9 (p. 71)

Note that since Equation 4.36 (p. 76) is dependent on the deviatoric stress, yielding is independent of the hydrostatic stress state. When σ_e is equal to the uniaxial yield stress, σ_y, the material is assumed to yield. The yield criterion Equation 4.7 (p. 70) is therefore,

$$F = \left[\frac{3}{2}(\{s\}-\{\alpha\})^T [M](\{s\}-\{\alpha\})\right]^{\frac{1}{2}} - \sigma_y = 0 \tag{4.38}$$

The associated flow rule yields

$$\left\{\frac{\partial Q}{\partial \sigma}\right\} = \left\{\frac{\partial F}{\partial \sigma}\right\} = \frac{3}{2\sigma_e}(\{s\}-\{a\}) \tag{4.39}$$

so that the increment in plastic strain is normal to the yield surface. The associated flow rule with the von Mises yield criterion is known as the Prandtl-Reuss flow equation.

The yield surface translation is defined as:

$$\{\alpha\} = 2G\{\varepsilon^{sh}\} \tag{4.40}$$

where:

G = shear modulus = E/(2 (1+ν))
E = Young's modulus (input as EX on **MP** command)
ν = Poisson's ratio (input as PRXY or NUXY on **MP** command)

The shift strain is computed analogously to Equation 4.24 (p. 73):

$$\{\varepsilon_n^{sh}\} = \{\varepsilon_{n-1}^{sh}\} + \{\Delta\varepsilon^{sh}\} \tag{4.41}$$

where:

$$\{\Delta\varepsilon^{sh}\} = \frac{C}{2G}\{\Delta\varepsilon^{pl}\}$$

$$C = \frac{2}{3}\frac{EE_T}{E - E_T} \tag{4.42}$$

where:

E = Young's modulus (input as EX on **MP** command)
E_T = tangent modulus from the bilinear uniaxial stress-strain curve

The yield surface translation $\{\varepsilon^{sh}\}$ is initially zero and changes with subsequent plastic straining.

The equivalent plastic strain is dependent on the loading history and is defined to be:

$$\hat{\varepsilon}_n^{pl} = \hat{\varepsilon}_{n-1}^{pl} + \Delta\hat{\varepsilon}^{pl} \tag{4.43}$$

where:

$\hat{\varepsilon}_n^{pl}$ = equivalent plastic strain for this time point (output as EPEQ)

$\hat{\varepsilon}_{n-1}^{pl}$ = equivalent plastic strain from the previous time point

The equivalent stress parameter is defined to be:

$$\hat{\sigma}_e^{pl} = \sigma_y + \frac{EE_T}{E - E_T}\hat{\varepsilon}_n^{pl} \tag{4.44}$$

where:

$\hat{\sigma}_e^{pl}$ = equivalent stress parameter (output as SEPL)

Note that if there is no plastic straining ($\hat{\varepsilon}^{pl} = 0$), then $\hat{\sigma}_e^{pl}$ is equal to the yield stress. $\hat{\sigma}_e^{pl}$ only has meaning during the initial, monotonically increasing portion of the load history. If the load were to be reversed after plastic loading, the stresses and therefore σ_e would fall below yield but $\hat{\sigma}_e^{pl}$ would register above yield (since $\hat{\varepsilon}^{pl}$ is nonzero).

4.2.11. Specialization for Multilinear Kinematic Hardening

This option (accessed with **TB**,MKIN and **TB**,KINH) uses the Besseling([53] (p. 923)) model also called the sublayer or overlay model (Zienkiewicz([54] (p. 923))) to characterize the material behavior. The material behavior is assumed to be composed of various portions (or subvolumes), all subjected to the same total strain, but each subvolume having a different yield strength. (For a plane stress analysis, the material can be thought to be made up of a number of different layers, each with a different thickness and yield stress.) Each subvolume has a simple stress-strain response but when combined the model can represent complex behavior. This allows a multilinear stress-strain curve that exhibits the Bauschinger (kinematic hardening) effect (Figure 4.1: Stress-Strain Behavior of Each of the Plasticity Options (p. 67) (b)).

The following steps are performed in the plasticity calculations:

1. The portion of total volume for each subvolume and its corresponding yield strength are determined.
2. The increment in plastic strain is determined for each subvolume assuming each subvolume is subjected to the same total strain.
3. The individual increments in plastic strain are summed using the weighting factors determined in step 1 to compute the overall or apparent increment in plastic strain.
4. The plastic strain is updated and the elastic strain is computed.

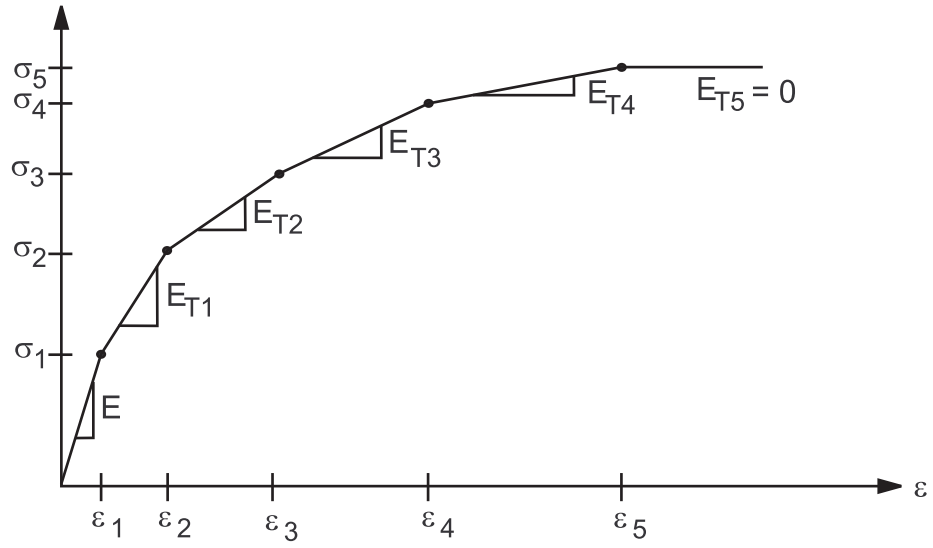
The portion of total volume (the weighting factor) and yield stress for each subvolume is determined by matching the material response to the uniaxial stress-strain curve. A perfectly plastic von Mises material is assumed and this yields for the weighting factor for subvolume k

$$w_k = \frac{E - E_{Tk}}{E - \frac{1-2\nu}{3}E_{Tk}} - \sum_{i=1}^{k-1} w_i \tag{4.45}$$

where:

w_k = the weighting factor (portion of total volume) for subvolume k and is evaluated sequentially from 1 to the number of subvolumes
E_{Tk} = the slope of the kth segment of the stress-strain curve (see Figure 4.5: Uniaxial Behavior for Multilinear Kinematic Hardening (p. 79))
Σ_{wi} = the sum of the weighting factors for the previously evaluated subvolumes

Figure 4.5: Uniaxial Behavior for Multilinear Kinematic Hardening



The yield stress for each subvolume is given by

$$\sigma_{yk} = \frac{1}{2(1+\nu)}(3E\varepsilon_k - (1-2\nu)\sigma_k) \tag{4.46}$$

where $(\varepsilon_k, \sigma_k)$ is the breakpoint in the stress-strain curve. The number of subvolumes corresponds to the number of breakpoints specified.

The increment in plastic strain $\{\Delta\varepsilon_k^{pl}\}$ for each subvolume is computed using a von Mises yield criterion with the associated flow rule. The section on specialization for bilinear kinematic hardening is followed but since each subvolume is elastic-perfectly plastic, C and therefore $\{\alpha\}$ is zero.

The plastic strain increment for the entire volume is the sum of the subvolume increments:

$$\{\Delta\varepsilon^{pl}\} = \sum_{i=1}^{N_{sv}} w_i \{\Delta\varepsilon_i^{pl}\} \tag{4.47}$$

where:

N_{sv} = number of subvolumes

The current plastic strain and elastic strain can then be computed for the entire volume via Equation 4.20 (p. 72) and Equation 4.21 (p. 72).

The equivalent plastic strain $\hat{\varepsilon}^{pl}$ (output as EPEQ) is defined by Equation 4.43 (p. 77) and equivalent stress parameter $\hat{\sigma}_e^{pl}$ (output as SEPL) is computed by evaluating the input stress-strain curve at $\hat{\varepsilon}^{pl}$ (after adjusting the curve for the elastic strain component). The stress ratio N (output as SRAT, Equation 4.25 (p. 73)) is defined using the σ_e and σ_y values of the first subvolume.

4.2.12. Specialization for Nonlinear Kinematic Hardening

The material model considered is a rate-independent version of the nonlinear kinematic hardening model proposed by Chaboche([244] (p. 934), [245] (p. 934)) (accessed with **TB**,CHAB). The constitutive equations are based on linear isotropic elasticity, a von Mises yield function and the associated flow rule. Like the bilinear and multilinear kinematic hardening options, the model can be used to simulate the monotonic hardening and the Bauschinger effect. The model is also applicable to simulate the ratcheting effect of materials. In addition, the model allows the superposition of several kinematic models as well as isotropic hardening models. It is thus able to model the complicated cyclic plastic behavior of materials, such as cyclic hardening or softening and ratcheting or shakedown.

The model uses the von Mises yield criterion with the associated flow rule, the yield function is:

$$F = \left[\frac{3}{2}(\{s\}-\{a\})^T[M](\{s\}-\{\alpha\})\right]^{\frac{1}{2}} - R = 0 \tag{4.48}$$

where:

R = isotropic hardening variable

According to the normality rule, the flow rule is written:

$$\{\Delta\varepsilon^{pl}\} = \lambda\left\{\frac{\partial Q}{\partial\sigma}\right\} \tag{4.49}$$

where:

λ = plastic multiplier

The back stress $\{\alpha\}$ is superposition of several kinematic models as:

$$\{\alpha\} = \sum_{i=1}^{n}\{\alpha\}_i \tag{4.50}$$

where:

n = number of kinematic models to be superposed.

The evolution of the back stress (the kinematic hardening rule) for each component is defined as:

$$\{\Delta\alpha\}_i = \frac{2}{3}C_i\{\Delta\varepsilon^{pl}\} - \gamma_i\{\alpha\}_i\Delta\hat{\varepsilon}^{pl} + \frac{1}{C_i}\frac{dC_i}{d\theta}\Delta\theta\{\alpha\} \tag{4.51}$$

where:

C_i, γ_i, i = 1, 2, ... n = material constants for kinematic hardening

The associated flow rule yields:

$$\left\{\frac{\partial Q}{\partial\sigma}\right\} = \left\{\frac{\partial F}{\partial\sigma}\right\} = \frac{3}{2}\frac{\{s\}-\{\alpha\}}{\sigma_e} \tag{4.52}$$

The plastic strain increment, Equation 4.49 (p. 80) is rewritten as:

$$\{\Delta\varepsilon^{pl}\} = \frac{3}{2}\lambda\frac{\{s\}-\{\alpha\}}{\sigma_e} \tag{4.53}$$

The equivalent plastic strain increment is then:

$$\Delta\hat{\varepsilon}^{pl} = \sqrt{\frac{2}{3}\{\Delta\varepsilon^{pl}\}^T[M]\{\Delta\varepsilon^{pl}\}} = \lambda \tag{4.54}$$

The accumulated equivalent plastic strain is:

$$\hat{\varepsilon}^{pl} = \sum\Delta\hat{\varepsilon}^{pl} \tag{4.55}$$

The isotropic hardening variable, R, can be defined by:

$$R = k + R_o\,\hat{\varepsilon}^{pl} + R_\infty(1-e^{-b\hat{\varepsilon}^{pl}}) \tag{4.56}$$

where:

k = elastic limit

R_o, R_∞, b = material constants characterizing the material isotropic hardening behavior.

The material hardening behavior, R, in Equation 4.48 (p. 80) can also be defined through bilinear or multilinear isotropic hardening options, which have been discussed early in Specialization for Hardening (p. 74).

The return mapping approach with consistent elastoplastic tangent moduli that was proposed by Simo and Hughes([252] (p. 935)) is used for numerical integration of the constitutive equation described above.

4.2.13. Specialization for Anisotropic Plasticity

There are two anisotropic plasticity options in ANSYS. The first option uses Hill's([50] (p. 923)) potential theory (accessed by **TB**,HILL command). The second option uses a generalized Hill potential theory (Shih and Lee([51] (p. 923))) (accessed by **TB**, ANISO command).

4.2.14. Hill Potential Theory

The anisotropic Hill potential theory (accessed by **TB**,HILL) uses Hill's([50] (p. 923)) criterion. Hill's criterion is an extension to the von Mises yield criterion to account for the anisotropic yield of the material. When this criterion is used with the isotropic hardening option, the yield function is given by:

$$f\{\sigma\} = \sqrt{\{\sigma\}^T[M]\{\sigma\}} - \sigma_0(\bar{\varepsilon}^p) \tag{4.57}$$

where:

σ_0 = reference yield stress

$\bar{\varepsilon}^p$ = equivalent plastic strain

and when it is used with the kinematic hardening option, the yield function takes the form:

$$f\{\sigma\} = \sqrt{(\{\sigma\} - \{\alpha\})^T [M](\{\sigma\} - \{\alpha\})} - \sigma_0 \tag{4.58}$$

The material is assumed to have three orthogonal planes of symmetry. Assuming the material coordinate system is perpendicular to these planes of symmetry, the plastic compliance matrix [M] can be written as:

$$[M] = \begin{bmatrix} G+H & -H & -G & 0 & 0 & 0 \\ -H & F+H & -F & 0 & 0 & 0 \\ -G & -F & F+G & 0 & 0 & 0 \\ 0 & 0 & 0 & 2N & 0 & 0 \\ 0 & 0 & 0 & 0 & 2L & 0 \\ 0 & 0 & 0 & 0 & 0 & 2M \end{bmatrix} \tag{4.59}$$

F, G, H, L, M and N are material constants that can be determined experimentally. They are defined as:

$$F = \frac{1}{2}\left(\frac{1}{R_{yy}^2} + \frac{1}{R_{zz}^2} - \frac{1}{R_{xx}^2}\right) \tag{4.60}$$

$$G = \frac{1}{2}\left(\frac{1}{R_{zz}^2} + \frac{1}{R_{xx}^2} - \frac{1}{R_{yy}^2}\right) \tag{4.61}$$

$$H = \frac{1}{2}\left(\frac{1}{R_{xx}^2} + \frac{1}{R_{yy}^2} - \frac{1}{R_{zz}^2}\right) \tag{4.62}$$

$$L = \frac{3}{2}\left(\frac{1}{R_{yz}^2}\right) \tag{4.63}$$

$$M = \frac{3}{2}\left(\frac{1}{R_{xz}^2}\right) \tag{4.64}$$

$$N = \frac{3}{2}\left(\frac{1}{R_{xy}^2}\right) \tag{4.65}$$

The yield stress ratios R_{xx}, R_{yy}, R_{zz}, R_{xy}, R_{yz} and R_{xz} are specified by the user and can be calculated as:

$$R_{xx} = \frac{\sigma_{xx}^y}{\sigma_0} \tag{4.66}$$

$$R_{yy} = \frac{\sigma_{yy}^y}{\sigma_0} \tag{4.67}$$

$$R_{zz} = \frac{\sigma_{zz}^y}{\sigma_0} \tag{4.68}$$

$$R_{xy} = \sqrt{3}\,\frac{\sigma_{xy}^y}{\sigma_0} \tag{4.69}$$

$$R_{yz} = \sqrt{3}\frac{\sigma_{yz}^{y}}{\sigma_0} \tag{4.70}$$

$$R_{xz} = \sqrt{3}\frac{\sigma_{xz}^{y}}{\sigma_0} \tag{4.71}$$

where:

σ_{ij}^{y} = yield stress values

Two notes:

- The inelastic compliance matrix should be positive definite in order for the yield function to exist.
- The plastic slope (see also Equation 4.42 (p. 77)) is calculated as:

$$E^{pl} = \frac{E_x E_t}{E_x - E_t} \tag{4.72}$$

where:

E_x = elastic modulus in x-direction
E_t = tangent modulus defined by the hardening input

4.2.15. Generalized Hill Potential Theory

The generalized anisotropic Hill potential theory (accessed by **TB**,ANISO) uses Hill's([50] (p. 923)) yield criterion, which accounts for differences in yield strengths in orthogonal directions, as modified by Shih and Lee([51] (p. 923)) accounting for differences in yield strength in tension and compression. An associated flow rule is assumed and work hardening as presented by Valliappan et al.([52] (p. 923)) is used to update the yield criterion. The yield surface is therefore a distorted circular cylinder that is initially shifted in stress space which expands in size with plastic straining as shown in Figure 4.2: Various Yield Surfaces (p. 68) (b).

The equivalent stress for this option is redefined to be:

$$\sigma_e = \left(\frac{1}{3}\{\sigma\}^T[M]\{\sigma\} - \frac{1}{3}\{\sigma\}^T\{L\}\right)^{\frac{1}{2}} \tag{4.73}$$

where [M] is a matrix which describes the variation of the yield stress with orientation and {L} accounts for the difference between tension and compression yield strengths. {L} can be related to the yield surface translation $\{\alpha\}$ of Equation 4.36 (p. 76) (Shih and Lee([51] (p. 923))) and hence the equivalent stress function can be interpreted as having an initial translation or shift. When σ_e is equal to a material parameter K, the material is assumed to yield. The yield criterion Equation 4.7 (p. 70) is then

$$3F = \{\sigma\}^T[M]\{\sigma\} - \{\sigma\}^T\{L\} - K = 0 \tag{4.74}$$

The material is assumed to have three orthogonal planes of symmetry. The plastic behavior can then be characterized by the stress-strain behavior in the three element coordinate directions and the corresponding shear stress-shear strain behavior. Therefore [M] has the form:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & 0 & 0 & 0 \\ M_{12} & M_{22} & M_{23} & 0 & 0 & 0 \\ M_{13} & M_{23} & M_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & M_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & M_{66} \end{bmatrix} \tag{4.75}$$

By evaluating the yield criterion Equation 4.74 (p. 83) for all the possible uniaxial stress conditions the individual terms of [M] can be identified:

$$M_{jj} = \frac{K}{\sigma_{+j}\sigma_{-j}}, j = 1 \text{ to } 6 \tag{4.76}$$

where:

σ_{+j} and σ_{-j} = tensile and compressive yield strengths in direction j (j = x, y, z, xy, yz, xz)

The compressive yield stress is handled as a positive number here. For the shear yields, $\sigma_{+j} = \sigma_{-j}$. Letting $M_{11} = 1$ defines K to be

$$K = \sigma_{+x}\sigma_{-x} \tag{4.77}$$

The strength differential vector {L} has the form

$$\{L\} = \lfloor L_1 \quad L_2 \quad L_3 \quad 0 \quad 0 \quad 0 \rfloor^T \tag{4.78}$$

and from the uniaxial conditions {L} is defined as

$$L_j = M_{jj}(\sigma_{+j} - \sigma_{-j}), j = 1 \text{ to } 3 \tag{4.79}$$

Assuming plastic incompressibility (i.e. no increase in material volume due to plastic straining) yields the following relationships

$$\begin{aligned} M_{11} + M_{12} + M_{13} &= 0 \\ M_{12} + M_{22} + M_{23} &= 0 \\ M_{13} + M_{23} + M_{33} &= 0 \end{aligned} \tag{4.80}$$

and

$$L_1 + L_2 + L_3 = 0 \tag{4.81}$$

The off-diagonals of [M] are therefore

$$\begin{aligned} M_{12} &= -\frac{1}{2}(M_{11} + M_{22} - M_{33}) \\ M_{13} &= -\frac{1}{2}(M_{11} - M_{22} + M_{33}) \\ M_{23} &= -\frac{1}{2}(-M_{11} + M_{22} + M_{33}) \end{aligned} \tag{4.82}$$

Note that Equation 4.81 (p. 84) (by means of Equation 4.76 (p. 84) and Equation 4.79 (p. 84)) yields the consistency equation

$$\frac{\sigma_{+x} - \sigma_{-x}}{\sigma_{+x}\sigma_{-x}} + \frac{\sigma_{+y} - \sigma_{-y}}{\sigma_{+y}\sigma_{-y}} + \frac{\sigma_{+z} - \sigma_{-z}}{\sigma_{+z}\sigma_{-z}} = 0 \tag{4.83}$$

that must be satisfied due to the requirement of plastic incompressibility. Therefore the uniaxial yield strengths are not completely independent.

The yield strengths must also define a closed yield surface, that is, elliptical in cross section. An elliptical yield surface is defined if the following criterion is met:

$$M_{11}^2 + M_{22}^2 + M_{33}^2 - 2(M_{11}M_{22} + M_{22}M_{33} + M_{11}M_{33}) < 0 \tag{4.84}$$

Otherwise, the following message is output: "THE DATA TABLE DOES NOT REPRESENT A CLOSED YIELD SURFACE. THE YIELD STRESSES OR SLOPES MUST BE MADE MORE EQUAL". This further restricts the independence of the uniaxial yield strengths. Since the yield strengths change with plastic straining (a consequence of work hardening), this condition must be satisfied throughout the history of loading. The program checks this condition through an equivalent plastic strain level of 20% (.20).

For an isotropic material,

$$\begin{aligned} &M_{11} = M_{22} = M_{33} = 1 \\ &M_{12} = M_{13} = M_{23} = -1/2 \\ &M_{44} = M_{55} = M_{66} = 3 \end{aligned} \tag{4.85}$$

and

$$L_1 = L_2 = L_3 = 0 \tag{4.86}$$

and the yield criterion (Equation 4.74 (p. 83) reduces down to the von Mises yield criterion

Equation 4.38 (p. 76) with $\{\alpha\} = 0$).

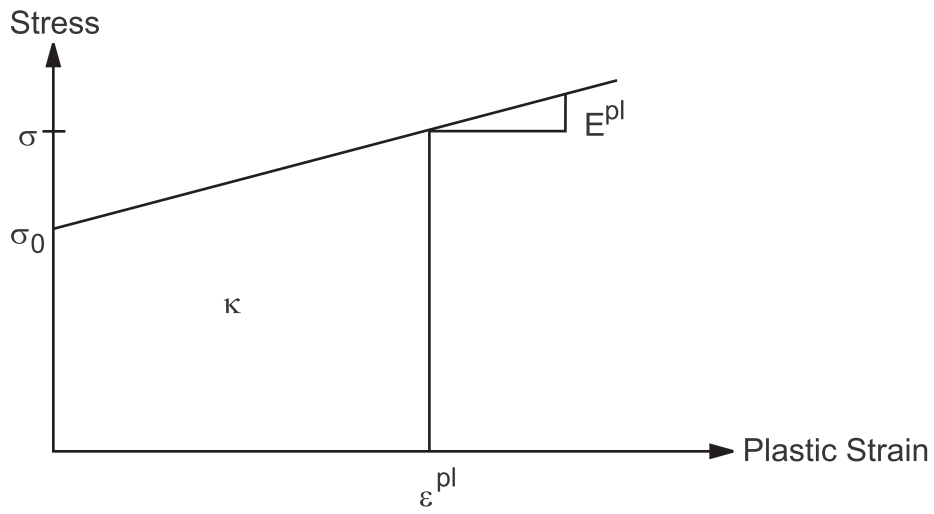
Work hardening is used for the hardening rule so that the subsequent yield strengths increase with increasing total plastic work done on the material. The total plastic work is defined by Equation 4.23 (p. 73) where the increment in plastic work is

$$\Delta\kappa = \{\overset{*}{\sigma}\}\{\Delta\varepsilon^{pl}\} \tag{4.87}$$

where:

$\{\overset{*}{\sigma}\}$ = average stress over the increment

Figure 4.6: Plastic Work for a Uniaxial Case



For the uniaxial case the total plastic work is simply

$$\kappa = \frac{1}{2}\varepsilon^{pl}(\sigma_o + \sigma) \tag{4.88}$$

where the terms are defined as shown in Figure 4.6: Plastic Work for a Uniaxial Case (p. 86).

For bilinear stress-strain behavior,

$$\sigma = \sigma_o + E^{pl}\varepsilon^{pl} \tag{4.89}$$

where:

$E^{pl} = \frac{EE_T}{E - E_T}$ = plastic slope (see also Equation 4.42 (p. 77))

E = elastic modulus

E_T = tangent moulus

$$E^{pl} = \frac{EE_T}{E - E_T} \tag{4.90}$$

Combining Equation 4.89 (p. 86) with Equation 4.88 (p. 86) and solving for the updated yield stress σ:

$$\sigma = \{2E^{pl}\kappa + \sigma_o^2\}^{\frac{1}{2}} \tag{4.91}$$

Extending this result to the anisotropic case gives,

$$\sigma_j = \{2E_j^{pl}\kappa + \sigma_{oj}^2\}^{\frac{1}{2}} \tag{4.92}$$

where j refers to each of the input stress-strain curves. Equation 4.92 (p. 86) determines the updated yield stresses by equating the amount of plastic work done on the material to an equivalent amount of plastic work in each of the directions.

The parameters [M] and {L} can then be updated from their definitions Equation 4.76 (p. 84) and Equation 4.79 (p. 84) and the new values of the yield stresses. For isotropic materials, this hardening rule reduces to the case of isotropic hardening.

The equivalent plastic strain $\hat{\varepsilon}^{pl}$ (output as EPEQ) is computed using the tensile x direction as the reference axis by substituting Equation 4.89 (p. 86) into Equation 4.88 (p. 86):

$$\hat{\varepsilon}^{pl} = \frac{-\sigma_{+x} + (\sigma_{+x}^2 + 2\kappa E_{+x}^{pl})^{\frac{1}{2}}}{E_{+x}^{pl}} \tag{4.93}$$

where the yield stress in the tensile x direction σ_{+x} refers to the initial (not updated) yield stress. The equivalent stress parameter $\hat{\sigma}_e^{pl}$ (output as SEPL) is defined as

$$\hat{\sigma}_e^{pl} = \sigma_{+x}^{pl} + \hat{E}_{+x}^{pl}\,\varepsilon \tag{4.94}$$

where again σ_{+x} is the initial yield stress.

4.2.16. Specialization for Drucker-Prager

The following topics concerning Drucker-Prager are available:
4.2.16.1. Classic Drucker-Prager Model
4.2.16.2. Extended Drucker-Prager Model

4.2.16.1. Classic Drucker-Prager Model

This option uses the Drucker-Prager yield criterion with either an associated or nonassociated flow rule (accessed with **TB**,DP). The yield surface does not change with progressive yielding, hence there is no hardening rule and the material is elastic- perfectly plastic (Figure 4.1: Stress-Strain Behavior of Each of the Plasticity Options (p. 67) (f) Drucker-Prager). The equivalent stress for Drucker-Prager is

$$\sigma_e = 3\beta\sigma_m + \left[\frac{1}{2}\{s\}^T[M]\{s\}\right]^{\frac{1}{2}} \tag{4.95}$$

where:

σ_m = mean or hydrostatic stress $= \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z)$
{s} = deviatoric stress Equation 4.37 (p. 76)
β = material constant
[M] = as defined with Equation 4.36 (p. 76)

This is a modification of the von Mises yield criterion (Equation 4.36 (p. 76) with $\{\alpha\} = \{0\}$) that accounts for the influence of the hydrostatic stress component: the higher the hydrostatic stress (confinement pressure) the higher the yield strength. β is a material constant which is given as

$$\beta = \frac{2\sin\phi}{\sqrt{3}(3 - \sin\phi)} \tag{4.96}$$

where:

ϕ = input angle of internal friction

The material yield parameter is defined as

$$\sigma_y = \frac{6c\cos\phi}{\sqrt{3}(3-\sin\phi)} \tag{4.97}$$

where:

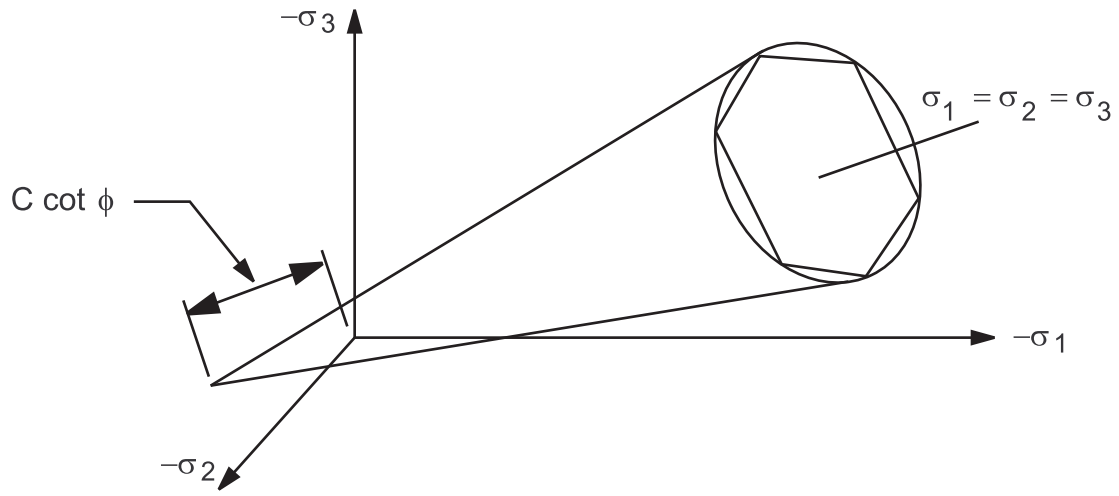
c = input cohesion value

The yield criterion Equation 4.7 (p. 70) is then

$$F = 3\beta\sigma_m + \left[\frac{1}{2}\{s\}^T[M]\{s\}\right]^{\frac{1}{2}} - \sigma_y = 0 \tag{4.98}$$

This yield surface is a circular cone (Figure 4.2: Various Yield Surfaces (p. 68)-c) with the material parameters Equation 4.96 (p. 87) and Equation 4.97 (p. 88) chosen such that it corresponds to the outer aspices of the hexagonal Mohr-Coulomb yield surface, Figure 4.7: Drucker-Prager and Mohr-Coulomb Yield Surfaces (p. 88).

Figure 4.7: Drucker-Prager and Mohr-Coulomb Yield Surfaces



$\{\frac{\partial F}{\partial \sigma}\}$ is readily computed as

$$\left\{\frac{\partial F}{\partial \sigma}\right\} = \beta\lfloor 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \rfloor^T + \frac{1}{\left[\frac{1}{2}\{s\}^T[M]\{s\}\right]^{\frac{1}{2}}}\{s\} \tag{4.99}$$

$\{\frac{\partial Q}{\partial \sigma}\}$ is similar, however β is evaluated using ϕ_f (the input "dilatancy" constant). When $\phi_f = \phi$, the flow rule is associated and plastic straining occurs normal to the yield surface and there will be a volumetric

expansion of the material with plastic strains. If ϕ_f is less than ϕ there will be less volumetric expansion and if ϕ_f is zero, there will be no volumetric expansion.

The equivalent plastic strain $\hat{\varepsilon}^{pl}$ (output as EPEQ) is defined by Equation 4.43 (p. 77) and the equivalent stress parameter $\hat{\sigma}_e^{pl}$ (output as SEPL) is defined as

$$\sigma_e^{pl} = \sqrt{3}(\sigma_y - 3\beta\sigma_m) \tag{4.100}$$

The equivalent stress parameter is interpreted as the von Mises equivalent stress at yield at the current hydrostatic stress level. Therefore for any integration point undergoing yielding (stress ratio (output as SRAT) >1), $\hat{\sigma}_e^{pl}$ should be close to the actual von Mises equivalent stress (output as SIGE) at the converged solution.

4.2.16.2. Extended Drucker-Prager Model

This option is an extension of the linear Drucker-Prager yield criterion (input with **TB**,EDP). Both yield surface and the flow potential, (input with *TB0PT* on **TB**,EDP command) can be taken as linear, hyperbolic and power law independently, and thus results in either an associated or nonassociated flow rule. The yield surface can be changed with progressive yielding of the isotropic hardening plasticity material options, see hardening rule Figure 4.1: Stress-Strain Behavior of Each of the Plasticity Options (p. 67) (c) Bilinear Isotropic and (d) Multilinear Isotropic.

The yield function with linear form (input with *TB0PT* = LYFUN) is:

$$F = q + \alpha\sigma_m - \sigma_Y(\hat{\varepsilon}_{pl}) = 0 \tag{4.101}$$

where:

α = material parameter referred to pressure sensitive parameter (input as C1 on **TBDATA** command using **TB**,EDP)

$q = \sqrt{\frac{3}{2}s:s}$ where q is the deviatoric Cauchy stress tensor

$\sigma_Y(\hat{\varepsilon}_{pl})$ = yield stress of material (input as C2 on **TBDATA** command or input using **TB**,MISO; **TB**,BISO; **TB**,NLISO; or **TB**,PLAST)

The yield function with hyperbolic form (input with *TB0PT* = HYFUN) is:

$$\sqrt{a^2 + q^2} + \alpha\sigma_m - \sigma_Y(\hat{\varepsilon}_{pl}) = 0 \tag{4.102}$$

where:

a = material parameter characterizing the shape of yield surface (input as C2 on **TBDATA** command using **TB**,EDP)

The yield function with power law form (input with *TB0PT* = PYFUN) is:

$$q^b + \alpha\sigma_m - \sigma_Y{}^b(\hat{\varepsilon}_{pl}) = 0 \tag{4.103}$$

where:

b = material parameter characterizing the shape of yield surface (input as C2 on **TBDATA** command using **TB**,EDP):

Similarly, the flow potential Q for linear form (input with *TB0PT* = LFPOT) is:

$$Q = q + \alpha\sigma_m - \sigma_Y(\hat{\varepsilon}_{pl}) \tag{4.104}$$

The flow potential Q for hyperbolic form (input with *TB0PT* = HFPOT) is:

$$Q = \sqrt{a^2 + q^2} + \alpha\sigma_m - \sigma_Y(\hat{\varepsilon}_{pl}) \tag{4.105}$$

The flow potential Q for power law form (input with *TB0PT* = PFPOT) is:

$$Q = q^b + \alpha\sigma_m - \sigma_Y{}^b(\hat{\varepsilon}_{pl}) \tag{4.106}$$

The plastic strain is defined as:

$$\dot{\varepsilon}_{pl} = \dot{\lambda}\frac{\partial Q}{\partial \sigma} \tag{4.107}$$

where:

$\dot{\lambda}$ = plastic multiplier

When the flow potential is the same as the yield function, the plastic flow rule is associated, which in turn results in a symmetric stiffness matrix. When the flow potential is different from the yield function, the plastic flow rule is nonassociated, and this results in an unsymmetric material stiffness matrix. By default, the unsymmetric stiffness matrix (accessed by **NROPT**,UNSYM) will be symmetricized.

4.2.17. Extended Drucker-Prager Cap Model

The cap model focuses on geomaterial plasticity resulting from compaction at low mean stresses followed by significant dilation before shear failure ([386] (p. 942)). A three-invariant cap plasticity model with three smooth yielding surfaces including a compaction cap, an expansion cap, and a shear envelope proposed by Pelessone ([387] (p. 942)) is described here.

Geomaterials typically have much higher triaxial strength in compression than in tension. The cap model accounts for this by incorporating the third-invariant of stress tensor (J_3) into the yielding functions.

Introduced first are functions to be used in the cap model, including shear failure envelope function, compaction cap function, expansion cap function, the Lode angle function, and hardening functions. Then, a unified yielding function for the cap model that is able to describe all the behaviors of shear, compaction, and expansion yielding surfaces is derived using the shear failure envelope and cap functions.

The following topics are covered:

4.2.17.1. Shear Failure Envelope Function

A typical geomaterial shear envelope function is based on the exponential format given below:

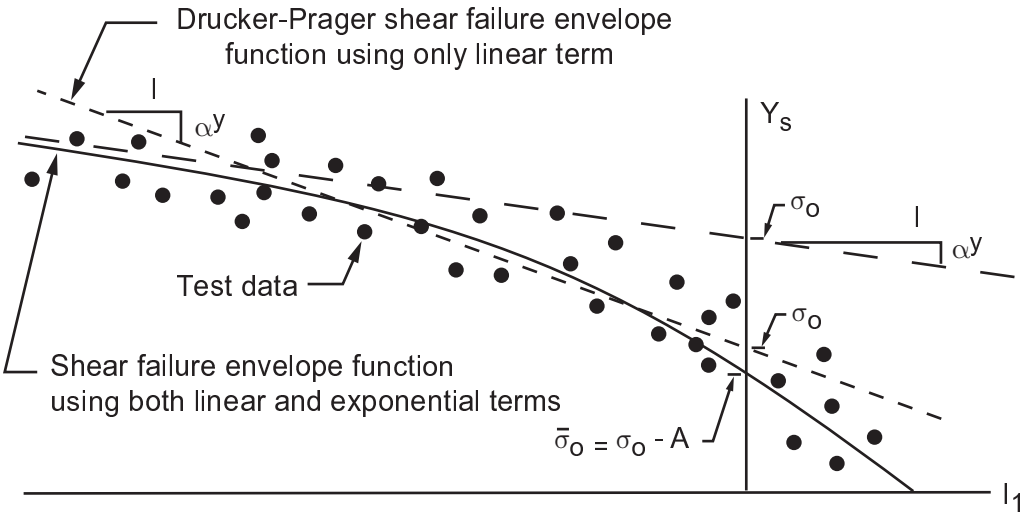
$$Y_s(I_1,\sigma_0) = \sigma_0 - Ae^{(\beta^y I_1)} - \alpha^y I_1 \tag{4.108}$$

where:

I_1 = first invariant of Cauchy stress tensor
subscript "s" = shear envelope function
superscript "y" = yielding related material constants
σ_0 = current cohesion-related material constant (input using **TB**,EDP with *TB0PT* = CYFUN)
A, β^y, α^y = material constants (input using **TB**,EDP with *TB0PT* = CYFUN)

Equation 4.108 (p. 91) reduces to the Drucker-Prager yielding function if parameter "A" is set to zero. It should be noted that all material constants in Equation 4.108 (p. 91) are defined based on I_1 and J_2, which are different from those in the previous sections. The effect of hydrostatic pressure on material yielding may be exaggerated at high pressure range by only using the linear term (Drucker-Prager) in Equation 4.108 (p. 91). Such an exaggeration is reduced by using both the exponential term and linear term in the shear function. Figure 4.8: Shear Failure Envelope Functions (p. 91) shows the configuration of the shear function. In Figure 4.8: Shear Failure Envelope Functions (p. 91) the dots are the testing data points, the finer dashed line is the fitting curve based on the Drucker-Prager linear yielding function, the solid curved line is the fitting curve based on Equation 4.108 (p. 91), and the coarser dashed line is the limited state of Equation 4.108 (p. 91) at very high pressures. In the figure $\bar{\sigma}_0 = \sigma_0 - A$ is the current modified cohesion obtained by setting I_1 in Equation 4.108 (p. 91) to zero.

Figure 4.8: Shear Failure Envelope Functions



4.2.17.2. Compaction Cap Function

The compaction cap function is formulated using the shear envelope function defined in Equation 4.108 (p. 91).

$$Y_c(I_1,K_0,\sigma_0) = 1 - H(K_0 - I_1)\left(\frac{I_1 - K_0}{R_c^y Y_s(K_0,\sigma_0)}\right)^2 \tag{4.109}$$

where:

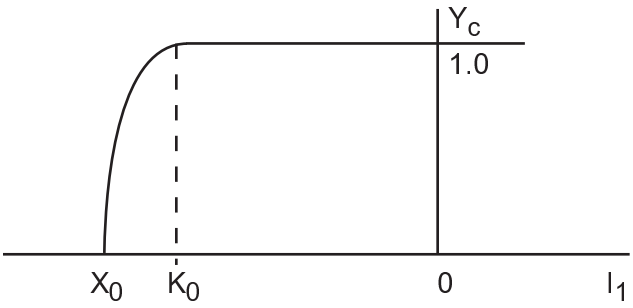
H = Heaviside (or unit step) function
subscript "c" = compaction cap-related function or constant
R = ratio of elliptical x-axis to y-axis (I_1 to J_2)
K_0 = key flag indicating the current transition point at which the compaction cap surface and shear portion intersect.

In Equation 4.109 (p. 92), Y_c is an elliptical function combined with the Heaviside function. Y_c is plotted in Figure 4.9: Compaction Cap Function (p. 92).

This function implies:

1. When I_1, the first invariant of stress, is greater than K_0, the compaction cap takes no effect on yielding. The yielding may happen in either shear or expansion cap portion.
2. When I_1 is less than K_0, the yielding may only happen in the compaction cap portion, which is shaped by both the shear function and the elliptical function.

Figure 4.9: Compaction Cap Function



4.2.17.3. Expansion Cap Function

Similarly, Y_t is an elliptical function combined with the Heaviside function designed for the expansion cap. Y_t is shown in Figure 4.10: Expansion Cap Function (p. 93).

$$Y_t(I_1,\sigma_0) = 1 - H(I_1)\left(\frac{I_1}{R_t^y Y_s(0,\sigma_0)}\right)^2 \tag{4.110}$$

where:

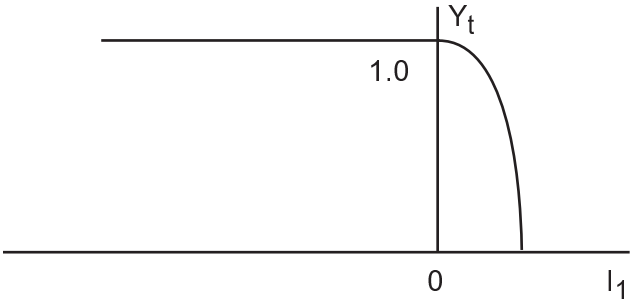
subscript "t" = expansion cap-related function or constant

This function implies that:

1. When I_1 is negative, the yielding may happen in either shear or compaction cap portion, while the tension cap has no effect on yielding.
2. When I_1 is positive, the yielding may only happen in the tension cap portion. The tension cap is shaped by both the shear function and by another elliptical function.

Equation 4.110 (p. 92) assumes that Y_t is only a function of σ_0 and not a function of K_0 as I_1 is set to zero in function Y_s.

Figure 4.10: Expansion Cap Function



4.2.17.4. Lode Angle Function

Unlike metals, the yielding and failure behaviors of geomaterials are affected by their relatively weak (compared to compression) tensile strength. The ability of a geomaterial to resist yielding is lessened by nonuniform stress states in the principal directions. The effect of reduced yielding capacity for such geomaterials is described by the Lode angle β and the ratio ψ of triaxial extension strength to compression strength. The Lode angle β can be written in a function of stress invariants J_2 and J_3:

$$\beta(J_2, J_3) = -\frac{1}{3}\sin^{-1}\left(\frac{3\sqrt{3}\,J_3}{2J_2^{3/2}}\right) \tag{4.111}$$

where:

J_2 and J_3 = second and third invariants of the deviatoric tensor of the Cauchy stress tensor.

The Lode angle function Γ is defined by:

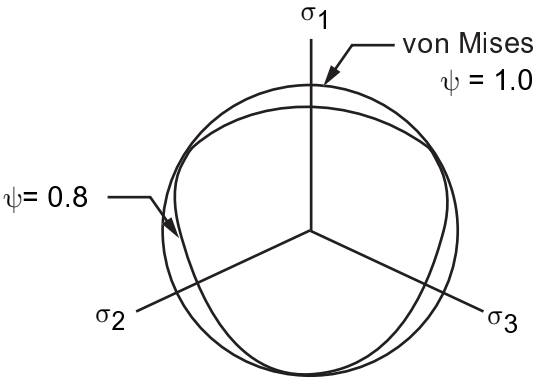
$$\Gamma(\beta, \psi) = \frac{1}{2}(1 + \sin 3\beta + \frac{1}{\psi}(1 - \sin 3\beta)) \tag{4.112}$$

where:

ψ = ratio of triaxial extension strength to compression strength

The three-invariant plasticity model is formulated by multiplying J_2 in the yielding function by the Lode angle function described by Equation 4.112 (p. 93). The profile of the yielding surface in a three-invariant plasticity model is presented in Figure 4.11: Yielding Surface in π-Plane (p. 94).

Figure 4.11: Yielding Surface in π-Plane



4.2.17.5. Hardening Functions

The cap hardening law is defined by describing the evolution of the parameter X_0, the intersection point of the compaction cap and the I_1axis. The evolution of X_0 is related only to the plastic volume strain ε_v^p. A typical cap hardening law has the exponential form proposed in Fossum and Fredrich([92] (p. 925)):

$$\varepsilon_v^p = W_1^c\{e^{(D_1^c - D_2^c(X_0 - X_i))(X_0 - X_i)} - 1\} \tag{4.113}$$

where:

X_i = initial value of X_0 at which the cap takes effect in the plasticity model.

W_1^c = maximum possible plastic volumetric strain for geomaterials.

Parameters D_1^c and D_2^c have units of 1 / (Force / Length) and 1 / (Force / Length)^2, respectively. All constants in Equation 4.113 (p. 94) are non-negative.

Besides cap hardening, another hardening law defined for the evolution of the cohesion parameter used in the shear portion described in Equation 4.108 (p. 91) is considered. The evolution of the modified cohesion $\bar{\sigma}_0$ is assumed to be purely shear-related and is the function of the effective deviatoric plastic strain γ_p:

$$\bar{\sigma}_0 = \sigma_0 - A = \bar{\sigma}_0(\gamma^p) \tag{4.114}$$

The effective deviatoric plastic strain γ_p is defined by its rate change as follows:

$$\dot{\gamma}^p = \{\frac{2}{3}(\dot{\varepsilon}^p - \frac{1}{3}\dot{\varepsilon}_v^p \mathbf{I}) : (\dot{\varepsilon}^p - \frac{1}{3}\dot{\varepsilon}_v^p \mathbf{I})\}^{\frac{1}{2}} \tag{4.115}$$

where:

ε_p = plastic strain tensor

"$\dot{\bullet}$" = rate change of variables

I = second order identity tensor

The unified and compacted yielding function for the cap model with three smooth surfaces is formulated using above functions as follows:

$$\begin{aligned} Y(\sigma, K_0, \sigma_0) &= Y(I_1, J_2, J_3, K_0, \sigma_0) \\ &= \Gamma^2(\beta, \psi) J_2 - Y_c(I_1, K_0, \sigma_0) Y_t(I_1, \sigma_0) Y_s^2(I_1, \sigma_0) \end{aligned} \tag{4.116}$$

where:

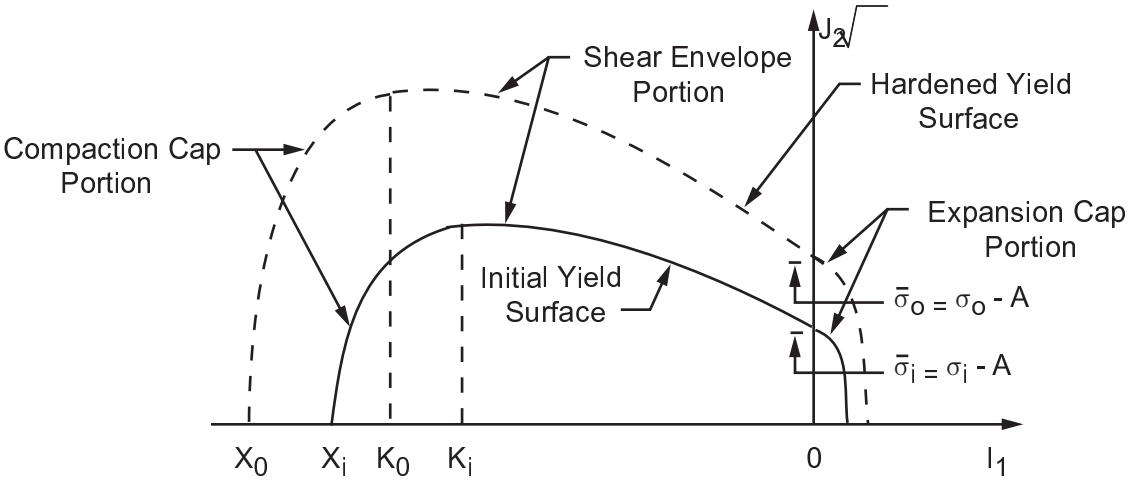
K_0 = function of both X_0 and σ_0

Again, the parameter X_0 is the intersection point of the compaction cap and the I_1 axis. The parameter K_0 is the state variable and can be implicitly described using X_0 and σ_0 given below:

$$K_0 = X_0 + R_c^y Y_s(K_0, \sigma_0) \tag{4.117}$$

The yielding model described in Equation 4.116 (p. 95) is used and is drawn in the J_2 and I_1 plane in Figure 4.12: Cap Model (p. 95).

Figure 4.12: Cap Model



The cap model also allows non-associated models for all compaction cap, shear envelope, and expansion cap portions. The nonassociated models are defined through using the yielding functions in Equation 4.116 (p. 95) as its flow potential functions, while providing different values for some material constants. It is written below:

$$\begin{aligned} F(\sigma, K_0, \sigma_0) &= F(I_1, J_2, J_3, K_0, \sigma_0) \\ &= \Gamma^2(\beta, \psi) J_2 - F_c(I_1, K_0, \sigma_0) F_t(I_1, \sigma_0) F_s^2(I_1, \sigma_0) \end{aligned} \tag{4.118}$$

where:

$$F_s(I_1,\sigma_0) = \sigma_0 - Ae^{(\beta^f I_1)} - \alpha^f I_1$$
$$F_c(I_1,K_0,\sigma_0) = 1 - H(K_0 - I_1)\left(\frac{I_1 - K_0}{R_c^f F_s(K_0,\sigma_0)}\right)^2 \quad (4.119)$$
$$F_t(I_1,\sigma_0) = 1 - H(I_1)\left(\frac{I_1}{R_t^f F_s(0,\sigma_0)}\right)^2$$

where:

superscript "f" = flow-related material constant

This document refers to the work of Schwer ([388] (p. 942)) and Foster ([389] (p. 943)) for the numerical formulations used in the Pelessone ([387] (p. 942)) smooth cap plasticity model. ANSYS, Inc. developed a new material integrator that is able to achieve a faster convergence rate for the transition zone between the cap and shear portions.

The flow functions in Equation 4.118 (p. 95) and Equation 4.119 (p. 96) are obtained by replacing β^y, α^y, R_c^y, and R_t^y in Equation 4.116 (p. 95) and Equation 4.117 (p. 95) with β^f, α^f, R_c^f, and R_t^f. The nonassociated cap model is input by using **TB**,EDP with *TBOPT* = CFPOT.

Shear hardening can be taken into account on by providing $\bar{\sigma}_0$ (via **TB**,MISO, **TB**,BISO, **TB**,NLISO, or **TB**,PLAS). The initial value of $\bar{\sigma}_0$ must be consistent to σ_i - A. This input regulates the relationship between the modified cohesion and the effective deviatoric plastic strain.

As the smooth models have more numerical advantages, it is often necessary to transfer nonsmooth caps such as the Sandler model ([386] (p. 942)) to a smooth model. To facilitate the model transformation from the nonsmooth cap model to the Pelessone smooth cap model ([387] (p. 942)), two simple and robust methods are used ([390] (p. 943)); rather than solving a group of nonlinear equations, the ANSYS, Inc. implementation solves only one scalar nonlinear equation.

Calibrating CAP Constants

Calibrating the CAP constants σ_i, β^Y, A, α^Y, β^Y, α^F and the hardening input for $\bar{\sigma}_0$ differs significantly from the other EDP options. The CAP parameters are all defined in relation to I_1 and I_2, while the other EDP coefficients are defined according to p and q.

4.2.18. Gurson's Model

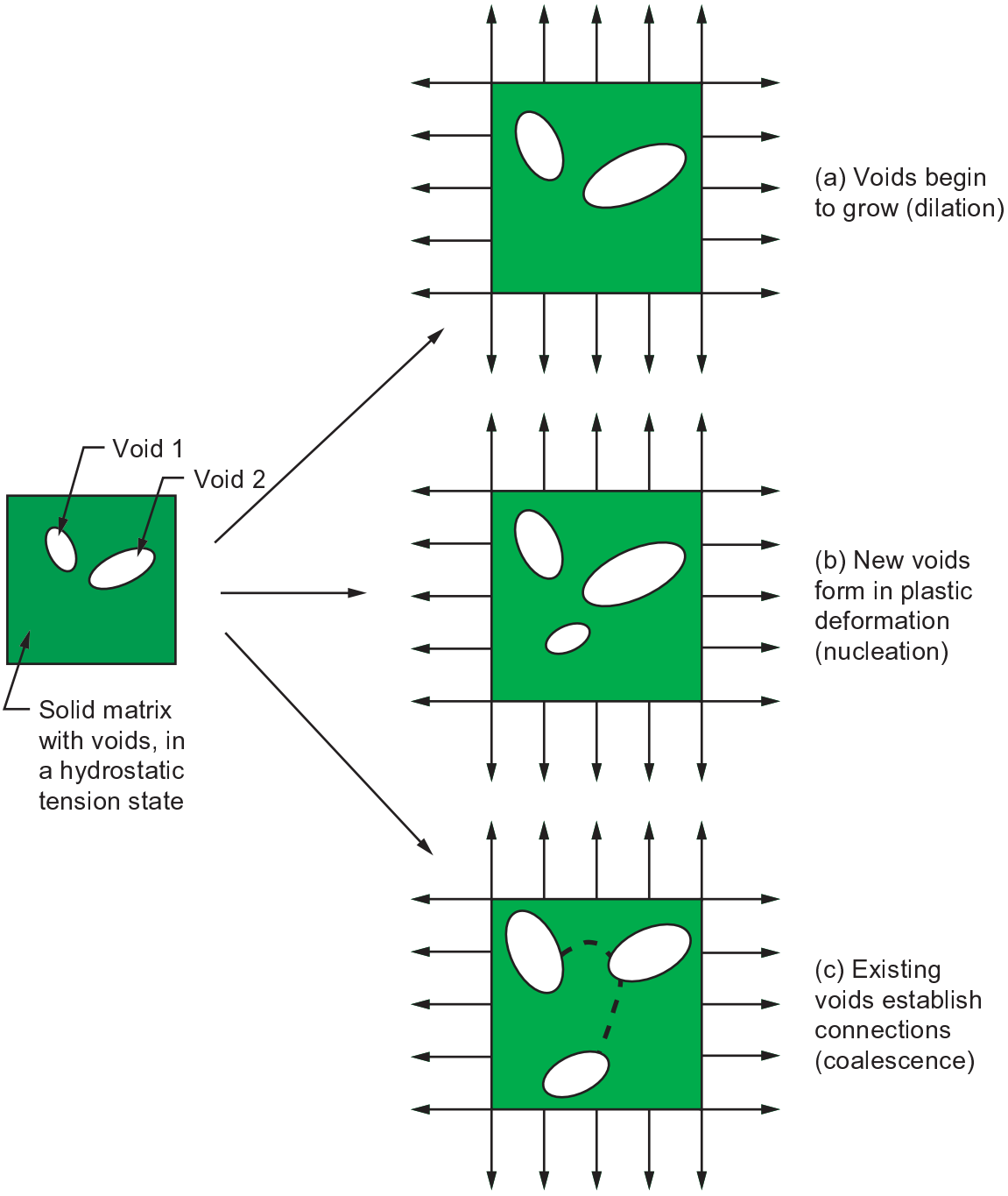
The Gurson Model is used to represent plasticity and damage in ductile porous metals. The model theory is based on Gurson([366] (p. 941)) and Tvergaard and Needleman([367] (p. 941)). When plasticity and damage occur, ductile metal goes through a process of void growth, nucleation, and coalescence. Gurson's method models the process by incorporating these microscopic material behaviors into macroscopic plasticity behaviors based on changes in the void volume fraction (porosity) and pressure. A porosity index increase corresponds to an increase in material damage, which implies a diminished material load-carrying capacity.

The microscopic porous metal representation in Figure 4.13: Growth, Nucleation, and Coalescence of Voids in Microscopic Scale (p. 98)(a), shows how the existing voids dilate (a phenomenon, called void growth) when the solid matrix is in a hydrostatic-tension state. The solid matrix portion is assumed to be incompressible when it yields, therefore any material volume growth (solid matrix plus voids) is due solely to the void volume expansion.

The second phenomenon is void nucleation which means that new voids are created during plastic deformation. Figure 4.13: Growth, Nucleation, and Coalescence of Voids in Microscopic Scale (p. 98)(b), shows the nucleation of voids resulting from the debonding of the inclusion-matrix or particle-matrix interface, or from the fracture of the inclusions or particles themselves.

The third phenomenon is the coalescence of existing voids. In this process, shown in Figure 4.13: Growth, Nucleation, and Coalescence of Voids in Microscopic Scale (p. 98)(c), the isolated voids establish connections. Although coalescence may not discernibly affect the void volume, the load carrying capacity of this material begins to decay more rapidly at this stage.

Figure 4.13: Growth, Nucleation, and Coalescence of Voids in Microscopic Scale



The evolution equation of porosity is given by

$$\dot{f} = \dot{f}_{growth} + \dot{f}_{nucleation} \tag{4.120}$$

where:

f = porosity

$\dot{}$ = rate change of variables

The evolution of the microscopic equivalent plastic work is:

$$\dot{\bar{\varepsilon}}^p = \frac{\sigma : \dot{\varepsilon}^p}{(1-f)\sigma_Y} \tag{4.121}$$

where:

$\bar{\varepsilon}^p$ = microscopic equivalent plastic strain
σ = Cauchy stress
: = inner product operator of two second order tensors
ε^p = macroscopic plastic strain
σ_Y = current yielding strength

The evolution of porosity related to void growth and nucleation can be stated in terms of the microscopic equivalent plastic strain, as follows:

$$\dot{f}_{growth} = (1-f)\dot{\varepsilon}^p : I \tag{4.122}$$

where:

I = second order identity tensor

The void nucleation is controlled by either the plastic strain or stress, and is assumed to follow a normal distribution of statistics. In the case of strain-controlled nucleation, the distribution is described in terms of the mean strain and its corresponding deviation. In the case of stress-controlled nucleation, the distribution is described in terms of the mean stress and its corresponding deviation. The porosity rate change due to nucleation is then given as follows:

$$\dot{f}_{nucleation} = \begin{cases} \dfrac{f_N \dot{\bar{\varepsilon}}^p}{S_N\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\bar{\varepsilon}^p - \varepsilon_N}{S_N}\right)^2} & \text{strain-controlled} \\ \dfrac{f_N(\dot{\sigma}_Y + \dot{p})}{S_N^\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\sigma_Y + p - \sigma_N}{S_N^\sigma}\right)^2} & \text{stress-controlled} \end{cases} \tag{4.123}$$

where:

f_N = volume fraction of the segregated inclusions or particles
ε_N = mean strain
S_N = strain deviation
σ_N = mean stress

S_N^σ = stress deviation (scalar with stress units)

$p = \frac{1}{3}\sigma : I$ = pressure

It should be noted that "stress controlled nucleation" means that the void nucleation is determined by the maximum normal stress on the interfaces between inclusions and the matrix. This maximum normal stress is measured by $\sigma_Y + p$. Thus, more precisely, the "stress" in the mean stress σ_N refers to $\sigma_Y + p$. This relationship better accounts for the effect of triaxial loading conditions on nucleation.

Given Equation 4.120 (p. 98) through Equation 4.123 (p. 99), the material yielding rule of the Gurson model is defined as follows:

$$\phi = \left(\frac{q}{\sigma_Y}\right)^2 + 2f^* q_1 \cosh\left(\frac{3}{2}\frac{q_2 p}{\sigma_Y}\right) - (1 + q_3 f^{*2}) = 0 \tag{4.124}$$

where:

q_1, q_2, and q_3 = Tvergaard-Needleman constants
σ_Y = yield strength of material
$q = \sqrt{\frac{3}{2}(\sigma - pI) : (\sigma - pI)}$ = equivalent stress

f*, the Tvergaard-Needleman function is:

$$f^*(f) = \begin{cases} f & \text{if} \le f_c \\ f_c + \dfrac{\frac{1}{q_1} - f_c}{f_F - f_c}(f - f_c) & \text{if} > f_c \end{cases} \tag{4.125}$$

where:

f_c = critical porosity
f_F = failure porosity

The Tvergaard-Needleman function is used to model the loss of material load carrying capacity, which is associated with void coalescence. When the current porosity f reaches a critical value f_c, the material load carrying capacity decreases more rapidly due to the coalescence. When the porosity f reaches a higher value f_F, the material load carrying capacity is lost completely. The associative plasticity model for the Gurson model has been implemented.

4.2.19. Gurson Plasticity with Isotropic/Chaboche Kinematic Hardening

Gurson's model accounts for hydrostatic pressure and material isotropic hardening effects on porous metals. The Chaboche model is essentially the von Mises plasticity model incorporating Chaboche-type kinematic hardening.

The Gurson-Chaboche model, also used for modeling porous metals, is an extension of the Gurson model, combining both material isotropic and kinematic hardening. The model is based on the work of Arndt, Svendsen and Klingbeil ([384] (p. 942)), and Besson and Guillemer-Neel ([385] (p. 942)).

The yielding function of the Gurson-Tvergaard and Needleman model is modified as follows:

$$Y = \frac{\bar{q}^2}{\sigma_m^2} + 2q_1 f^* \cosh\left(\frac{3}{2} q_2 \frac{p}{\sigma_a}\right) - 1 - q_3 f^{*2} \tag{4.126}$$

where:

$$\bar{q} = \sqrt{\frac{3}{2}(M - \bar{p}I) : (M - \bar{p}I)}$$

$$M = \sigma - X_{eff}$$

$$\bar{p} = \frac{1}{3}\text{trace}(M)$$

$$p = \frac{1}{3}\text{trace}(\sigma)$$

$$\sigma_a = \sigma_m + q(X)$$

σ_m = uniaxial yielding strength

$$q(X) = \sqrt{\frac{3}{2}\{X - \frac{1}{3}\text{trace}(X)\} : \{X - \frac{1}{3}\text{trace}(X)I\}}$$

$$X_{eff} = (1 - \sqrt{q_3}f^*)X$$

X_{eff} is the total effective back stress, and the total back stress X is the summation of several sub-back stresses:

$$X = \sum_{i=1}^{n} X_i$$

where the evolution of each sub-back stress is defined by the Chaboche kinematical law:

$$\dot{X}_i = \frac{2}{3}C_i\dot{\varepsilon}^p - \gamma_i\dot{\bar{\varepsilon}}^m X_i + \dot{d}_i(T)X_i \tag{4.127}$$

where T is the temperature and:

$$\dot{d}_i(T) = \frac{1}{C_i}\frac{dC_i}{dT}\dot{T}$$

The evolution of the equivalent plastic strain $\dot{\bar{\varepsilon}}^m$ is defined through:

$$\dot{\bar{\varepsilon}}^m = \frac{M : \dot{\varepsilon}^p}{(1-f)\sigma_m}$$

This model first requires the input parameters for Gurson plasticity with isotropic hardening, and then additional input parameters for Chaboche kinematic hardening. For more information, see Hardening in the *Material Reference*.

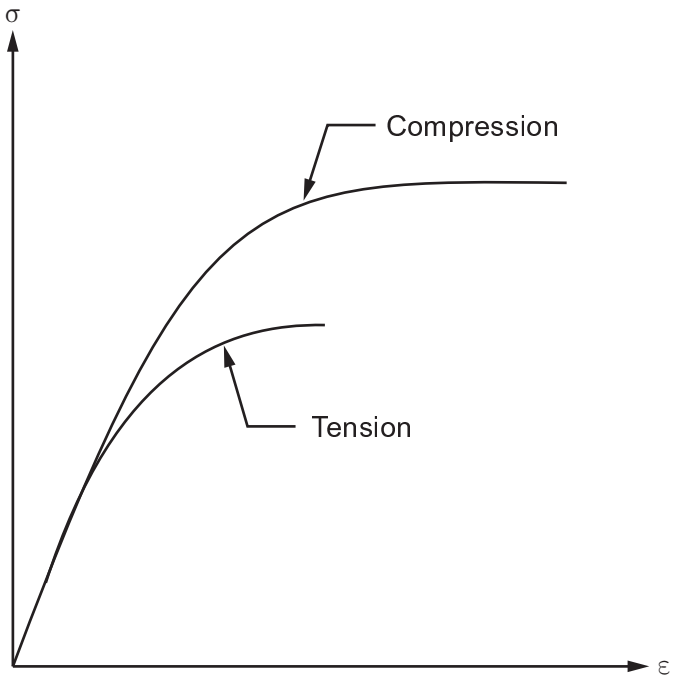
4.2.20. Cast Iron Material Model

The cast iron plasticity model is designed to model gray cast iron. The microstructure of gray cast iron can be looked at as a two-phase material, graphite flakes inserted into a steel matrix (Hjelm([334] (p. 939))). This microstructure leads to a substantial difference in behavior in tension and compression. In tension, the material is more brittle with low strength and cracks form due to the graphite flakes. In compression, no cracks form, the graphite flakes behave as incompressible media that transmit stress and the steel matrix only governs the overall behavior.

The model assumes isotropic elastic behavior, and the elastic behavior is assumed to be the same in tension and compression. The plastic yielding and hardening in tension may be different from that in compression (see Figure 4.14: Idealized Response of Gray Cast Iron in Tension and Compression (p. 102)).

The plastic behavior is assumed to harden isotropically and that restricts the model to monotonic loading only.

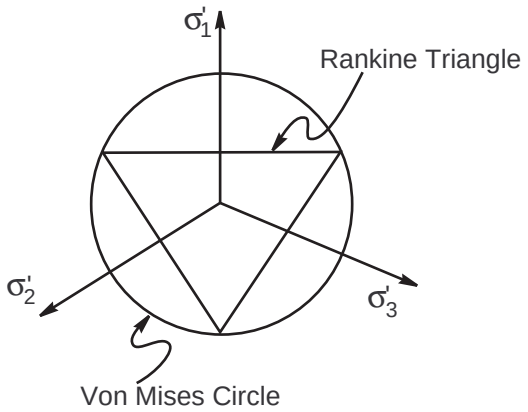
Figure 4.14: Idealized Response of Gray Cast Iron in Tension and Compression



Yield Criteria

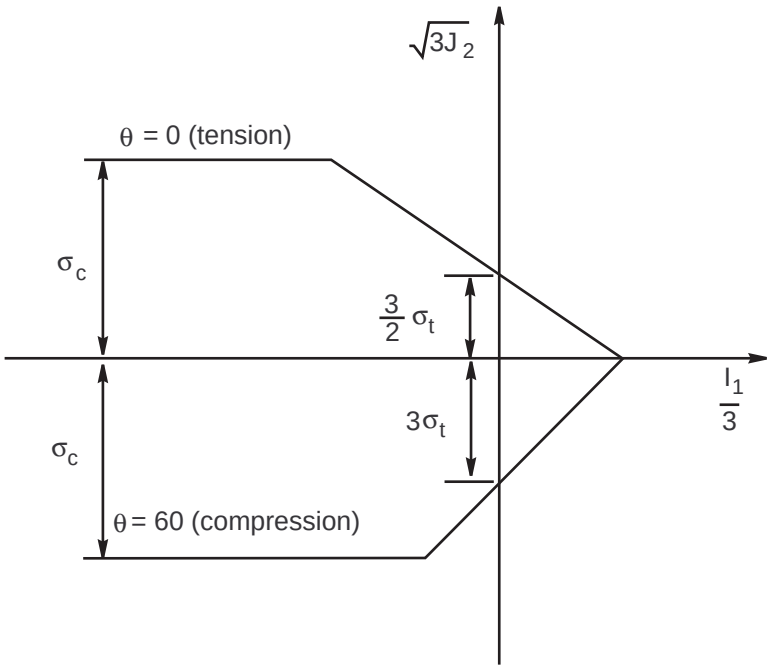
A composite yield surface is used to describe the different behavior in tension and compression. The tension behavior is pressure dependent and the Rankine maximum stress criterion is used. The compression behavior is pressure independent and the von Mises yield criterion is used. The yield surface is a cylinder with a tension cutoff (cap). Figure 4.15: Cross-Section of Yield Surface (p. 102) shows a cross section of the yield surface on principal deviatoric-stress space and Figure 4.16: Meridian Section of Yield Surface (p. 103) shows a meridional sections of the yield surface for two different stress states, compression ($\theta = 60$) and tension ($\theta = 0$).

Figure 4.15: Cross-Section of Yield Surface



(Viewed along the hydrostatic pressure axis)

Figure 4.16: Meridian Section of Yield Surface



(von Mises cylinder with tension cutoff)

The yield surface for tension and compression "regimes" are described by Equation 4.128 (p. 103) and Equation 4.129 (p. 103) (Chen and Han([332] (p. 939))).

The yield function for the tension cap is:

$$f_t = \frac{2}{3}\cos(\theta)\sigma_e + p - \sigma_t = 0 \tag{4.128}$$

and the yield function for the compression regime is:

$$f_c = \sigma_e - \sigma_c = 0 \tag{4.129}$$

where:

$p = I_1 / 3 = tr(\sigma) / 3$ = hydrostatic pressure

$\sigma_e = \left(\frac{3}{2}S:S\right)^{\frac{1}{2}}$ = von Mises equivalent stress

S = deviatoric stress tensor

$\theta = \frac{1}{3}\arccos\left(\frac{3\sqrt{3}J_3}{2J_2^{3/2}}\right)$ = Lode angle

$J_2 = \frac{1}{2}S:S$ = second invariant of deviatoric stress tensor

$J_3 = \det(S)$ = third invariant of deviatoric stress tensor

σ_t = tension yield stress

σ_c = compression yield stress

Flow Rule

The plastic strain increments are defined as:

$$\dot{\varepsilon}^{pl} = \dot{\lambda}\frac{\partial Q}{\partial \sigma} \tag{4.130}$$

where Q is the so-called plastic flow potential, which consists of the von Mises cylinder in compression and modified to account for the plastic Poisson's ratio in tension, and takes the form:

$$Q = \sigma_e - \sigma_c \qquad \text{for } p < -\bar{\sigma}_c / 3 \tag{4.131}$$

$$\frac{(p-Q)^2}{c^2} + \sigma_e^2 = 9Q^2 \qquad \text{for } p \geq -\bar{\sigma}_c / 3 \tag{4.132}$$

and

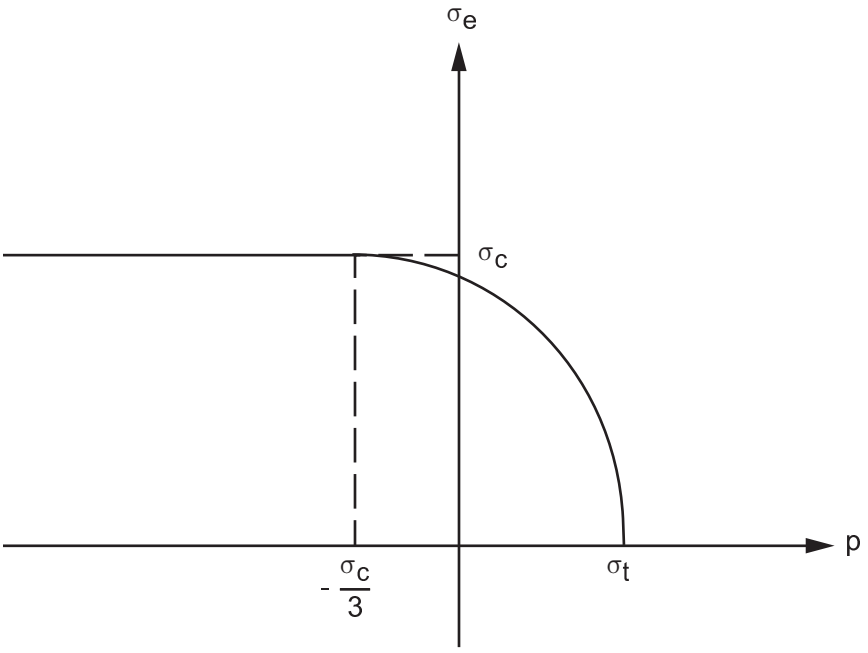
where:

$$c = \sqrt{\frac{9(1-2\nu)^{pl}}{5+2\nu^{pl}}}$$

ν^{pl} = plastic Poisson's ratio (input using **TB**,CAST)

Equation 4.132 (p. 104) is for less than 0.5. When ν^{pl} = 0.5, the equation reduces to the von Mises cylinder. This is shown below:

Figure 4.17: Flow Potential for Cast Iron



As the flow potential is different from the yield function, nonassociated flow rule, the resulting material Jacobian is unsymmetric.

Hardening

The yield stress in uniaxial tension, σ_t, depends on the equivalent uniaxial plastic strain in tension, $\bar{\varepsilon}_t^{pl}$, and the temperature T. Also the yield stress in uniaxial compression, σ_c, depends on the equivalent uniaxial plastic strain in compression, $\bar{\varepsilon}_c^{pl}$, and the temperature T.

To calculate the change in the equivalent plastic strain in tension, the plastic work expression in the uniaxial tension case is equated to general plastic work expression as:

$$\sigma_t \Delta \bar{\varepsilon}_t^{pl} = \{\sigma\}^T \{\Delta \varepsilon^{pl}\} \tag{4.133}$$

where:

$\{\Delta \varepsilon^{pl}\}$ = plastic strain vector increment

Equation 4.130 (p. 104) leads to:

$$\Delta \bar{\varepsilon}_t^{pl} = \frac{1}{\sigma_t} \{\sigma\}^T \{\Delta \varepsilon^{pl}\} \tag{4.134}$$

In contrast, the change in the equivalent plastic strain in compression is defined as:

$$\Delta \bar{\varepsilon}_c^{p} = \Delta \hat{\varepsilon}^{pl} \tag{4.135}$$

where:

$\Delta \hat{\varepsilon}^{pl}$ = equivalent plastic strain increment

The yield and hardening in tension and compression are provided using the **TB**,UNIAXIAL command which has two options, tension and compression.

4.3. Rate-Dependent Plasticity (Including Creep and Viscoplasticity)

Rate-dependent plasticity describes the flow rule of materials, which depends on time. The deformation of materials is now assumed to develop as a function of the strain rate (or time). An important class of applications of this theory is high temperature creep. Several options are provided in ANSYS to characterize the different types of rate-dependent material behaviors. The creep option is used for describing material creep over a relative long period or at low strain. The rate-dependent plasticity option adopts a unified creep approach to describe material behavior that is strain rate dependent. Anand's viscoplasticity option is another rate-dependent plasticity model for simulations such as metal forming. Other than other these built-in options, a rate-dependent plasticity model may be incorporated as user material option through the user programmable feature.

4.3.1. Creep

The following topics related to creep are available:

4.3.1.1. Definition and Limitations
4.3.1.2. Calculation of Creep
4.3.1.3. Time Step Size

See also Extended Drucker-Prager (EDP) Creep Model (p. 112) and Cap Creep Model (p. 117).

4.3.1.1. Definition and Limitations

Creep is defined as material deforming under load over time in such a way as to tend to relieve the stress. Creep may also be a function of temperature and neutron flux level. The term *relaxation* has also been used interchangeably with creep.

The von Mises or Hill stress potentials can be used for creep analysis. For the von Mises potential, the material is assumed to be isotropic and the basic solution technique used is the initial-stiffness Newton-Raphson method.

The options available for creep are described in Rate-Dependent Viscoplastic Materials in the *Material Reference*.

Four different types of creep are available and the effects of the first three may be added together except as noted:

- *Primary creep* is accessed with C_6 (C_i values refer to the i^{th} value given in the **TBDATA** command with **TB**,CREEP). The creep calculations are bypassed if $C_1 = 0$.
- *Secondary creep* is accessed with C_{12}. These creep calculations are bypassed if $C_7 = 0$. They are also bypassed if a primary creep strain was computed using the option $C_6 = 9$, 10, 11, 13, 14, or 15, since they include secondary creep in their formulations.
- *Irradiation-induced creep* is accessed with C_{66}.
- *User-specified creep* may be accessed with $C_6 = 100$. See User Routines and Non-Standard Uses of the *Advanced Analysis Guide* for more details.

The creep calculations are also bypassed if:

1. (change of time) $\leq$ 10-6
2. (input temperature + T_{off}) $\leq$ 0 where T_{off} = offset temperature (input on **TOFFST** command).
3. For $C_6 = 0$ case: A special effective strain based on ε^e and ε^{cr} is computed. A bypass occurs if it is equal to zero.

4.3.1.2. Calculation of Creep

The creep equations are integrated with an explicit Euler forward algorithm, which is efficient for problems having small amounts of contained creep strains. A modified total strain is computed:

$$\{\varepsilon_n'\} = \{\varepsilon_n\} - \{\varepsilon_n^{pl}\} - \{\varepsilon_n^{th}\} - \{\varepsilon_{n-1}^{cr}\} \tag{4.136}$$

This equation is analogous to Equation 4.18 (p. 72) for plasticity. The superscripts are described with Understanding Theory Reference Notation (p. 2) and subscripts refer to the time point n. An equivalent modified total strain is defined as:

$$\varepsilon_{et} = \frac{1}{\sqrt{2}(1+\nu)}\left[(\varepsilon_x' - \varepsilon_y')^2 + (\varepsilon_y' - \varepsilon_z')^2 + (\varepsilon_z' - \varepsilon_x')^2 + \frac{3}{2}(\gamma_{xy}')^2 + \frac{3}{2}(\gamma_{yz}')^2 + \frac{3}{2}(\gamma_{zx}')^2\right]^{\frac{1}{2}} \quad (4.137)$$

Also an equivalent stress is defined by:

$$\sigma_e = E\,\varepsilon_{et} \quad (4.138)$$

where:

E = Young's modulus (input as EX on **MP** command)
ν = Poisson's ratio (input as PRXY or NUXY on **MP** command)

The equivalent creep strain increment ($\Delta\varepsilon^{cr}$) is computed as a scalar quantity from the relations given in Rate-Dependent Viscoplastic Materials and is normally positive. If C_{11} = 1, a decaying creep rate is used rather than a rate that is constant over the time interval. This option is normally not recommended, as it can seriously underestimate the total creep strain where primary stresses dominate. The modified equivalent creep strain increment $(\Delta\varepsilon_m^{cr})$, which would be used in place of the equivalent creep strain increment ($\Delta\varepsilon^{cr}$) if C_{11} = 1, is computed as:

$$\Delta\varepsilon_m^{cr} = \varepsilon_{et}\left(1 - \frac{1}{e^A}\right) \quad (4.139)$$

where:

e = 2.718281828 (base of natural logarithms)
$A = \Delta\varepsilon^{cr}/\varepsilon_{et}$

Next, the creep ratio (a measure of the increment of creep strain) for this integration point (C_s) is computed as:

$$C_s = \frac{\Delta\varepsilon^{cr}}{\varepsilon_{et}} \quad (4.140)$$

The largest value of C_s for all elements at all integration points for this iteration is called C_{max} and is output with the label "CREEP RATIO".

The creep strain increment is then converted to a full strain tensor. N_c is the number of strain components for a particular type of element. If N_c = 1,

$$\Delta\varepsilon_x^{cr} = \Delta\varepsilon^{cr}\left(\frac{\varepsilon_x'}{\varepsilon_{et}}\right) \quad (4.141)$$

Note that the term in brackets is either +1 or -1. If N_c = 4,

$$\Delta\varepsilon_x^{cr} = \frac{\Delta\varepsilon^{cr}}{\varepsilon_{et}}\frac{(2\varepsilon_x' - \varepsilon_y' - \varepsilon_z')}{2(1+\nu)} \quad (4.142)$$

$$\Delta\varepsilon_y^{cr} = \frac{\Delta\varepsilon^{cr}}{\varepsilon_{et}} \frac{(2\varepsilon_y' - \varepsilon_z' - \varepsilon_x')}{2(1+\nu)} \tag{4.143}$$

$$\Delta\varepsilon_z^{cr} = -\Delta\varepsilon_x^{cr} - \Delta\varepsilon_y^{cr} \tag{4.144}$$

$$\Delta\varepsilon_{xy}^{cr} = \frac{\Delta\varepsilon^{cr}}{\varepsilon_{et}} \frac{3}{2(1+\nu)} \gamma_{xy}' \tag{4.145}$$

The first three components are the three normal strain components, and the fourth component is the shear component. If N_c = 6, components 1 through 4 are the same as above, and the two additional shear components are:

$$\Delta\varepsilon_{yz}^{cr} = \frac{\Delta\varepsilon^{cr}}{\varepsilon_{et}} \frac{3}{2(1+\nu)} \gamma_{yz}' \tag{4.146}$$

$$\Delta\varepsilon_{xz}^{cr} = \frac{\Delta\varepsilon^{cr}}{\varepsilon_{et}} \frac{3}{2(1+\nu)} \gamma_{xz}' \tag{4.147}$$

Next, the elastic strains and the total creep strains are calculated as follows, using the example of the x-component:

$$(\varepsilon_x^{el})_n = (\varepsilon_x')_n - \Delta\varepsilon_x^{cr} \tag{4.148}$$

$$(\varepsilon_x^{cr})_n = (\varepsilon_x^{cr})_{n-1} + \Delta\varepsilon_x^{cr} \tag{4.149}$$

Stresses are based on $(\varepsilon_x')_n$. This gives the correct stresses for imposed force problems and the maximum stresses during the time step for imposed displacement problems.

4.3.1.3. Time Step Size

A stability limit is placed on the time step size (Zienkiewicz and Cormeau([154] (p. 929))). This is because an explicit integration procedure is used in which the stresses and strains are referred to time t_{n-1} (however, the temperature is at time t_n). The creep strain rate is calculated using time t_n. It is recommended to use a time step such that the creep ratio C_{max} is less than 0.10. If the creep ratio exceeds 0.25, the run terminates with the message: "CREEP RATIO OF . . . EXCEEDS STABILITY LIMIT OF .25." Automatic Time Stepping (p. 685) discusses the automatic time stepping algorithm which may be used with creep in order to increase or decrease the time step as needed for an accurate yet efficient solution.

4.3.2. Rate-Dependent Plasticity

The rate-dependent plasticity model includes four options: Perzyna([296] (p. 937)), Peirce et al.([297] (p. 937)), EVH([244] (p. 934)), and Anand([159] (p. 929)). The options are defined via the **TB**,RATE command's *TBOPT* field (where *TBOPT* = PERZYNA, PEIRCE, EVH, and ANAND, respectively), and are available with most current-technology elements.

The material hardening behavior is assumed to be isotropic. The integration of the material constitutive equations are based a return mapping procedure (Simo and Hughes([252] (p. 935))) to enforce both stress and material tangential stiffness matrix are consistent at the end of time step. A typical application of this material model is the simulation of material deformation at high strain rate, such as impact.

The following topics describe each rate-dependent plasticity option:

4.3.2.1. Perzyna Option

The Perzyna option has the following form:

$$\dot{\hat{\varepsilon}}_{pl} = \gamma\left(\frac{\sigma}{\sigma_o} - 1\right)^{1/m} \tag{4.150}$$

where:

$\dot{\hat{\varepsilon}}_{pl}$ = equivalent plastic strain rate
m = strain rate hardening parameter (input as C1 via **TBDATA** command)
γ = material viscosity parameter (input as C2 via **TBDATA** command)
σ = equivalent stress
σ_o = static yield stress of material (defined using **TB**,BISO; **TB**,MISO; or **TB**,NLISO commands)

Note

σ_o is a function of some hardening parameters in general.

As γ tends to ∞, or m tends to zero or $\dot{\hat{\varepsilon}}_{pl}$ tends to zero, the solution converges to the static (rate-independent) solution. However, for this material option when m is very small (< 0.1), the solution shows difficulties in convergence (Peric and Owen([298] (p. 937))).

4.3.2.2. Peirce Option

The Peirce option has the following form:

$$\dot{\hat{\varepsilon}}_{pl} = \gamma\left[\left(\frac{\sigma}{\sigma_o}\right)^{1/m} - 1\right] \tag{4.151}$$

Similar to the Perzyna model, the solution converges to the static (rate-independent) solution, as γ tends to ∞, or m tends to zero, or $\dot{\hat{\varepsilon}}_{pl}$ tends to zero. For small value of m, this option shows much better convergency than PERZYNA option (Peric and Owen([298] (p. 937))).

4.3.2.3. Exponential Visco-Hardening (EVH) Option

For information about this option, see Exponential Visco-Hardening (EVH) Option in the *Material Reference*.

4.3.2.4. Anand Viscoplasticity Option

Metal under elevated temperature, such as the hot-metal-working problems, the material physical behaviors become very sensitive to strain rate, temperature, history of strain rate and temperature, and

strain hardening and softening. The systematical effect of all these complex factors can be taken into account and modeled via Anand viscoplasticity([159] (p. 929), [147] (p. 929)). The Anand option is categorized into the group of the unified plasticity models where the inelastic deformation refers to all irreversible deformation that can not be simply or specifically decomposed into the plastic deformation derived from the rate-independent plasticity theories and the part resulted from the creep effect. Compare to the traditional creep approach, the Anand option introduces a single scalar internal variable "s", called the *deformation resistance*, which is used to represent the isotropic resistance to inelastic flow of the material.

Although the Anand option was originally developed for the metal forming application ([159] (p. 929), [147] (p. 929)), it is however applicable for general applications involving strain and temperature effect, including but not limited to such as solder join analysis, high temperature creep etc.

The inelastic strain rate is described by the flow equation as follows:

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\bar{\varepsilon}}^{pl}\left(\frac{3}{2}\frac{\mathbf{S}}{q}\right) \tag{4.152}$$

where:

$\dot{\boldsymbol{\varepsilon}}^{pl}$ = inelastic strain rate tensor

$\dot{\bar{\varepsilon}}^{pl}$ = rate of accumulated equivalent plastic strain

S, the deviator of the Cauchy stress tensor, is:

$$\mathbf{S} = \boldsymbol{\sigma} - p\mathbf{I} \text{ and } p = \frac{1}{3}\text{tr}(\boldsymbol{\sigma}) \tag{4.153}$$

and q, equivalent stress, is:

$$q = (\frac{3}{2}\mathbf{S} : \mathbf{S})^{\frac{1}{2}} \tag{4.154}$$

where:

p = one-third of the trace of the Cauchy stress tensor
σ = Cauchy stress tensor
I = second order identity tensor
":" = inner product of two second-order tensors

The rate of accumulated equivalent plastic strain, $\dot{\bar{\varepsilon}}^{pl}$, is defined as follows:

$$\dot{\bar{\varepsilon}}^{pl} = (\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \dot{\boldsymbol{\varepsilon}}^{pl})^{\frac{1}{2}} \tag{4.155}$$

The equivalent plastic strain rate is associated with equivalent stress, q, and deformation resistance, s, by:

$$\dot{\bar{\varepsilon}}^{pl} = Ae^{(-\frac{Q}{R\theta})}\{\sinh(\xi\frac{q}{s})\}^{\frac{1}{m}} \tag{4.156}$$

A = constant with the same unit as the strain rate

Q = activation energy with unit of energy/volume
R = universal gas constant with unit of energy/volume/temperature
θ = absolute temperature
ξ = dimensionless scalar constant
s = internal state variable
m = dimensionless constant

Equation 4.156 (p. 110) implies that the inelastic strain occurs at any level of stress (more precisely, deviation stress). This theory is different from other plastic theories with yielding functions where the plastic strain develops only at a certain stress level above yielding stress.

The evolution of the deformation resistance is dependent of the rate of the equivalent plastic strain and the current deformation resistance. It is:

$$\dot{s} = \oplus h_0 \left| 1 - \frac{s}{s^*} \right|^a \dot{\varepsilon}^{pl} \tag{4.157}$$

where:

a = dimensionless constant
h_0 = constant with stress unit
s* = deformation resistance saturation with stress unit

The sign, $\oplus$, is determined by:

$$\oplus = \begin{cases} +1 & \text{if } s \le s^* \\ -1 & \text{if } s > s^* \end{cases} \tag{4.158}$$

The deformation resistance saturation s* is controlled by the equivalent plastic strain rate as follows:

$$s^* = \hat{s} \left\{ \frac{\dot{\varepsilon}^{pl}}{A} e^{\frac{Q}{R\theta}} \right\}^n \tag{4.159}$$

where:

$\hat{s}$ = constant with stress unit
n = dimensionless constant

Because of the $\oplus$, Equation 4.157 (p. 111) is able to account for both strain hardening and strain softening. The strain softening refers to the reduction on the deformation resistance. The strain softening process occurs when the strain rate decreases or the temperature increases. Such changes cause a great reduction on the saturation s* so that the current value of the deformation resistance s may exceed the saturation.

The material constants and their units specified for the Anand option are listed in Table 4.3: Material Parameter Units for the Anand Option (p. 112). All constants must be positive, except constant "a", which must be 1.0 or greater. The inelastic strain rate in Anand's definition of material is temperature and stress dependent as well as dependent on the rate of loading. Determination of the material parameters

is performed by curve-fitting a series of the stress-strain data at various temperatures and strain rates as in Anand([159] (p. 929)) or Brown et al.([147] (p. 929)).

Table 4.3: Material Parameter Units for the Anand Option

TBDATA Constant	Parameter	Meaning	Units
1	s_o	Initial value of deformation resistance	stress, e.g. psi, MPa
2	Q/R	Q = activation energy	energy / volume, e.g. kJ / mole
		R = universal gas constant	energy / (volume temperature), e.g. kJ / (mole - K
3	A	pre-exponential factor	1 / time e.g. 1 / second
4	ξ	multiplier of stress	dimensionless
5	m	strain rate sensitivity of stress	dimensionless
6	h_o	hardening/softening constant	stress e.g. psi, MPa
7	$\hat{S}$	coefficient for deformation resistance saturation value	stress e.g. psi, MPa
8	n	strain rate sensitivity of saturation (deformation resistance) value	dimensionless
9	a	strain rate sensitivity of hardening or softening	dimensionless

where:

kJ = kilojoules
K = kelvin

If h_0 is set to zero, the deformation resistance goes away and the Anand option reduces to the traditional creep model.

4.3.3. Extended Drucker-Prager (EDP) Creep Model

Long term loadings such as gravity and other dead loadings greatly contribute inelastic responses of geomaterials. In such cases the inelastic deformation is resulted not only from material yielding but also from material creep. The part of plastic deformation is rate-independent and the creep part is time or rate-dependent.

In the cases of loading at a low level (not large enough to cause the material to yield), the inelastic deformation may still occur because of the creep effect. To account for the creep effect, the material model extended Drucker-Prager (EDP model) combines rate-independent EDP (except for the cap model) with implicit creep functions. The combination has been done in such a way that yield functions and flow rules defined for rate-independent plasticity are fully exploited for creep deformation, advantageous for complex models as the required data input is minimal.

4.3.3.1. EDP Inelastic Strain Rate Decomposition

We first assume that the material point yields so that both plastic deformation and creep deformation occur. Figure 4.18: Material Point in Yielding Condition Elastically Predicted (p. 113) illustrates such a stress state. We next decompose the inelastic strain rate as follows:

$$\dot{\varepsilon}^{in} = \dot{\varepsilon}^{pl} + \dot{\varepsilon}^{cr} \tag{4.160}$$

where:

$\dot{\varepsilon}^{in}$ = inelastic strain rate tensor

$\dot{\varepsilon}^{pl}$ = plastic strain rate tensor

$\dot{\varepsilon}^{cr}$ = creep strain rate tensor

The plastic strain rate is further defined as follows:

$$\dot{\varepsilon}^{pl} = \dot{\lambda}^{pl} \frac{\partial Q}{\partial \sigma} \tag{4.161}$$

where:

$\dot{\lambda}^{pl}$ = plastic multiplier

Q = flow function that has been previously defined in Equation 4.104 (p. 90), Equation 4.105 (p. 90), and Equation 4.106 (p. 90) in Extended Drucker-Prager Model (p. 89)

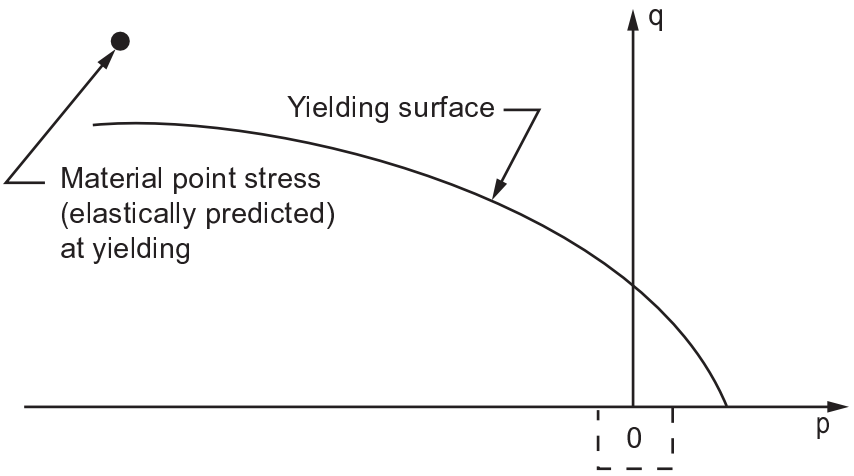
Here we also apply these plastic flow functions to the creep strain rate as follows:

$$\dot{\varepsilon}^{cr} = \dot{\lambda}^{cr} \frac{\partial Q}{\partial \sigma} \tag{4.162}$$

where:

$\dot{\lambda}^{cr}$ = creep multiplier

Figure 4.18: Material Point in Yielding Condition Elastically Predicted



4.3.3.2. EDP Yielding and Hardening Conditions

As material yields, the real stress should always be on the yielding surface. This implies:

$$F(\sigma,\sigma_Y) = F(p,q,\sigma_Y) = 0 \tag{4.163}$$

where:

F = yielding function defined in Equation 4.104 (p. 90), Equation 4.105 (p. 90), and Equation 4.106 (p. 90) in Extended Drucker-Prager Model (p. 89)
σ_Y = yielding stress

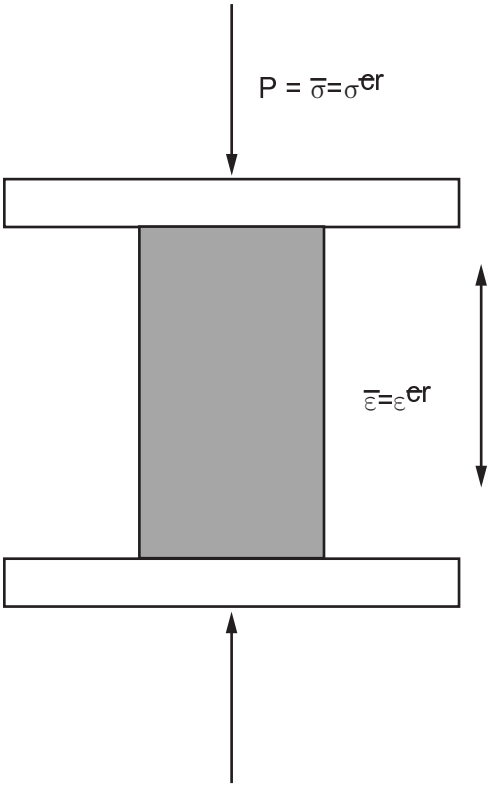
Here we strictly assume that the material hardening is only related to material yielding and not related to material creep. This implies that material yielding stress σ_Y is only the function of the equivalent plastic strain ($\bar{\varepsilon}^{pl}$) as previously defined in the rate-independent extended Drucker-Prager model. We still write it out below for completeness:

$$\sigma_Y = \sigma_Y(\bar{\varepsilon}^{pl}) \tag{4.164}$$

4.3.3.3. EDP Creep Measurements

The creep behaviors could be measured through a few simple tests such as the uniaxial compression, uniaxial tension, and shear tests. We here assume that the creep is measured through the uniaxial compression test described in Figure 4.19: Uniaxial Compression Test (p. 114).

Figure 4.19: Uniaxial Compression Test



The measurements in the test are the vertical stress $\bar{\sigma}$ and vertical creep strain $\bar{\varepsilon}$ at temperature T. The creep test is targeted to be able to describe material creep behaviors in a general implicit rate format as follows:

$$\dot{\bar{\varepsilon}} = h^{cr}(\bar{\varepsilon}, \bar{\sigma}, T, t) \tag{4.165}$$

We define the equivalent creep strain and the equivalent creep stress through the equal creep work as follows:

$$\dot{\bar{\varepsilon}}^{cr}\bar{\sigma}^{cr} = \dot{\varepsilon}^{cr} : \sigma \tag{4.166}$$

where:

$\bar{\varepsilon}^{cr}$ and $\bar{\sigma}^{cr}$ = equivalent creep strain and equivalent creep stress to be defined.

For this particular uniaxial compression test, the stress and creep strain are:

$$\sigma = \begin{bmatrix} -\bar{\sigma} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \varepsilon^{cr} = \begin{bmatrix} -\bar{\varepsilon} & 0 & 0 \\ 0 & \varepsilon_y^{cr} & 0 \\ 0 & 0 & \varepsilon_z^{cr} \end{bmatrix} \tag{4.167}$$

Inserting (Equation 4.167 (p. 115)) into (Equation 4.166 (p. 115)) , we conclude that for this special test case the equivalent creep strain and the equivalent creep stress just recover the corresponding test measurements. Therefore, we are able to simply replace the two test measurements in (Equation 4.165 (p. 115)) with two variables of the equivalent creep strain and the equivalent creep stress as follows:

$$\dot{\varepsilon}^{cr} = h^{cr}(\bar{\varepsilon}^{cr}, \bar{\sigma}^{cr}, T, t) \tag{4.168}$$

Once the equivalent creep stress for any arbitrary stress state is obtained, we can insert it into (Equation 4.168 (p. 115)) to compute the material creep rate at this stress state. We next focus on the derivation of the equivalent creep stress for any arbitrary stress state.

4.3.3.4. EDP Equivalent Creep Stress

We first introduce the creep isosurface concept. Figure 4.20: Creep Isosurface (p. 116) shows any two material points A and B at yielding but they are on the same yielding surface. We say that the creep behaviors of point A and point B can be measured by the same equivalent creep stress if any and the yielding surface is called the creep isosurface. We now set point B to a specific point, the intersection between the yielding curve and the straight line indicating the uniaxial compression test. From previous creep measurement discussion, we know that point B has $-\bar{\sigma}^{cr}/3$ for the coordinate p and $\bar{\sigma}^{cr}$ for the coordinate q. Point B is now also on the yielding surface, which immediately implies:

$$F(-\bar{\sigma}^{cr}/3, \bar{\sigma}^{cr}, \sigma_Y) = 0 \tag{4.169}$$

It is interpreted from (Equation 4.169 (p. 115)) that the yielding stress σ_Y is the function of the equivalent creep stress $\bar{\sigma}^{cr}$. Therefore, we have:

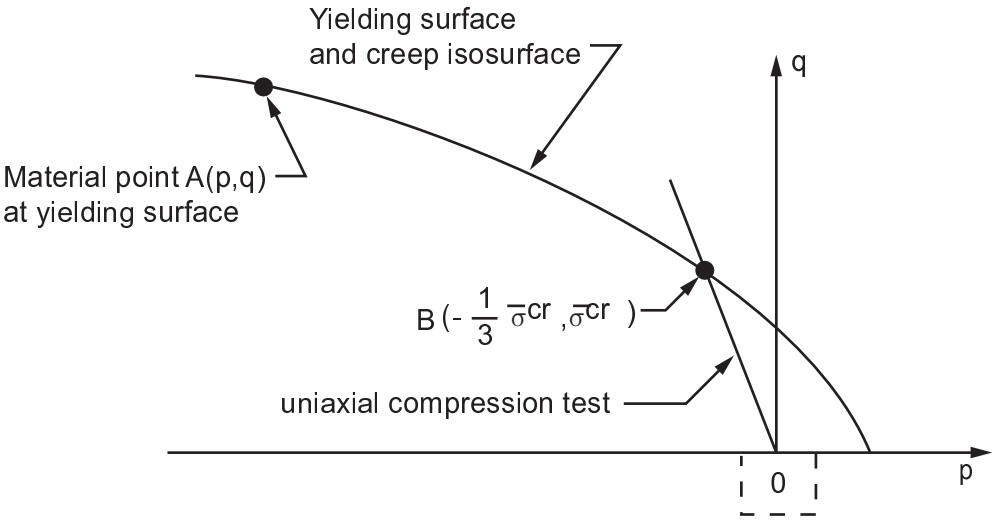
$$\sigma_Y = \sigma_Y(\bar{\sigma}^{cr}) \tag{4.170}$$

We now insert (Equation 4.170 (p. 115)) into the yielding condition (Equation 4.163 (p. 114)) again:

$$F(p,q,\sigma_Y(\bar{\sigma}^{cr}))=0 \tag{4.171}$$

We then solve (Equation 4.171 (p. 116)) for the equivalent creep stress $\bar{\sigma}^{cr}$ for material point A on the isosurface but with any arbitrary coordinates (p,q). (Equation 4.171 (p. 116)) is, in general, a nonlinear equation and the iteration procedure must be followed for searching its root. In the local material iterations, for a material stress point not on the yielding surface but out of the yielding surface like the one shown in Figure 4.18: Material Point in Yielding Condition Elastically Predicted (p. 113), (Equation 4.171 (p. 116)) is also valid and the equivalent creep stress solved is always positive.

Figure 4.20: Creep Isosurface



4.3.3.5. EDP Elastic Creep and Stress Projection

When the loading is at a low level or the unloading occurs, the material doesn't yield and is at an elastic state from the point view of plasticity. However, the inelastic deformation may still exist fully due to material creep. In this situation, the equivalent creep stress obtained from (Equation 4.171 (p. 116)) may be negative in some area. If this is the case, (Equation 4.171 (p. 116)) is not valid any more. To solve this difficulty, the stress-projection method shown in Figure 4.21: Stress Projection (p. 117) is used, where the real stress σ is multiplied by an unknown scalar β so that the projected stress $\sigma^* = \beta\sigma$ is on the yielding surface. The parameter β can be obtained by solving the following equation:

$$F(\sigma^*,\sigma_Y)=F(\beta\sigma,\sigma_Y)=0 \tag{4.172}$$

Again, Equation 4.172 (p. 116) is a nonlinear equation except in the linear Drucker-Prager model. Because the projected stress σ^* is on the yielding surface, the equivalent creep stress denoted as $\bar{\sigma}^{cr*}$ and calculated by inserting σ^* into (Equation 4.171 (p. 116)), as follows:

$$F(\sigma^*,\sigma_Y(\bar{\sigma}^{cr*}))=0 \tag{4.173}$$

is always positive. The real equivalent creep stress $\bar{\sigma}^{cr}$ is obtained by simply rescaling $\bar{\sigma}^{cr*}$ as follows:

$$\bar{\sigma}^{cr}=\bar{\sigma}^{cr*}/\beta \tag{4.174}$$

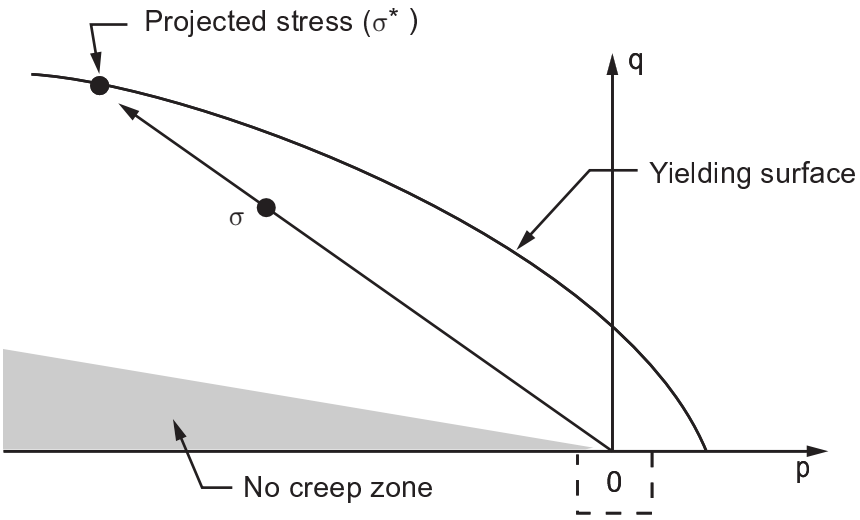
For creep flow in this situation, (Equation 4.162 (p. 113)) can be simply modified as follows:

$$\dot{\varepsilon}^{cr} = \dot{\lambda}^{cr}\beta\frac{\partial Q}{\partial \sigma^*} \tag{4.175}$$

For stress in a particular continuous domain indicated by the shaded area in Figure 4.21: Stress Projection (p. 117), the stresses are not able to be projected on the yielding surface (that is, Equation 4.172 (p. 116)) has no positive value of solution for β). For stresses in this area, no creep is assumed. This assumption makes some sense partially because this area is pressure-dominated and the EDP models are shear-dominated.

Having Equation 4.160 (p. 113),Equation 4.161 (p. 113), Equation 4.162 (p. 113), or Equation 4.175 (p. 117), Equation 4.163 (p. 114), Equation 4.164 (p. 114), Equation 4.166 (p. 115), and Equation 4.168 (p. 115), the EDP creep model is a mathematically well-posed problem.

Figure 4.21: Stress Projection



4.3.4. Cap Creep Model

The cap creep model is an extension of the cap (rate-independent plasticity) model. The extension is based on creep theory similar to that of the Extended Drucker-Prager (EDP) creep model.

Several concepts related to material creep theory (such as the equivalent creep stress, equivalent creep strain, and isosurfaces) are used in the cap creep model and are described in the EDP creep theory.

Unlike EDP which requires only one creep test measurement, a cap creep model requires *two* independent creep test measurements (as described in Commands Used for Cap Creep (p. 121)) to account for both shear-dominated creep and compaction-dominated creep behaviors.

The following topics describing the cap creep model are available:

4.3.4.1. Assumptions and Restrictions
4.3.4.2. Functions and Potentials
4.3.4.3. Cap Zones
4.3.4.4. Equivalent Creep Stress
4.3.4.5. Elastic Creep and Stress Projection
4.3.4.6. Commands Used for Cap Creep

For related information, see Extended Drucker-Prager Cap Model (p. 90) and Extended Drucker-Prager (EDP) Creep Model (p. 112) in this reference.

4.3.4.1. Assumptions and Restrictions

The following assumptions and restrictions apply to the cap creep model:

- Creep in both the shear and creep-expansion zones is measured via uniaxial testing.
- The compaction creep zone is measured via hydrostatic testing.
- Shear creep and compaction creep are independent of each other.
- Shear creep exists in all zones insofar as possible.
- The Lode angle effect is ignored.
- Material hardening is dependent on plastic strain only.

4.3.4.2. Functions and Potentials

The following yielding equation applies of the cap portion of the cap creep model:

$$\begin{aligned} Y(\sigma, K_0, \sigma_0) &= Y(I_1, J_2, K_0, \sigma_0) \\ &= J_2 - F_t(I_1, \sigma_0) Y_s^2(0, \sigma_0) \end{aligned} \tag{4.176}$$

This equation represents the flow function of the cap creep model:

$$\begin{aligned} F(\sigma, K_0, \sigma_0) &= F(I_1, J_2, K_0, \sigma_0) \\ &= J_2 - F_t(I_1, \sigma_0) F_s^2(0, \sigma_0) \end{aligned} \tag{4.177}$$

The equations are similar to those used for yielding and flow function (respectively) in the cap (rate-independent plasticity) model.

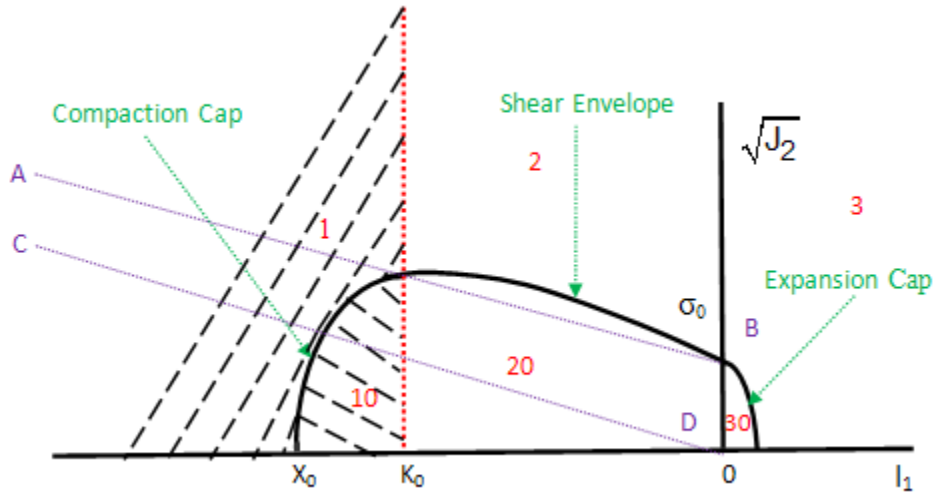
The tension cap portion for the cap creep model is nearly identical to the cap (rate-independent plasticity) yielding function except that the J_3 (Lode angle) effect is ignored. The notations and conventions remain the same.

Equation 4.176 (p. 118) is also used to calculate the equivalent creep stresses, and Equation 4.177 (p. 118) is used to calculate the creep strain.

4.3.4.3. Cap Zones

For the sake of convenience, the cap domain shown in the following figure is divided into six zones (labeled 1, 2, 3, 10, 20, and 30):

Figure 4.22: Cap Zoning



Following is a brief description of each zone:

- **Zone 1** shows the compaction zone where the material yields and creeps. **Zone 10** represents a compaction-dominated zone; its inelastic behavior is still subject to a certain degree of creep due to compaction stress.
- **Zone 2** shows the shear zone where the material yields and creeps due to the shear effect. **Zone 20** represents an elastic shear creep zone.
- **Zone 3** shows the expansion zone with both yield and creep. **Zone 30** represents an elastic creep zone.

Creep in zones 2, 20, 3, and 30 is measured via uniaxial compression testing. Those zones do not experience compaction creep contributed from the compaction cap.

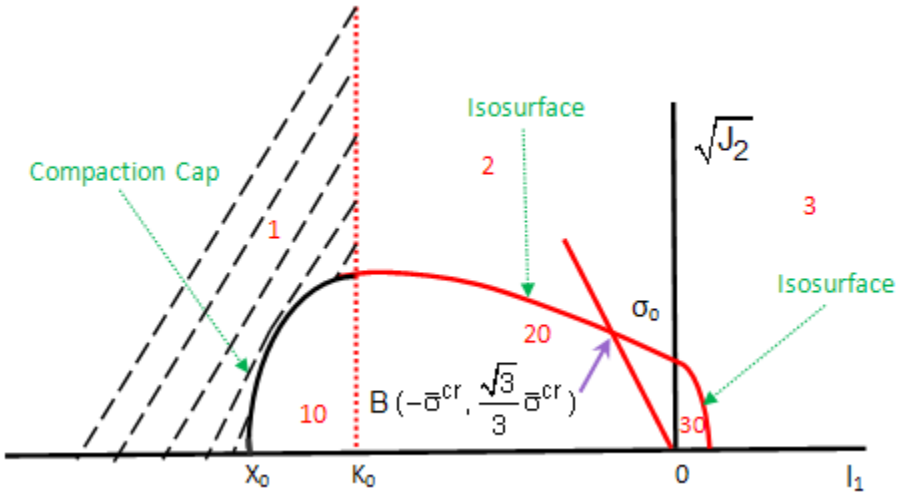
Zones 1 and 10 may also experience shear creep because a certain amount of shear creep exists in *all* zones. It is assumed that stress points below line CD are not subject to shear creep.

4.3.4.4. Equivalent Creep Stress

The shear and expansion zones share the same uniaxial compression for creep measurement. The derivation for the equivalent creep stress for the two zones is similar to that used for extended Drucker-Prager creep stress.

In coordinate system $(I_1, \sqrt{J_2})$ shown in the following figure, the coordinates of the stress points from the uniaxial compression test are $(-\bar{\sigma}^{cr}, \frac{\sqrt{3}}{3}\bar{\sigma}^{cr})$, where $\bar{\sigma}^{cr}$ is the measured stress or the equivalent creep stress.

Figure 4.23: Uniaxial Compression Test and Creep Isosurface for Shear and Expansion Zones



The point also on the shear yielding surface should satisfy the corresponding yielding function, as follows:

$$Y(\sigma,\sigma_0) = Y(I_1, J_2, \sigma_0) = J_2 - Y_s^2(I_1,\sigma_0) \tag{4.178}$$

or

$$Y(-\bar{\sigma}^{cr}, (\frac{\sqrt{3}}{3}\bar{\sigma}^{cr})^2, \sigma_Y) = (\frac{\sqrt{3}}{3}\bar{\sigma}^{cr})^2 - Y_s^2(-\bar{\sigma}^{cr},\sigma_0) = 0 \tag{4.179}$$

Equation 4.179 (p. 120) implies $\sigma_0 = \sigma_0(\bar{\sigma}^{cr})$, which is used in the yielding condition

$$Y(\sigma,\sigma_0) = Y(I_1, J_2, \sigma_0) = J_2 - Y_t(I_1,\sigma_0)Y_s^2(I_1,\sigma_0)$$

resulting in:

$$\begin{aligned} Y(\sigma,\sigma_0(\bar{\sigma}^{cr})) &= Y(I_1, J_2, \sigma_0(\bar{\sigma}^{cr})) \\ &= J_2 - Y_t(I_1,\sigma_0(\bar{\sigma}^{cr}))Y_s^2(I_1,\sigma_0(\bar{\sigma}^{cr})) = 0 \end{aligned} \tag{4.180}$$

Equation 4.180 (p. 120) is then solved for the equivalent creep stress $\bar{\sigma}^{cr}$ for stresses at either the shear zone or the expansion zone. As shown in the EDP creep model, the measured creep strain component in the axial direction is the equivalent creep strain; therefore, the creep rate function for both shear and expansion zones is:

$$\dot{\bar{\varepsilon}}_{st}^{cr} = h_{st}^{cr}(\bar{\varepsilon}_{st}^{cr}, \bar{\sigma}^{cr}, T)$$

where $\bar{\varepsilon}_{st}^{cr}$ is the equivalent creep strain for the shear or expansion zone and T is the temperature.

4.3.4.4.1. Equivalent Creep Stress in the Compaction Zone

The equivalent creep stress for compaction cap is assumed to be independent of J_2 and only related to the first invariant of stress:

$$\bar{\sigma}_c^{cr} = K_0 - I_1$$

The creep test measurement for the compaction cap is regulated via hydrostatic compression testing. The measurement and record are

- $|3p|$ where p is the hydrostatic pressure, and
- $\left|\varepsilon_V^{cr}\right| = \bar{\varepsilon}_V^{cr}$ where ε_V^{cr} is the volume creep strain.

Thus,.the creep rate function for the compaction cap is:

$$\dot{\bar{\varepsilon}}_V{}^{cr} = h_c^{cr}(\bar{\varepsilon}_V{}^{cr}, \bar{\sigma}_c^{cr}, T) \tag{4.181}$$

4.3.4.5. Elastic Creep and Stress Projection

In zones 30, 20, 10, and 1 (where the material does not yield at the shear envelope criterion), loading is light; it also possible that unloading occurs. Inelastic deformation may still exist, however, due exclusively to material shear creep. In such a case, the equivalent creep stress obtained from the previous derivation may be negative. To avoid a negative equivalent creep stress, the stress-projection method proposed in EDP Elastic Creep and Stress Projection (p. 116) is used.

4.3.4.6. Commands Used for Cap Creep

To model cap creep, use the input from the cap (rate-independent plasticity) model. The cap creep model ignores any tension-to-compression-strength ratio values and sets them to 1.

Creep data is input via the **TB**,CREEP command. Because the cap creep model requires *two* independent creep function inputs, an additional material table data command, **TBEO**, allows shear and compaction creep data to be defined separately.

4.4. Gasket Material

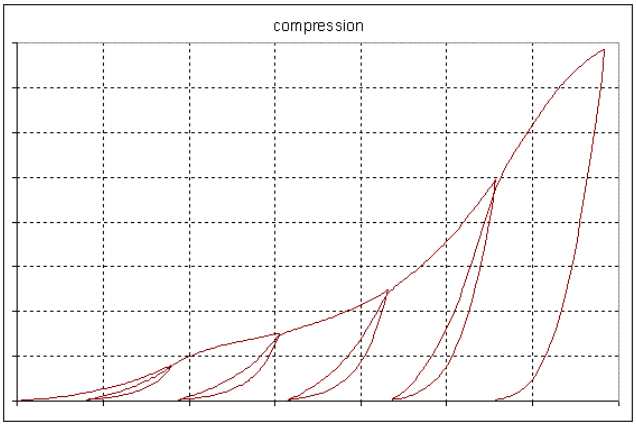
Gasket joints are essential components in most of structural assemblies. Gaskets as sealing components between structural components are usually very thin and made of many materials, such as steel, rubber and composites. From a mechanics point of view, gaskets act to transfer the force between mating components. The gasket material is usually under compression. The material under compression exhibits high nonlinearity. The gasket material also shows quite complicated unloading behavior. The primary deformation of a gasket is usually confined to 1 direction, that is through-thickness. The stiffness contribution from membrane (in-plane) and transverse shear are much smaller, and are neglected.

The table option GASKET allows gasket joints to be simulated with the interface elements, in which the through-thickness deformation is decoupled from the in-plane deformation, see INTER192 - 2-D 4-Node Gasket (p. 604), INTER193 - 2-D 6-Node Gasket (p. 605), INTER194 - 3-D 16-Node Gasket (p. 605), and INTER195 - 3-D 8-Node Gasket (p. 607) for detailed description of interface elements. The user can directly input the experimentally measured complex pressure-closure curve (compression curve) and several unloading pressure-closure curves for characterizing the through thickness deformation of gasket material.

Figure 4.24: Pressure vs. Deflection Behavior of a Gasket Material (p. 122) shows the experimental pressure vs. closure (relative displacement of top and bottom gasket surfaces) data for a graphite composite

gasket material. The sample was unloaded and reloaded 5 times along the loading path and then unloaded at the end of the test to determine the unloading stiffness of the material.

Figure 4.24: Pressure vs. Deflection Behavior of a Gasket Material



4.4.1. Stress and Deformation

The gasket pressure and deformation are based on the local element coordinate systems. The gasket pressure is actually the stress normal to the gasket element midsurface in the gasket layer. Gasket deformation is characterized by the closure of top and bottom surfaces of gasket elements, and is defined as:

$$d = u^{TOP} - u^{BOTTOM} \tag{4.182}$$

Where, u^{TOP} and u^{BOTTOM} are the displacement of top and bottom surfaces of interface elements in the local element coordinate system based on the mid-plane of element.

4.4.2. Material Definition

The input of material data of a gasket material is specified by the command (**TB**,GASKET). The input of material data considers of 2 main parts: general parameters and pressure closure behaviors. The general parameters defines initial gasket gap, the stable stiffness for numerical stabilization, and the stress cap for gasket in tension. The pressure closure behavior includes gasket compression (loading) and tension data (unloading).

The GASKET option has followings sub-options:

Sub-option	Description
PARA	Define gasket material general parameters
COMP	Define gasket compression data
LUNL	Define gasket linear unloading data
NUNL	Define gasket nonlinear unloading data

A gasket material can have several options at the same time. When no unloading curves are defined, the material behavior follows the compression curve while it is unloaded.

4.4.3. Thermal Deformation

The thermal deformation is taken into account by using an additive decomposition in the total deformation, d, as:

$$d = d_i + d_{th} + d_o \tag{4.183}$$

where:

d = relative total deformation between top and bottom surfaces of the interface element
d_i = relative deformation between top and bottom surfaces causing by the applying stress, this can be also defined as mechanical deformation
d_{th} = relative thermal deformation between top and bottom surfaces due to free thermal expansion
d_o = initial gap of the element and is defined by sub-option PARA

The thermal deformation causing by free thermal expansion is defined as:

$$d_{th} = \alpha * \Delta T * h \tag{4.184}$$

where:

α = coefficient of thermal expansion (input as ALPX on **MP** command)
ΔT = temperature change in the current load step
h = thickness of layer at the integration point where thermal deformation is of interest

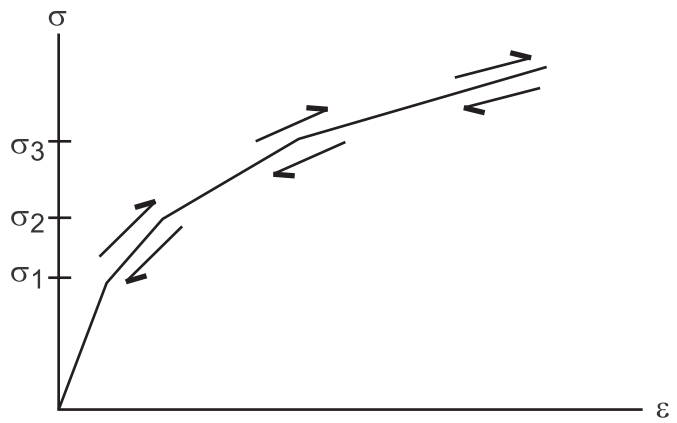
4.5. Nonlinear Elasticity

4.5.1. Overview and Guidelines for Use

The ANSYS program provides a capability to model nonlinear (multilinear) elastic materials (input using **TB**,MELAS). Unlike plasticity, no energy is lost (the process is conservative).

Figure 4.25: Stress-Strain Behavior for Nonlinear Elasticity (p. 123) represents the stress-strain behavior of this option. Note that the material unloads along the same curve, so that no permanent inelastic strains are induced.

Figure 4.25: Stress-Strain Behavior for Nonlinear Elasticity



The total strain components $\{\varepsilon_n\}$ are used to compute an equivalent total strain measure:

$$\varepsilon_e^t = \frac{1}{\sqrt{2}(1+\nu)}\left[(\varepsilon_x-\varepsilon_y)^2+(\varepsilon_y-\varepsilon_z)^2+(\varepsilon_z-\varepsilon_x)^2 + \frac{3}{2}(\varepsilon_{xy})^2+\frac{3}{2}(\varepsilon_{yz})^2+\frac{3}{2}(\varepsilon_{xz})^2\right]^{\frac{1}{2}} \tag{4.185}$$

ε_e^t is used with the input stress-strain curve to get an equivalent value of stress σ_e .

The elastic (linear) component of strain can then be computed:

$$\{\varepsilon_n^{el}\} = \frac{\sigma_e}{E\,\varepsilon_e^t}\{\varepsilon_n\} \tag{4.186}$$

and the "plastic" or nonlinear portion is therefore:

$$\{\varepsilon_n^{pl}\} = \{\varepsilon_n\} - \{\varepsilon_n^{el}\} \tag{4.187}$$

In order to avoid an unsymmetric matrix, only the symmetric portion of the tangent stress-strain matrix is used:

$$[D_{ep}] = \frac{\sigma_e}{E\,\varepsilon_e}[D] \tag{4.188}$$

which is the secant stress-strain matrix.

4.6. Hyperelasticity

Hyperelasticity refers to a constitutive response that is derivable from an elastic free energy potential and is typically used for materials which experience large elastic deformation. Applications for elastomers such as vulcanized rubber and synthetic polymers, along with some biological materials, often fall into this category.

The microstructure of polymer solids consists of chain-like molecules. The flexibility of these molecules allows for an irregular molecular arrangement and, as a result, the behavior is very complex. Polymers are usually isotropic at small deformation and anisotropic at larger deformation as the molecule chains realign to the loading direction. Under an essentially monotonic loading condition, however, many polymer materials can be approximated as isotropic, which has historically been popular in the modeling of polymers.

Some classes of hyperelastic materials cannot be modeled as isotropic. An example is fiber reinforced polymer composites. Typical fiber patterns include unidirectional and bidirectional, and the fibers can have a stiffness that is 50-1000 times that of the polymer matrix, resulting in a strongly anisotropic material behavior. Another class of anisotropic materials that can experience large deformation is bio-materials, such as muscles and arteries, in which the anisotropic behavior is due to their fibrous structure.

The typical volumetric behavior of hyperelastic materials can be grouped into two classes. Materials such as polymers typically have small volumetric changes during deformation and these are incompressible or nearly-incompressible materials. An example of the second class of materials is foams, which can experience large volumetric changes during deformation, and these are compressible materials.

The available hyperelastic material constitutive models are derived from strain-energy potentials that are functions of the deformation invariants. An exception is the response function model which obtains the constitutive response functions directly from experimental data. The hyperelastic material models are defined through data tables (**TB**,HYPER or **TB**,AHYPER) and include:

- **Incompressible or nearly-incompressible isotropic models:** Neo-Hookean, Mooney-Rivlin, Polynomial Form, Ogden Potential, Arruda-Boyce, Gent, Yeoh, and Extended Tube. These models work with following elements following elements: SHELL181, PLANE182, PLANE183, SOLID185, SOLID186 , SOLID187, SOLID272, SOLID273, SOLID285, SOLSH190, SHELL208, SHELL209, SHELL281, PIPE288, PIPE289, and ELBOW290.
- **Compressible isotropic models:** Blatz-Ko and Ogden Compressible Foam options are applicable to compressible foam or foam-type materials. These models work with the following elements: SHELL181, PLANE182, PLANE183, SOLID185, SOLID186 , SOLID187, SOLID272, SOLID273, SOLID285, SOLSH190, SHELL208, SHELL209, SHELL281, PIPE288, PIPE289, and ELBOW290.
- **Incompressible or nearly-incompressible isotropic response function hyperelastic model.** This model (**TB**,EXPE) uses experimental data and works with the following elements: SHELL181, PLANE182, PLANE183, SOLID185, SOLID186 , SOLID187, SOLID272, SOLID273, SOLID285, SOLSH190, SHELL208, SHELL209, SHELL281, PIPE288, PIPE289, and ELBOW290.
- **Invariant-based anisotropic strain-energy potential.** This option (**TB**,AHYPER) works with the following elements: PLANE182 and PLANE183 with plane strain and axisymmetric options, and SOLID185, SOLID186, SOLID187, SOLID272, SOLID273, SOLID285, and SOLSH190.

4.6.1. Finite Strain Elasticity

A material is said to be hyperelastic if there exists an elastic potential function W (or strain-energy density function) which is a scalar function of one of the strain or deformation tensors, whose derivative with respect to a strain component determines the corresponding stress component. This can be expressed by:

$$S_{ij} = \frac{\partial W}{\partial E_{ij}} \equiv 2\frac{\partial W}{\partial C_{ij}} \tag{4.189}$$

where:

S_{ij} = components of the second Piola-Kirchhoff stress tensor
W = strain-energy function per unit undeformed volume
E_{ij} = components of the Lagrangian strain tensor
C_{ij} = components of the right Cauchy-Green deformation tensor

The Lagrangian strain may be expressed as follows:

$$E_{ij} = \frac{1}{2}(C_{ij} - \delta_{ij}) \tag{4.190}$$

where:

δ_{ij} = Kronecker delta ($\delta_{ij} = 1$, i = j; $\delta_{ij} = 0$, $i \neq j$)

The deformation tensor C_{ij} is comprised of the products of the deformation gradients F_{ij}

$$C_{ij} = F_{ki}F_{kj} = \text{component of the Cauchy-Green deformation tensor} \tag{4.191}$$

where:

F_{ij} = components of the deformation gradient tensor
X_i = undeformed position of a point in direction i
$x_i = X_i + u_i$ = deformed position of a point in direction i
u_i = displacement of a point in direction i

The Kirchhoff stress is defined:

$$\tau_{ij} = F_{ik} S_{kl} F_{jl} \tag{4.192}$$

and the Cauchy stress is obtained by:

$$\sigma_{ij} = \frac{1}{J}\tau_{ij} = \frac{1}{J} F_{ik} S_{kl} F_{jl} \tag{4.193}$$

The eigenvalues (principal stretch ratios) of C_{ij} are λ_1^2, λ_2^2, and λ_3^2, and exist only if:

$$\det\left\lfloor C_{ij} - \lambda_p^2 \delta_{ij} \right\rfloor = 0 \tag{4.194}$$

which can be re-expressed as:

$$\lambda_p^6 - I_1 \lambda_p^4 + I_2 \lambda_p^2 - I_3 = 0 \tag{4.195}$$

where:

I_1, I_2, and I_3 = invariants of C_{ij},

$$\begin{aligned} I_1 &= \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \\ I_2 &= \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2 \\ I_3 &= \lambda_1^2\lambda_2^2\lambda_3^2 = J^2 \end{aligned} \tag{4.196}$$

and

$$J = \det\left\lfloor F_{ij} \right\rfloor \tag{4.197}$$

J is also the ratio of the deformed elastic volume over the reference (undeformed) volume of materials (Ogden([295] (p. 937)) and Crisfield([294] (p. 937))).

When there is thermal volume strain, the volume ratio J is replaced by the elastic volume ratio J_{el} which is defined as the total volume ratio J over thermal volume ratio J_{th}, as:

$$J_{el} = J / J_{th} \tag{4.198}$$

and the thermal volume ratio J_{th} is:

$$J_{th} = (1 + \alpha \Delta T)^3 \tag{4.199}$$

where:

α = coefficient of the thermal expansion
ΔT = temperature difference about the reference temperature

4.6.2. Deviatoric-Volumetric Multiplicative Split

Under the assumption that material response is isotropic, it is convenient to express the strain-energy function in terms of strain invariants or principal stretches (Simo and Hughes([252] (p. 935))).

$$W = W(I_1, I_2, I_3) = W(I_1, I_2, J) \tag{4.200}$$

or

$$W = W(\lambda_1, \lambda_2, \lambda_3) \tag{4.201}$$

Define the volume-preserving part of the deformation gradient, $\bar{F}_{ij}$, as:

$$\bar{F}_{ij} = J^{-1/3} F_{ij} \tag{4.202}$$

and thus

$$\bar{J} = \det\left[\bar{F}_{ij}\right] = 1 \tag{4.203}$$

The modified principal stretch ratios and invariants are then:

$$\bar{\lambda}_p = J^{-1/3}\lambda_p \qquad (p = 1,2,3) \tag{4.204}$$

$$\bar{I}_p = J^{-2p/3} I_p \tag{4.205}$$

The strain-energy potential can then be defined as:

$$W = W(\bar{I}_1, \bar{I}_2, J) = W(\bar{\lambda}_1, \bar{\lambda}_2, \bar{\lambda}_3, J) \tag{4.206}$$

4.6.3. Isotropic Hyperelasticity

Following are several forms of strain-energy potential (W) provided (as options *TB0PT* in **TB**,HYPER) for the simulation of incompressible or nearly incompressible hyperelastic materials.

4.6.3.1. Arruda-Boyce Model

The form of the strain-energy potential for Arruda-Boyce model is:

$$\begin{aligned} W = \mu\Bigg[&\frac{1}{2}(\bar{I}_1 - 3) + \frac{1}{20\lambda_L^2}(\bar{I}_1^2 - 9) + \frac{11}{1050\lambda_L^4}(\bar{I}_1^3 - 27) \\ &+ \frac{19}{7050\lambda_L^6}(\bar{I}_1^4 - 81) + \frac{519}{673750\lambda_L^8}(\bar{I}_1^5 - 243)\Bigg] + \frac{1}{d}\left(\frac{J^2 - 1}{2} - \ln J\right) \end{aligned} \tag{4.207}$$

where:

- μ = initial shear modulus of material (input on **TBDATA** commands with **TB**,HYPER)
- λ_L = limiting network stretch (input on **TBDATA** commands with **TB**,HYPER)
- d = material incompressibility parameter (input on **TBDATA** commands with **TB**,HYPER)

The initial bulk modulus is:

$$K = \frac{2}{d} \tag{4.208}$$

As the parameter λ_L goes to infinity, the model is converted to Neo-Hookean form.

4.6.3.2. Blatz-Ko Model

The form of strain-energy potential for the Blatz-Ko model is:

$$W = \frac{\mu}{2}\left(\frac{I_2}{I_3} + 2\sqrt{I_3} - 5\right) \tag{4.209}$$

where:

μ = initial shear modulus of material (input on **TBDATA** commands with **TB**,HYPER)

The initial bulk modulus is defined as:

$$k = \frac{5}{3}\mu \tag{4.210}$$

4.6.3.3. Extended Tube Model

The extended-tube model is a physics-based polymer model which introduces the physical consideration on the molecular scale into the formulation of the strain-energy potential. The model considers the topological constraints as well as the limited chain extensibility of network chains in the filled rubbers.

The elastic strain-energy potential consists of the elastic energy from the crosslinked network and the constraint network as well as the volumetric strain energy:

$$W = \frac{G_c}{2}\left[\frac{\left(1-\delta^2\right)\left(\bar{I}_1 - 3\right)}{1-\delta^2\left(\bar{I}_1 - 3\right)} + \ln\left(1-\delta^2\left(\bar{I}_1 - 3\right)\right)\right] + \frac{2G_e}{\beta^2}\sum_{i=1}^{3}\left(\bar{\lambda}_i^{-\beta} - 1\right) + \frac{1}{d}(J-1)^2 \tag{4.211}$$

where the initial shear modulus is given by $G = G_c + G_e$, and:

G_e = constraint contribution to modulus
G_c = crosslinked contribution to modulus
δ = extensibility parameter
β = empirical parameter ($0 \le \beta \le 1$)
d_1 = material incompressibility parameter

The model is equivalent to a two-term Ogden model with the following parameters:

$$\alpha_1 = 2,\ \alpha_2 = -\beta,$$
$$\mu_1 = G_c,\ \mu_2 = -\frac{2}{\beta}G_e,$$
$$\delta = 0.$$

4.6.3.4. Gent Model

The form of the strain-energy potential for the Gent model is:

$$W = -\frac{\mu J_m}{2}\ln\left(1-\frac{\bar{I}_1-3}{J_m}\right)+\frac{1}{d}\left(\frac{J^2-1}{2}-\ln J\right) \tag{4.212}$$

where:

μ = initial shear modulus of material (input on **TBDATA** commands with **TB**,HYPER)

J_m = limiting value of $\bar{I}_1 - 3$ (input on **TBDATA** commands with **TB**,HYPER)

d = material incompressibility parameter (input on **TBDATA** commands with **TB**,HYPER)

The initial bulk modulus is:

$$K = \frac{2}{d} \tag{4.213}$$

As the parameter J_m goes to infinity, the model is converted to Neo-Hookean form.

4.6.3.5. Mooney-Rivlin

This option includes two-, three-, five-, and nine-term Mooney-Rivlin models. The form of the strain-energy potential for a two-parameter Mooney-Rivlin model is:

$$W = c_{10}(\bar{I}_1-3)+c_{01}(\bar{I}_2-3)+\frac{1}{d}(J-1)^2 \tag{4.214}$$

where:

c_{10}, c_{01}, d = material constants (input on **TBDATA** commands with **TB**,HYPER)

The form of the strain-energy potential for a three-parameter Mooney-Rivlin model is

$$W = c_{10}(\bar{I}_1-3)+c_{01}(\bar{I}_2-3)+c_{11}(\bar{I}_1-3)(\bar{I}_2-3)+\frac{1}{d}(J-1)^2 \tag{4.215}$$

where:

c_{10}, c_{01}, c_{11}, d = material constants (input on **TBDATA** commands with **TB**,HYPER)

The form of the strain-energy potential for five-parameter Mooney-Rivlin model is:

$$\begin{aligned} W = {} & c_{10}(\bar{I}_1-3)+c_{01}(\bar{I}_2-3)+c_{20}(\bar{I}_1-3)^2 \\ & +c_{11}(\bar{I}_1-3)(\bar{I}_2-3)+c_{02}(\bar{I}_2-3)^2+\frac{1}{d}(J-1)^2 \end{aligned} \tag{4.216}$$

where:

c_{10}, c_{01}, c_{20}, c_{11}, c_{02}, d = material constants (input on **TBDATA** commands with **TB**,HYPER)

The form of the strain-energy potential for nine-parameter Mooney-Rivlin model is:

$$W = c_{10}(\bar{I}_1 - 3) + c_{01}(\bar{I}_2 - 3) + c_{20}(\bar{I}_1 - 3)^2 + c_{11}(\bar{I}_1 - 3)(\bar{I}_2 - 3) + c_{02}(\bar{I}_2 - 3)^2 + c_{30}(\bar{I}_1 - 3)^3 + c_{21}(\bar{I}_1 - 3)^2(\bar{I}_2 - 3) + c_{12}(\bar{I}_1 - 3)(\bar{I}_2 - 3)^2 + c_{03}(\bar{I}_2 - 3)^3 + \frac{1}{d}(J - 1)^2 \quad (4.217)$$

where:

c_{10}, c_{01}, c_{20}, c_{11}, c_{02}, c_{30}, c_{21}, c_{12}, c_{03}, d = material constants (input on **TBDATA** commands with **TB**,HYPER)

The initial shear modulus is given by:

$$\mu = 2(c_{10} + c_{01}) \quad (4.218)$$

The initial bulk modulus is:

$$K = \frac{2}{d} \quad (4.219)$$

4.6.3.6. Neo-Hookean

The form Neo-Hookean strain-energy potential is:

$$W = \frac{\mu}{2}(\bar{I}_1 - 3) + \frac{1}{d}(J - 1)^2 \quad (4.220)$$

where:

μ = initial shear modulus of materials (input on **TBDATA** commands with **TB**,HYPER)
d = material incompressibility parameter (input on **TBDATA** commands with **TB**,HYPER)

The initial bulk modulus is related to the material incompressibility parameter by:

$$K = \frac{2}{d} \quad (4.221)$$

where:

K = initial bulk modulus

4.6.3.7. Ogden Compressible Foam Model

The strain-energy potential of the Ogden compressible foam model is based on the principal stretches of left-Cauchy strain tensor, which has the form:

$$W = \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i}(J^{\alpha_i/3}(\bar{\lambda}_1^{\alpha_i} + \bar{\lambda}_2^{\alpha_i} + \bar{\lambda}_3^{\alpha_i}) - 3) + \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i\beta_i}(J^{-\alpha_i\beta_i} - 1) \quad (4.222)$$

where:

N = material constant (input as NPTS on **TB**,HYPER)
μ_i, α_i, β_i = material constants (input on **TBDATA** commands with **TB**,HYPER)

The initial shear modulus, μ, is given as:

$$\mu = \frac{\sum_{i=1}^{N} \mu_i \alpha_i}{2} \tag{4.223}$$

The initial bulk modulus K is defined by:

$$K = \sum_{i=1}^{N} \mu_i \alpha_i \left(\frac{1}{3} + \beta_i \right) \tag{4.224}$$

For N = 1, α_1 = -2, μ_1= –μ, and β = 0.5, the Ogden option is equivalent to the Blatz-Ko option.

4.6.3.8. Ogden Potential

The Ogden form of strain-energy potential is based on the principal stretches of left-Cauchy strain tensor, which has the form:

$$W = \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} (\bar{\lambda}_1^{\alpha_i} + \bar{\lambda}_2^{\alpha_i} + \bar{\lambda}_3^{\alpha_i} - 3) + \sum_{k=1}^{N} \frac{1}{d_k} (J - 1)^{2k} \tag{4.225}$$

where:

N = material constant (input as NPTS on **TB**,HYPER)
μ_i, α_i, d_k = material constants (input on **TBDATA** commands with **TB**,HYPER)

Similar to the Polynomial form, there is no limitation on N. A higher N can provide better fit the exact solution, however, it may, on the other hand, cause numerical difficulty in fitting the material constants and also it requests to have enough data to cover the entire range of interest of the deformation. Therefore a value of N > 3 is not usually recommended.

The initial shear modulus, μ, is given as:

$$\mu = \frac{1}{2} \sum_{i=1}^{N} \alpha_i \mu_i \tag{4.226}$$

The initial bulk modulus is:

$$K = \frac{2}{d_1} \tag{4.227}$$

For N = 1 and α_1 = 2, the Ogden potential is equivalent to the Neo-Hookean potential. For N = 2, α_1 = 2 and α_2 = -2, the Ogden potential can be converted to the 2 parameter Mooney-Rivlin model.

4.6.3.9. Polynomial Form

The polynomial form of strain-energy potential is

$$W = \sum_{i+j=1}^{N} c_{ij} (\bar{I}_1 - 3)^i (\bar{I}_2 - 3)^j + \sum_{k=1}^{N} \frac{1}{d_k} (J - 1)^{2k} \tag{4.228}$$

where:

N = material constant (input as NPTS on **TB**,HYPER)
c_{ij}, d_k = material constants (input on **TBDATA** commands with **TB**,HYPER)

In general, there is no limitation on N. A higher N may provide better fit the exact solution, however, it may, on the other hand, cause numerical difficulty in fitting the material constants and requires enough data to cover the entire range of interest of deformation. Therefore a very higher N value is not usually recommended.

The Neo-Hookean model can be obtained by setting N = 1 and c_{01} = 0. Also for N = 1, the two parameters Mooney-Rivlin model is obtained, for N = 2, the five parameters Mooney-Rivlin model is obtained and for N = 3, the nine parameters Mooney-Rivlin model is obtained.

The initial shear modulus is defined:

$$\mu = 2(c_{10} + c_{01}) \tag{4.229}$$

The initial bulk modulus is:

$$K = \frac{2}{d_1} \tag{4.230}$$

4.6.3.10. Yeoh Model

The Yeoh model is also called the reduced polynomial form. The strain-energy potential is:

$$W = \sum_{i=1}^{N} c_{i0} (\bar{I}_1 - 3)^i + \sum_{k=1}^{N} \frac{1}{d_k}(J-1)^{2k} \tag{4.231}$$

where:

N = material constant (input as NPTS on **TB**,HYPER)
C_{i0} = material constants (input on **TBDATA** commands with **TB**,HYPER)
d_k = material constants (input on **TBDATA** commands with **TB**,HYPER)

The Neo-Hookean model can be obtained by setting N = 1.

The initial shear modulus is defined:

$$\mu = 2c_{10} \tag{4.232}$$

The initial bulk modulus is:

$$K = \frac{2}{d_1} \tag{4.233}$$

4.6.4. Anisotropic Hyperelasticity

The anisotropic constitutive strain-energy density function W is:

$$W = W_v(J) + W_d(\bar{\mathbf{C}}, \mathbf{A} \otimes \mathbf{A}, \mathbf{B} \otimes \mathbf{B}) \tag{4.234}$$

where:

W_v = volumetric part of the strain energy
W_d = deviatoric part of strain energy (often called isochoric part of the strain energy)

It is assumed that the material is nearly incompressible or purely incompressible. The volumetric part W_v is absolutely independent of the isochoric part W_d.

The volumetric part, W_v, is assumed to be only function of J as:

$$W_v(J) = \frac{1}{d} \cdot (J-1)^2 \tag{4.235}$$

The isochoric part W_d is a function of the invariants $\bar{I}_1, \bar{I}_2, \bar{I}_4, \bar{I}_5, \bar{I}_6, \bar{I}_7, \bar{I}_8$ of the isochoric part of the right Cauchy Green tensor $\bar{\mathbf{C}}$ and the two constitutive material directions **A**, **B** in the undeformed configuration. The material directions yield so-called structural tensors $\mathbf{A} \otimes \mathbf{A}, \mathbf{B} \otimes \mathbf{B}$ of the microstructure of the material. Thus, the strain-energy density yields:

$$\begin{aligned} W_d(\mathbf{C}, \mathbf{A} \otimes \mathbf{A}, \mathbf{B} \otimes \mathbf{B}) = & \sum_{i=1}^{3} a_i (\bar{I}_1 - 3)^i + \sum_{j=1}^{3} b_j (\bar{I}_2 - 3)^j + \sum_{k=2}^{6} c_k (\bar{I}_4 - 1)^k \\ & + \sum_{l=2}^{6} d_l (\bar{I}_5 - 1)^l + \sum_{m=2}^{6} e_m (\bar{I}_6 - 1)^m + \sum_{n=2}^{6} f_n (\bar{I}_7 - 1)^n + \sum_{o=2}^{6} g_o (\bar{I}_8 - \varsigma)^o \end{aligned} \tag{4.236}$$

where:

$$|\mathbf{A}| = 1, |\mathbf{B}| = 1$$

Two strain energy potentials, as forms of polynomial or exponential functions, are available for characterizing the isochoric part of the strain-energy potential.

The third invariant $\bar{I}_3$ is ignored due to the incompressible assumption. The parameter ς is defined as:

$$\varsigma = (\mathbf{A} \cdot \mathbf{B})^2 \tag{4.237}$$

In Equation 4.236 (p. 133) the irreducible basis of invariants:

$$\begin{aligned} \bar{I}_1 &= \mathrm{tr}\bar{\mathbf{C}} & \bar{I}_2 &= \frac{1}{2}(\mathrm{tr}^2\bar{\mathbf{C}} - \mathrm{tr}\bar{\mathbf{C}}^2) \\ \bar{I}_4 &= \mathbf{A} \cdot \bar{\mathbf{C}}\mathbf{A} & \bar{I}_5 &= \mathbf{A} \cdot \bar{\mathbf{C}}^2\mathbf{A} \\ \bar{I}_6 &= \mathbf{B} \cdot \bar{\mathbf{C}}\mathbf{B} & \bar{I}_7 &= \mathbf{B} \cdot \bar{\mathbf{C}}^2\mathbf{B} \\ \bar{I}_8 &= (\mathbf{A} \cdot \mathbf{B})\mathbf{A} \cdot \bar{\mathbf{C}}\mathbf{B} \end{aligned} \tag{4.238}$$

The exponential-function-based strain energy potential function is given as:

$$W_d\left(\bar{\mathbf{C}}, \mathbf{A} \otimes \mathbf{A}, \mathbf{B} \otimes \mathbf{B}\right) = \sum_{i=1}^{3} a_i \left(\bar{I}_1 - 3\right)^i + \sum_{j=1}^{3} b_j \left(\bar{I}_2 - 3\right)^j + \frac{c_1}{2c_2}\left\{\exp\left[c_2\left(\bar{I}_4 - 1\right)^2\right] - 1\right\} + \frac{e_1}{2e_2}\left\{\exp\left[e_2\left(\bar{I}_6 - 1\right)^2\right] - 1\right\}$$

For information about the splitting of the volumetric and deviatoric parts of the strain-density function, see Finite Strain Elasticity (p. 125). For further information about this material model, see Anisotropic Hyperelasticity in the *Material Reference*.

4.6.5. USER Subroutine

The option of user subroutine allows users to define their own strain-energy potential. Use of subroutine `userhyper` is necessary to provide the derivatives of the strain-energy potential with respect to the

strain invariants. See the *Guide to ANSYS User Programmable Features* for more information about writing a user hyperelasticity subroutine.

4.6.6. Output Quantities

Stresses (output quantities S) are true (Cauchy) stresses in the global coordinate system. They are computed from the second Piola-Kirchhoff stresses using:

$$\sigma_{ij} = \frac{\rho}{\rho_o} f_{ik} S_{kl} f_{jl} = \frac{1}{\sqrt{I_3}} f_{ik} S_{kl} f_{jl} \tag{4.240}$$

where:

ρ, ρ_o = mass densities in the current and initial configurations

Strains (output as EPEL) are the Hencky (logarithmic) strains (see Equation 3.6 (p. 31)). They are in the global coordinate system. Thermal strain (output as EPTH) is reported as:

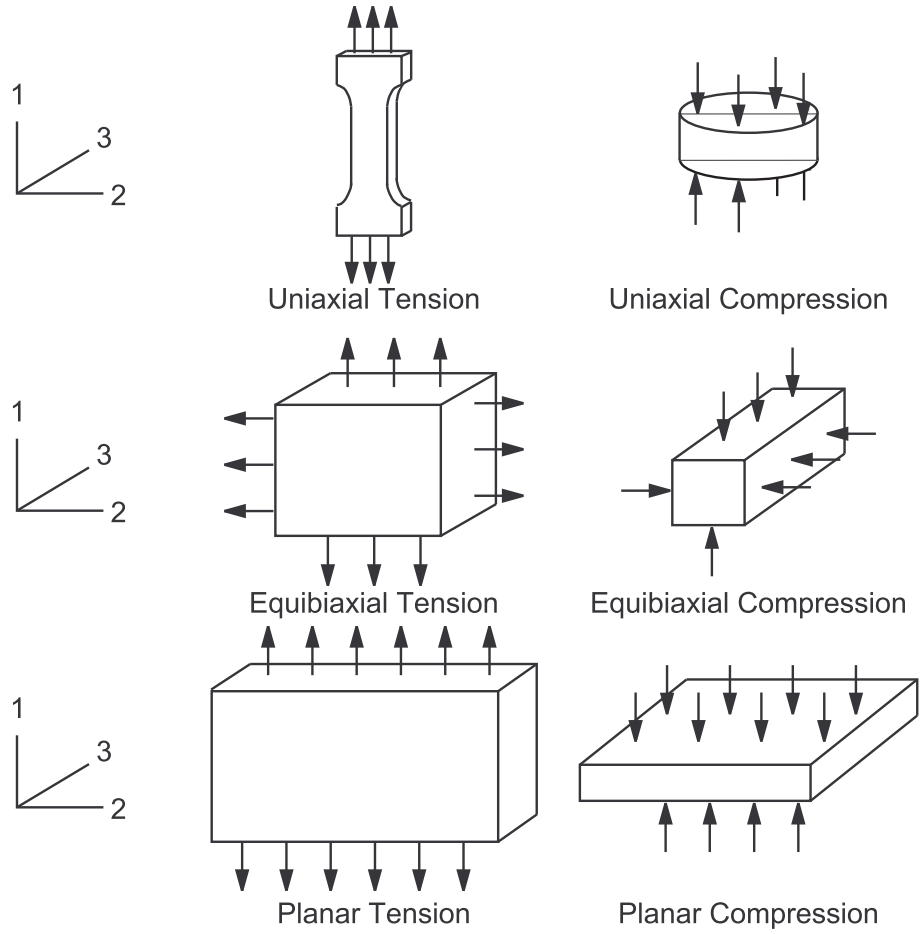
$$\varepsilon_{th} = \ln(1 + \alpha \Delta T) \tag{4.241}$$

4.6.7. Hyperelasticity Material Curve Fitting

The hyperelastic constants in the strain-energy density function of a material determine its mechanical response. Therefore, in order to obtain successful results during a hyperelastic analysis, it is necessary to accurately assess the material constants of the materials being examined. Material constants are generally derived for a material using experimental stress-strain data. It is recommended that this test data be taken from several modes of deformation over a wide range of strain values. In fact, it has been observed that to achieve stability, the material constants should be fit using test data in at least as many deformation states as will be experienced in the analysis.

For hyperelastic materials, simple deformation tests (consisting of six deformation modes) can be used to accurately characterize the material constants (see Material Curve Fitting in the *Structural Analysis Guide* for details). All the available laboratory test data will be used to determine the hyperelastic material constants. The six different deformation modes are graphically illustrated in Figure 4.26: Illustration of Deformation Modes (p. 135). Combinations of data from multiple tests will enhance the characterization of the hyperelastic behavior of a material.

Figure 4.26: Illustration of Deformation Modes

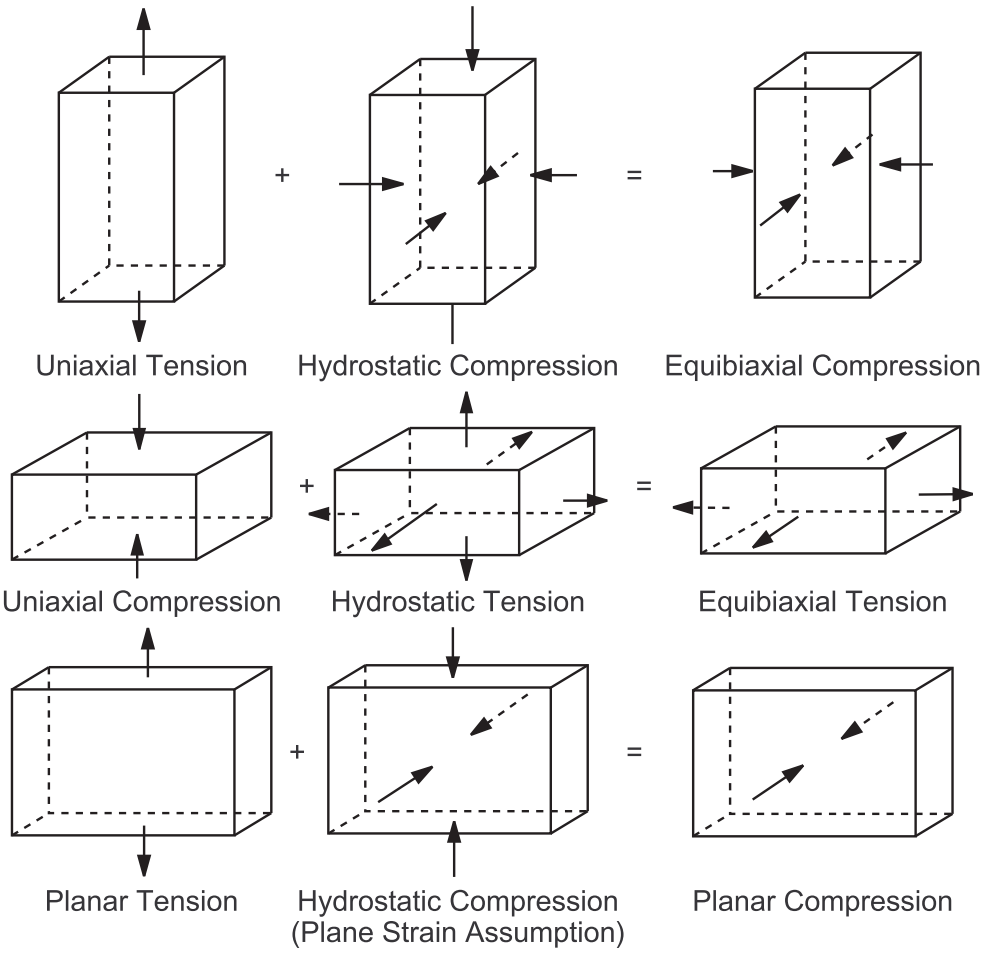


Although the algorithm accepts up to six different deformation states, it can be shown that apparently different loading conditions have identical deformations, and are thus equivalent. Superposition of tensile or compressive hydrostatic stresses on a loaded incompressible body results in different stresses, but does not alter deformation of a material. As depicted in Figure 4.27: Equivalent Deformation Modes (p. 136), we find that upon the addition of hydrostatic stresses, the following modes of deformation are identical:

1. Uniaxial Tension and Equibiaxial Compression.
2. Uniaxial Compression and Equibiaxial Tension.
3. Planar Tension and Planar Compression.

With several equivalent modes of testing, we are left with only three independent deformation states for which one can obtain experimental data.

Figure 4.27: Equivalent Deformation Modes



The following sections outline the development of hyperelastic stress relationships for each independent testing mode. In the analyses, the coordinate system is chosen to coincide with the principal directions of deformation. Thus, the right Cauchy-Green strain tensor can be written in matrix form by:

$$[C] = \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix} \tag{4.242}$$

where:

$\lambda_i = 1 + \varepsilon_i \equiv$ principal stretch ratio in the ith direction

ε_i = principal value of the engineering strain tensor in the ith direction

The principal invariants of C_{ij} are:

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \tag{4.243}$$

$$I_2 = \lambda_1^2\lambda_2^2 + \lambda_1^2\lambda_3^2 + \lambda_2^2\lambda_3^2 \tag{4.244}$$

$$I_3 = \lambda_1^2\lambda_2^2\lambda_3^2 \tag{4.245}$$

For each mode of deformation, fully incompressible material behavior is also assumed so that third principal invariant, I_3, is identically one:

$$\lambda_1^2\lambda_2^2\lambda_3^2 = 1 \tag{4.246}$$

Finally, the hyperelastic Piola-Kirchhoff stress tensor, Equation 4.189 (p. 125) can be algebraically manipulated to determine components of the Cauchy (true) stress tensor. In terms of the left Cauchy-Green strain tensor, the Cauchy stress components for a volumetrically constrained material can be shown to be:

$$\sigma_{ij} = -p\delta_{ij} + \text{dev}\left[2\frac{\partial W}{\partial I_1}b_{ij} - 2I_3\frac{\partial W}{\partial I_2}b_{ij}^{-1}\right] \tag{4.247}$$

where:

p = pressure

$b_{ij} = F_{ik}F_{jk}$ = Left Cauchy-Green deformation tensor

4.6.7.1. Uniaxial Tension (Equivalently, Equibiaxial Compression)

As shown in Figure 4.26: Illustration of Deformation Modes (p. 135), a hyperelastic specimen is loaded along one of its axis during a uniaxial tension test. For this deformation state, the principal stretch ratios in the directions orthogonal to the 'pulling' axis will be identical. Therefore, during uniaxial tension, the principal stretches, λ_i, are given by:

$$\lambda_1 = \text{stretch in direction being loaded} \tag{4.248}$$

$$\lambda_2 = \lambda_3 = \text{stretch in directions not being loaded} \tag{4.249}$$

Due to incompressibility Equation 4.246 (p. 137):

$$\lambda_2\lambda_3 = \lambda_1^{-1} \tag{4.250}$$

and with Equation 4.249 (p. 137),

$$\lambda_2 = \lambda_3 = \lambda_1^{-1/2} \tag{4.251}$$

For uniaxial tension, the first and second strain invariants then become:

$$I_1 = \lambda_1^2 + 2\lambda_1^{-1} \tag{4.252}$$

and

$$I_2 = 2\lambda_1 + \lambda_1^{-2} \tag{4.253}$$

Substituting the uniaxial tension principal stretch ratio values into the Equation 4.247 (p. 137), we obtain the following stresses in the 1 and 2 directions:

$$\sigma_{11} = -p + 2\partial W/\partial I_1\ \lambda_1^2 - 2\partial W/\partial I_2\ \lambda_1^{-2} \tag{4.254}$$

and

$$\sigma_{22} = -p + 2\partial W/\partial I_1\ \lambda_1^{-1} - 2\partial W/\partial I_2\ \lambda_1 = 0 \tag{4.255}$$

Subtracting Equation 4.255 (p. 137) from Equation 4.254 (p. 137), we obtain the principal true stress for uniaxial tension:

$$\sigma_{11} = 2(\lambda_1^2 - \lambda_1^{-1})[\partial W/\partial I_1 + \lambda_1^{-1}\,\partial W/\partial I_2] \tag{4.256}$$

The corresponding engineering stress is:

$$T_1 = \sigma_{11}\lambda_1^{-1} \tag{4.257}$$

4.6.7.2. Equibiaxial Tension (Equivalently, Uniaxial Compression)

During an equibiaxial tension test, a hyperelastic specimen is equally loaded along two of its axes, as shown in Figure 4.26: Illustration of Deformation Modes (p. 135). For this case, the principal stretch ratios in the directions being loaded are identical. Hence, for equibiaxial tension, the principal stretches, λ_i, are given by:

$$\lambda_1 = \lambda_2 = \text{stretch ratio in direction being loaded} \tag{4.258}$$

$$\lambda_3 = \text{stretch in direction not being loaded} \tag{4.259}$$

Utilizing incompressibility Equation 4.246 (p. 137), we find:

$$\lambda_3 = \lambda_1^{-2} \tag{4.260}$$

For equibiaxial tension, the first and second strain invariants then become:

$$I_1 = 2\lambda_1^2 + \lambda_1^{-4} \tag{4.261}$$

and

$$I_2 = \lambda_1^4 + 2\lambda_1^{-2} \tag{4.262}$$

Substituting the principal stretch ratio values for equibiaxial tension into the Cauchy stress Equation 4.247 (p. 137), we obtain the stresses in the 1 and 3 directions:

$$\sigma_{11} = -p + 2\,\partial W/\partial I_1\ \lambda_1^2 - 2\,\partial W/\partial I_2\ \lambda_1^{-2} \tag{4.263}$$

and

$$\sigma_{33} = -p + 2\,\partial W/\partial I_1\,\lambda_1^{-4} - 2\,\partial W/\partial I_2\,\lambda_1^4 = 0 \tag{4.264}$$

Subtracting Equation 4.264 (p. 138) from Equation 4.263 (p. 138), we obtain the principal true stress for equibiaxial tension:

$$\sigma_{11} = 2(\lambda_1^2 - \lambda_1^{-4})[\partial W/\partial I_1 + \lambda_1^2\,\partial W/\partial I_2] \tag{4.265}$$

The corresponding engineering stress is:

$$T_1 = \sigma_{11}\lambda_1^{-1} \tag{4.266}$$

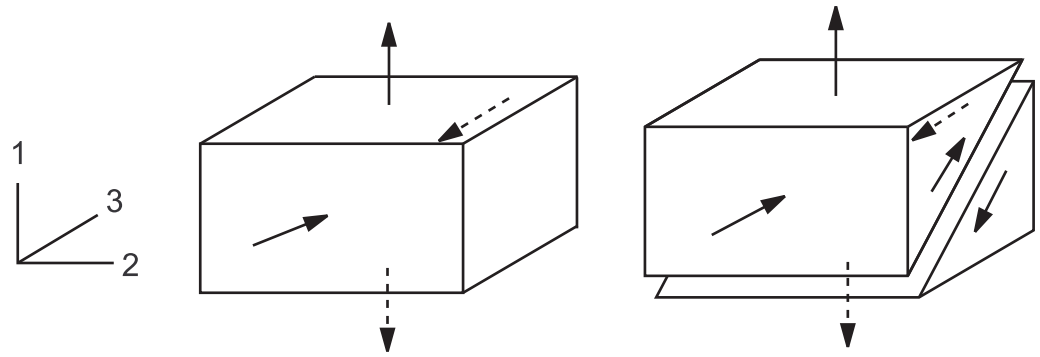
4.6.7.3. Pure Shear

(Uniaxial Tension and Uniaxial Compression in Orthogonal Directions)

Pure shear deformation experiments on hyperelastic materials are generally performed by loading thin, short and wide rectangular specimens, as shown in Figure 4.28: Pure Shear from Direct Compon-

ents (p. 139). For pure shear, plane strain is generally assumed so that there is no deformation in the 'wide' direction of the specimen: $\lambda_2 = 1$.

Figure 4.28: Pure Shear from Direct Components



Due to incompressibility Equation 4.246 (p. 137), it is found that:

$$\lambda_3 = \lambda_1^{-1} \tag{4.267}$$

For pure shear, the first and second strain invariants are:

$$I_1 = \lambda_1^2 + \lambda_1^{-2} + 1 \tag{4.268}$$

and

$$I_2 = \lambda_1^2 + \lambda_1^{-2} + 1 \tag{4.269}$$

Substituting the principal stretch ratio values for pure shear into the Cauchy stress Equation 4.247 (p. 137), we obtain the following stresses in the 1 and 3 directions:

$$\sigma_{11} = -p + 2\partial W/\partial I_1 \lambda_1^2 - 2\partial W/\partial I_2 \lambda_1^{-2} \tag{4.270}$$

and

$$\sigma_{33} = -p + 2\partial W/\partial I_1 \lambda_1^{-2} - 2\partial W/\partial I_2 \lambda_1^2 = 0 \tag{4.271}$$

Subtracting Equation 4.271 (p. 139) from Equation 4.270 (p. 139), we obtain the principal pure shear true stress equation:

$$\sigma_{11} = 2(\lambda_1^2 - \lambda_1^{-2})[\partial W/\partial I_1 + \partial W/\partial I_2] \tag{4.272}$$

The corresponding engineering stress is:

$$T_1 = \sigma_{11}\lambda_1^{-1} \tag{4.273}$$

4.6.7.4. Volumetric Deformation

The volumetric deformation is described as:

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda, J = \lambda^3 \tag{4.274}$$

As nearly incompressible is assumed, we have:

$$\lambda \approx 1 \tag{4.275}$$

The pressure, P, is directly related to the volume ratio J through:

$$P = \frac{\partial W}{\partial J} \tag{4.276}$$

4.6.7.5. Least Squares Fit Analysis

By performing a least squares fit analysis the Mooney-Rivlin constants can be determined from experimental stress-strain data and Equation 4.255 (p. 137), Equation 4.265 (p. 138), and Equation 4.272 (p. 139). Briefly, the least squares fit minimizes the sum of squared error between experimental and Cauchy predicted stress values. The sum of the squared error is defined by:

$$E = \sum_{i=1}^{n} (T_i^E - T_i(c_j))^2 \tag{4.277}$$

where:

E = least squares residual error

T_i^E = experimental stress values

$T_i(C_j)$ = engineering stress values (function of hyperelastic material constants

n = number of experimental data points

Equation 4.277 (p. 140) is minimized by setting the variation of the squared error to zero: $\delta E^2 = 0$. This yields a set of simultaneous equations which can be used to solve for the hyperelastic constants:

$$\begin{aligned} &\partial E^2 / \partial C_1 = 0 \\ &\partial E^2 / \partial C_2 = 0 \\ &\quad \vdots \\ &\text{etc.} \end{aligned} \tag{4.278}$$

It should be noted that for the pure shear case, the hyperelastic constants cannot be uniquely determined from Equation 4.272 (p. 139). In this case, the shear data must by supplemented by either or both of the other two types of test data to determine the constants.

4.6.8. Experimental Response Functions

From Equation 4.247 (p. 137), the deviatoric stress is determined solely by the deformation and the derivatives of the elastic potential function $\partial W / \partial I_1$, $\partial W / \partial I_2$, and $\partial W / \partial J$. These derivatives are called the response functions and are determined analytically for the hyperelastic potentials in Isotropic Hyperelasticity (p. 127) and Anisotropic Hyperelasticity (p. 132). These response functions can also be determined directly from experimental data, bypassing the need to fit the potential function's parameters to the experimental data.

The constitutive Equation 4.256 (p. 138), Equation 4.265 (p. 138), and Equation 4.272 (p. 139) give an explicit relationship between the stress, deformation, and response functions. The experimental data consists of the measured deformation and stress so that the only unknowns in the constitutive equations are the response functions.

The volumetric response function ($\partial W / \partial J$) is determined either analytically (from the polynomial volumetric potential function given as the second term on the right side of Equation 4.228 (p. 131)) or as experimental data of volume ratio versus pressure. Given the experimental data, the volume ratio for a general deformation is used to determine the pressure and hence the volumetric response function.

The response functions for the first and second deformation invariant are determined from the experimental data from uniaxial tension, equibiaxial tension, pure shear or combined uniaxial tension and compression experiments. Additionally, for incompressible materials, uniaxial compression experiments are equivalent to equibiaxial tension and can be used in place of equibiaxial data to determine the response functions.

Combined uniaxial tension plus compression data cannot be combined with other data sets (except pressure-volume), and gives only a material behavior that depends on the first invariant response function. Combinations of the other experimental data sets can be used to determine the first and second invariant response functions. Any combination of uniaxial tension, equibiaxial tension or pure shear data can be used.

For a single set of experimental data, the respective constitutive equation has two unknown response functions. Setting the second invariant response function ($\partial W / \partial I_2$) to zero, the constitutive equation can be solved for the first invariant response function ($\partial W / \partial I_1$).

Given two sets of experimental data for independent deformations (uniaxial-equibiaxial, uniaxial-pure shear, or equibiaxial-pure shear), the resulting two equations can be solved for the two unknown response functions. With experimental data for uniaxial tension, equibiaxial tension, and pure shear deformations, the resulting three constitutive equations can be solved for a best fit of the two response functions.

The response functions can therefore be determined from experimental data over the range of experimental deformation; however, the constitutive response is needed for any general deformation. The constitutive response is obtained by determining an experimental deformation that is nearest to the general deformation, where the definition of nearest is that the general deformation and the experimental deformation have the same first invariant I_1.

For example, given any arbitrary deformation state, Equation 4.243 (p. 136) gives the equation for the first invariant. Equating this equation to the uniaxial deformation first invariant of Equation 4.252 (p. 137) gives

$$\lambda_1^2 + \lambda_2^2 + \lambda_3^2 = \lambda^2 + \frac{2}{\lambda} \tag{4.279}$$

where:

λ_1,λ_2, and λ_3 = stretches for the general deformation
λ = unknown uniaxial stretch that yields an equivalent value of the first invariant I_1.

Solving the resulting equation for the tensile value of λ gives the uniaxial deformation that is nearest to the general deformation. Then, λ is used to determine the response function from the uniaxial experimental data, which corresponds to the required response function for the general deformation.

In general, Equation 4.279 (p. 141) has two valid solutions, one in tension and one in compression. For use with the uniaxial tension experimental data, the tensile solution is used. For use with the combined uniaxial tension and compression data, the tension or compression λ value is chosen so that the differ-

ence between the uniaxial deformation gradient's eigenvalues are closest to the eigenvalues of the actual deformation gradient.

The nearest equibiaxial and pure shear deformations are determined in a similar manner by equating the first invariant for the general deformation with the first invariant for the equibiaxial or pure shear deformation. Then the experimental stretch is determined by choosing the tensile value from the resulting valid solutions for λ.

4.6.9. Material Stability Check

Stability checks are provided for the Mooney-Rivlin hyperelastic materials. A nonlinear material is stable if the secondary work required for an arbitrary change in the deformation is always positive. Mathematically, this is equivalent to:

$$d\sigma_{ij}d\varepsilon_{ij} > 0 \tag{4.280}$$

where:

$d\sigma$ = change in the Cauchy stress tensor corresponding to a change in the logarithmic strain

Since the change in stress is related to the change in strain through the material stiffness tensor, checking for stability of a material can be more conveniently accomplished by checking for the positive definiteness of the material stiffness.

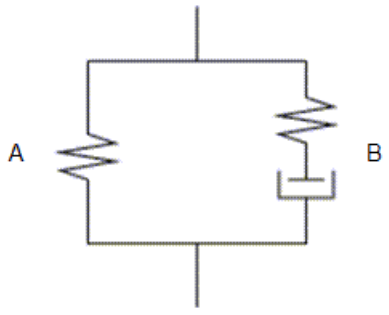
The material stability checks are done at the end of preprocessing but before an analysis actually begins. At that time, the program checks for the loss of stability for six typical stress paths (uniaxial tension and compression, equibiaxial tension and compression, and planar tension and compression). The range of the stretch ratio over which the stability is checked is chosen from 0.1 to 10. If the material is stable over the range then no message will appear. Otherwise, a warning message appears that lists the Mooney-Rivlin constants and the critical values of the nominal strains where the material first becomes unstable.

4.7. Bergstrom-Boyce

The Bergstrom-Boyce material model (**TB**,BB) is a phenomenological-based, highly nonlinear material model used to model typical elastomers and biological materials. The model allows for a nonlinear stress-strain relationship, creep, and rate-dependence.

The Bergstrom-Boyce model is based on a spring (A) in parallel with a spring and damper (B) in series, as shown in Figure 4.29: Bergstrom-Boyce Material Model Representation (p. 142). The material model is associated with time-dependent stress-strain relationships without complete stress relaxation. All components (springs and damper) are highly nonlinear.

Figure 4.29: Bergstrom-Boyce Material Model Representation



The stress state in A can be found in the tensor form of the deformation gradient tensor ($F = dx_i / dX_j$) and material parameters, as follows:

$$\sigma_A = \frac{1}{J_A}\frac{\mu_A}{3}\frac{L^{-1}\left(\bar{\lambda}_A^* / \lambda_A^{lock}\right)}{\bar{\lambda}_A^* / \lambda_A^{lock}}\mathrm{dev}\left[\tilde{B}_A^*\right] + K\left[J_A - 1\right]\tilde{I} \tag{4.281}$$

where

σ_A = stress state in A

μ_A = initial shear modulus of A

λ_A^{lock} = limiting chain stretch of A

K = bulk modulus

J_A = det[F]

$\tilde{B}_A^* = J^{-2/3}\tilde{F}\tilde{F}^T$

$\bar{\lambda}_A^* = \sqrt{\mathrm{tr}[\tilde{B}^*]/3}$

$L^{-1}(x)$ = inverse Langevin function, where the Langevin function is given by Equation 4.282 (p. 143):

For numerical efficiency, the Padé ([373] (p. 942)) approximation shown below is used for the inverse Langevin function (). This approximation will differ from the polynomial approximation used for the Arruda-Boyce model.

$$L^{-1}(x) \approx x\frac{3 - x^2}{1 - x^2} \tag{4.282}$$

The stress in the viscoelastic component of the material (B) is a function of the deformation and the rate of deformation. Of the total deformation in B, a portion takes place in the elastic component while the rest of the deformation takes place in the viscous component. Because the stress in the elastic portion is equal to the stress plastic portion, the total stress can be written merely as a function of the elastic deformation, as shown in Equation 4.283 (p. 143):

$$\sigma_B = \frac{1}{J_B^e}\frac{\mu_B}{3}\frac{L^{-1}\left(\bar{\lambda}_B^{e*} / \lambda_B^{lock}\right)}{\bar{\lambda}_B^{e*} / \lambda_B^{lock}}\mathrm{dev}\left[\tilde{B}_B^{e*}\right] + K\left[J_B^e - 1\right]\tilde{I} \tag{4.283}$$

All variables in this equation are analogous to the variables in Equation 4.281 (p. 143). The viscous deformation can be found from the total deformation and the elastic deformation: $F_B^p = \left[F_B^e\right]^{-1} * F$

Correct solutions for F_B^p and F_B^e will satisfy:

$$\dot{\tilde{F}}_B^p\left(\tilde{F}_B^p\right)^{-1} = \dot{\gamma}_B\tilde{N}_B \tag{4.284}$$

where

$\tilde{N}_B = \tilde{T}_B / \tau$ is the direction of the stress tensor,

$\dot{\gamma}_B = \dot{\gamma}_0 \left(\bar{\lambda}_B^p - 1 + \epsilon \right)^C \left[\frac{\tau / \sqrt{2}}{\tau_{base}} \right]^m$ is the effective creep rate,

$\tau = \mathrm{tr}\left[\tilde{T}_B \tilde{T}_B \right]^{0.5}$ is the Frobenius norm of the stress tensor,

$\bar{\lambda}_B^p = \sqrt{\frac{\mathrm{tr}\left[(\tilde{F}_B^p)^T \tilde{F}_B^p \right]}{3}}$ is the inelastic network stretch,

and $\dot{\gamma}_0 / \tau_{base}^m$, ϵ, C, and m are material input parameters.

As $\dot{\gamma}_B$ is a function of the deformation ($\bar{\lambda}_B^p$) and τ is based on the stress tensor, Equation 4.284 (p. 143) is expanded to:

$$\dot{\tilde{F}}_B^p \left(\tilde{F}_B^p \right)^{-1} = \dot{\gamma}_0 \left(\bar{\lambda}_B^p - 1 + \epsilon \right)^C \left[\frac{\tau / \sqrt{2}}{\tau_{base}} \right]^m \tilde{N}_B \tag{4.285}$$

Once Equation 4.285 (p. 144) is satisfied, the corresponding stress tensor from component B is added to the stress tensor from component A to find the total stress, as shown in Equation 4.286 (p. 144):

$$\sigma_{tot} = \sigma_A + \sigma_B \tag{4.286}$$

For more information about the constitutive model, see references [371] (p. 941) and [372] (p. 942). For more information about the implementation, see reference [373] (p. 942).

4.8. Mullins Effect

The Mullins effect (**TB**,CDM) is a phenomenon typically observed in compliant filled polymers. It is characterized by a decrease in material stiffness during loading and is readily observed during cyclic loading as the material response along the unloading path differs noticeably from the response that along the loading path. Although the details about the mechanisms responsible for the Mullins effect have not yet been settled, they might include debonding of the polymer from the filler particles, cavitation, separation of particle clusters, and rearrangement of the polymer chains and particles.

In the body of literature that exists concerning this phenomenon, a number of methods have been proposed as constitutive models for the Mullins effect. The model is a maximum load modification to the nearly- and fully-incompressible hyperelastic constitutive models already available. In this model, the virgin material is modeled using one of the available hyperelastic potentials, and the Mullins effect modifications to the constitutive response are proportional to the maximum load in the material history.

4.8.1. The Pseudo-elastic Model

The Ogden-Roxburgh [378] (p. 942) pseudo-elastic model (**TB**,CDM,,,,PSE2) of the Mullins effect is a modification of the standard thermodynamic formulation for hyperelastic materials and is given by:

$$W(F_{ij},\eta) = \eta W_0(F_{ij}) + \phi(\eta) \tag{4.287}$$

where

W_0(Fij) = virgin material deviatoric strain-energy potential
η = evolving scalar damage variable
$\Phi(\eta)$ = damage function

The arbitrary limits $0 < \eta \leq 1$ are imposed with $\eta = 1$ defined as the state of the material without any changes due to the Mullins effect. Then, along with equilibrium, the damage function is defined by:

$$\begin{aligned} &\phi(1) = 0 \\ &\phi'(\eta) = -W_0(F_{ij}) \end{aligned} \tag{4.288}$$

which implicitly defines the Ogden-Roxburgh parameter η. Using Equation 4.288 (p. 145), deviatoric part of the second Piola-Kirchhoff stress tensor is then:

$$\begin{aligned} S_{ij} &= 2\frac{\partial W}{\partial C_{ij}} \\ &= \eta 2\frac{\partial W_0}{\partial C_{ij}} \\ &= \eta S_{ij}^0 \end{aligned} \tag{4.289}$$

The modified Ogden-Roxburgh damage function [379] (p. 942) has the following functional form of the damage variable:

$$\eta = 1 - \frac{1}{r}\,\mathrm{erf}\left[\frac{W_m - W_0}{m + \beta W_m}\right] \tag{4.290}$$

where r, m, and β are material parameters and W_m is the maximum virgin potential over the time interval $t \in [0, t_0]$:

$$W_m = \max_{t\in[0,t_0]}[W_0(t)] \tag{4.291}$$

The tangent stiffness tensor $\mathbb{D}_{ijkl}$ for a constitutive model defined by Equation 4.287 (p. 145) is expressed as follows:

$$\begin{aligned} \mathbb{D}_{ijkl} &= 4\frac{\partial^2 W}{\partial C_{ij}\partial C_{kl}} \\ &= 4\eta\frac{\partial^2 W_0}{\partial C_{ij}\partial C_{kl}} + 4\frac{\partial W_0}{\partial C_{ij}}\frac{\partial \eta}{\partial C_{kl}} \end{aligned} \tag{4.292}$$

The differential for η in Equation 4.290 (p. 145) is:

$$\frac{\partial \eta}{\partial C_{ij}} = \frac{2}{\sqrt{\pi r}\,(m+\beta W_m)} e^{\left(\frac{W_m - W_0}{m+\beta W_m}\right)} \frac{\partial W_0}{\partial C_{ij}} \tag{4.293}$$

4.9. Viscoelasticity

A material is said to be viscoelastic if the material has an elastic (recoverable) part as well as a viscous (nonrecoverable) part. Upon application of a load, the elastic deformation is instantaneous while the viscous part occurs over time.

The viscoelastic model usually depicts the deformation behavior of glass or glass-like materials and may simulate cooling and heating sequences of such material. These materials at high temperatures turn into viscous fluids and at low temperatures behave as solids. Further, the material is restricted to be thermorheologically simple (TRS), which assumes the material response to a load at a high temperature over a short duration is identical to that at a lower temperature but over a longer duration. The material model is available with elements LINK180, SHELL181, PLANE182, PLANE183, SOLID185, SOLID186, SOLID187, BEAM188, BEAM189, SOLSH190, SHELL208, SHELL209, REINF264, REINF265, SOLID272, SOLID273, SHELL281, SOLID285, PIPE288, PIPE289, and ELBOW290 for small-deformation and large-deformation viscoelasticity.

The following topics related to viscoelasticity are available:

4.9.1. Small Strain Viscoelasticity

In this section, the constitutive equations and the numerical integration scheme for small strain viscoelasticity are discussed. Large-strain viscoelasticity is presented in Large-Deformation Viscoelasticity (p. 150).

4.9.2. Constitutive Equations

A material is viscoelastic if its stress response consists of an elastic part and viscous part. Upon application of a load, the elastic response is instantaneous while the viscous part occurs over time. Generally, the stress function of a viscoelastic material is given in an integral form. Within the context of small strain theory, the constitutive equation for an isotropic viscoelastic material can be written as:

$$\sigma = \int_0^t 2G(t-\tau)\frac{de}{d\tau}d\tau + I\int_0^t K(t-\tau)\frac{d\Delta}{d\tau}\,d\tau \tag{4.294}$$

where:

σ = Cauchy stress
e = deviatoric part of the strain
Δ = volumetric part of the strain

G(t) = shear relaxation kernel function
K(t) = bulk relaxation kernel function
t = current time
τ = past time
I = unit tensor

For the elements LINK180, SHELL181, PLANE182, PLANE183, SOLID185, SOLID186, SOLID187, BEAM188, SOLSH190, SHELL208, SHELL209, REINF264, REINF265, SOLID272, SOLID273, SHELL281, SOLID285, PIPE288, PIPE289, and ELBOW290, the kernel functions are represented in terms of Prony series, which assumes that:

$$G = G_\infty + \sum_{i=1}^{n_G} G_i \exp\left(-\frac{t}{\tau_i^G}\right) \tag{4.295}$$

$$K = K_\infty + \sum_{i=1}^{n_K} K_i \exp\left(-\frac{t}{\tau_i^K}\right) \tag{4.296}$$

where:

G_∞, G_i = shear elastic moduli

K_∞, K_i = bulk elastic moduli

τ_i^G, τ_i^K = relaxation times for each Prony component

Introducing the relative moduli:

$$\alpha_i^G = G_i / G_0 \tag{4.297}$$

$$\alpha_i^K = K_i / K_0 \tag{4.298}$$

where:

$$G_0 = G_\infty + \sum_{i=1}^{n_G} G_i$$

$$K_0 = K_\infty + \sum_{i=1}^{n_K} K_i$$

The kernel functions can be equivalently expressed as:

$$G = G_0\left[\alpha_\infty^G + \sum_{i=1}^{n_G} \alpha_i^G \exp\left(-\frac{t}{\tau_i^G}\right)\right], \qquad K = K_0\left[\alpha_\infty^K + \sum_{i=1}^{n_K} \alpha_i^K \exp\left(-\frac{t}{\tau_i^K}\right)\right] \tag{4.299}$$

The integral function Equation 4.294 (p. 146) can recover the elastic behavior at the limits of very slow and very fast load. Here, G_0 and K_0 are, respectively, the shear and bulk moduli at the fast load limit (i.e. the instantaneous moduli), and G_∞ and K_∞ are the moduli at the slow limit. The elasticity parameters input correspond to those of the fast load limit. Moreover by admitting Equation 4.295 (p. 147), the deviatoric and volumetric parts of the stress are assumed to follow different relaxation behavior.

The number of Prony terms for shear n_G and for volumetric behavior n_K need not be the same, nor do the relaxation times τ_i^G and τ_i^K.

The Prony representation has a prevailing physical meaning in that it corresponds to the solution of the classical differential model (the parallel Maxwell model) of viscoelasticity. This physical rooting is the key to understand the extension of the above constitutive equations to large-deformation cases as well as the appearance of the time-scaling law (for example, pseudo time) at the presence of time-dependent viscous parameters.

4.9.3. Numerical Integration

To perform finite element analysis, the integral Equation 4.294 (p. 146) need to be integrated. The integration scheme proposed by Taylor([112] (p. 926)) and subsequently modified by Simo([327] (p. 939)) is adapted. We will delineate the integration procedure for the deviatoric stress. The pressure response can be handled in an analogous way. To integrate the deviatoric part of Equation 4.294 (p. 146), first, break the stress response into components and write:

$$s = s_\infty + \sum_i^{n_G} s_i \tag{4.300}$$

where:

s = deviatoric stress

$S_\infty = 2G_\infty e$

In addition,

$$s_i = \int_0^t 2G_i \exp\left(-\frac{t-\tau}{\tau_i^G}\right)\frac{de}{d\tau}d\tau \tag{4.301}$$

One should note that

$$\begin{aligned}(s_i)_{n+1} &= \int_0^{t_{n+1}} 2G_i \exp\left(-\frac{t_{n+1}-\tau}{\tau_i^G}\right)\frac{de}{d\tau}d\tau \\ &= \int_0^{t_n} 2G_i \exp\left(-\frac{t_n+\Delta t-\tau}{\tau_i^G}\right)\frac{de}{d\tau}d\tau \\ &\quad + \int_{t_n}^{t_{n+1}} 2G_i \exp\left(-\frac{t_n-\tau}{\tau_i^G}\right)\frac{de}{d\tau}d\tau\end{aligned} \tag{4.302}$$

where:

$\Delta t = t_{n+1} - t_n$.

The first term of Equation 4.302 (p. 148) is readily recognized as: $\exp(-\frac{\Delta t}{\tau_i^G})(s_i)_n$.

Using the middle point rule for time integration for the second term, a recursive formula can be obtained as:

$$(s_i)_{n+1} = \exp\left(-\frac{\Delta t}{\tau_i^G}\right)(s_i)_n + 2\exp\left(-\frac{\Delta t}{2\tau_i^G}\right)G_i\Delta e \tag{4.303}$$

where:

$\Delta e = e_{n+1} - e_n$.

4.9.4. Thermorheological Simplicity

Materials viscous property depends strongly on temperature. For example, glass-like materials turn into viscous fluids at high temperatures while behave like solids at low temperatures. In reality, the temperature effects can be complicated. The so called *thermorheological simplicity* is an assumption based on the observations for many glass-like materials, of which the relaxation curve at high temperature is identical to that at a low temperature if the time is properly scaled (Scherer([326] (p. 939))). In essence, it stipulates that the relaxation times (of all Prony components) obey the scaling law

$$\tau_i^G(T) = \frac{\tau_i^G(T_r)}{A(T,T_r)}, \qquad \tau_i^K(T) = \frac{\tau_i^K(T_r)}{A(T,T_r)} \tag{4.304}$$

Here, $A(T, T_r)$ is called the shift function. Under this assumption (and in conjunction with the differential model), the deviatoric stress function can be shown to take the form

$$s = \int_0^t 2\left[G_\infty + \sum_{i=1}^{n_G} G_i \exp\left(-\frac{\xi_t - \xi_s}{\tau_i^G}\right)\right]\frac{de}{d\tau}d\tau \tag{4.305}$$

likewise for the pressure part. Here, notably, the Prony representation still holds with the time t, τ in the integrand being replaced by:

$$\xi_t = \int_0^t \exp(At)d\tau \text{ and } \xi_s = \int_0^s \exp(At)d\tau$$

here ξ is called *pseudo* (or reduced) time. In Equation 4.305 (p. 149), τ_i^G is the decay time at a given temperature.

The assumption of thermorheological simplicity allows for not only the prediction of the relaxation time over temperature, but also the simulation of mechanical response under prescribed temperature histories. In the latter situation, *A* is an implicit function of time *t* through T = T(t). In either case, the stress equation can be integrated in a manner similar to Equation 4.300 (p. 148). Indeed,

$$\begin{aligned}(s_i)_{n+1} &= \int_0^{t_{n+1}} 2G_i \exp\left(-\frac{\xi_{n+1} - \xi_n}{\tau_i^G}\right)\frac{de}{d\tau}d\tau \\ &= \int_0^{t_n} 2G_i \exp\left(-\frac{\Delta\xi + \xi_n - \xi_s}{\tau_i^G}\right)\frac{de}{d\tau}d\tau \\ &\quad + \int_{t_n}^{t_{n+1}} 2G_i \exp\left(-\frac{\xi_{n+1} - \xi_s}{\tau_i^G}\right)\frac{de}{d\tau}d\tau\end{aligned} \tag{4.306}$$

Using the middle point rule for time integration on Equation 4.306 (p. 149) yields

$$(s_i)_{n+1} = \exp\left(-\frac{\Delta\xi}{\tau_i^G}\right)(s_i)_n + 2\exp\left(-\frac{\Delta\xi_{\frac{1}{2}}}{\tau_i^G}\right)G_i\Delta e \tag{4.307}$$

where:

$$\Delta\xi = \int_{t_n}^{t_{n+1}} A(T(\tau))d\tau$$

$$\Delta\xi_{\frac{1}{2}} = \int_{t_{n+\frac{1}{2}}}^{t_{n+1}} A(T(\tau))d\tau$$

Two widely used shift functions, namely the Williams-Landel-Ferry shift function and the Tool-Naray-anaswamy shift function, are available. The form of the functions are given in Shift Functions (p. 153).

4.9.5. Large-Deformation Viscoelasticity

Two types of large-deformation viscoelasticity models are implemented: large-deformation small strain, and large-deformation large strain. The first is associated with hypo-type constitutive equations, and the latter is based on hyperelasticity, which includes isotropic and anisotropic hyperelasticity.

4.9.6. Visco-Hypoelasticity

For visco-hypoelasticity model, the constitutive equations are formulated in terms of the rotated stress $R^T\sigma R$, here R is the rotation arising from the polar decomposition of the deformation gradient F. Let $R^T\sigma R = \Sigma + pI$ where Σ is the deviatoric part and p is the pressure. It is evident that $\Sigma = R^T S R$. The stress response function is given by:

$$\Sigma = \int_0^t 2\left[G_\infty + \sum_{i=1}^{n_G} G_i \exp\left(-\frac{t-\tau}{\tau_i^G}\right)\right](R^T dR)d\tau \tag{4.308}$$

$$p = \int_0^t \left[K_\infty + \sum_{i=1}^{n_K} K_i \exp\left(-\frac{t-\tau}{\tau_i^K}\right)\right]tr(D)d\tau \tag{4.309}$$

where:

d = deviatoric part of the rate of deformation tensor D.

This stress function is consistent with the generalized differential model in which the stress rate is replaced by Green-Naghdi rate.

To integrate the stress function, one perform the same integration scheme in Equation 4.300 (p. 148) to the rotated stress Equation 4.308 (p. 150) to yield:

$$(\Sigma_i)_{n+1} = \exp\left(-\frac{\Delta t}{\tau_i^G}\right)(\Sigma_i)_n + 2\exp\left(-\frac{\Delta t}{2\tau_i^G}\right)G_i R^T_{n+\frac{1}{2}}(d_{n+\frac{1}{2}})R_{n+\frac{1}{2}} \tag{4.310}$$

where:

$R_{n+\frac{1}{2}}$ = rotation tensor arising from the polar decomposition of the middle point deformation gradient $F_{n+\frac{1}{2}} = \frac{1}{2}(F_{n+1} + F_n)$

In the actual implementations, the rate of deformation tensor is replaced by the strain increment and we have

$$D_{n+\frac{1}{2}}\Delta t \approx \Delta\mu_{n+\frac{1}{2}} = \text{symm}(\nabla_{n+\frac{1}{2}}\Delta u) \tag{4.311}$$

where:

symm[.] = symmetric part of the tensor.

From $\Sigma = R^T s R$ and using Equation 4.310 (p. 150) and Equation 4.311 (p. 151), it follows that the deviatoric Cauchy stress is given by

$$(S_i)_{n+1} = \exp\left(-\frac{\Delta t}{\tau_i^G}\right)\Delta R(S_i)_n\Delta R^T + 2\exp\left(-\frac{\Delta t}{2\tau_i^G}\right)G_i\Delta R_{\frac{1}{2}}(\Delta e_{n+\frac{1}{2}})\Delta R^T_{n+\frac{1}{2}} \tag{4.312}$$

where:

$\Delta R = R_{n+1}R_n^T$

$\Delta R_{\frac{1}{2}} = R_{n+1}R^T_{n+\frac{1}{2}}$

$\Delta e_{n+\frac{1}{2}}$ = deviatoric part of $\Delta\varepsilon_{n+\frac{1}{2}}$

The pressure response can be integrated in a similar manner and the details are omitted.

4.9.7. Large-Strain Visco-Hyperelasticity

Large-strain visco-hyperelasticity is based on the formulation proposed by (Simo([327] (p. 939))), amended here to take into account the viscous volumetric response and the thermorheological simplicity. Simo's formulation is an extension of the small strain theory. Again, the viscoelastic behavior is specified separately by the underlying elasticity and relaxation behavior.

This decomposition of the energy function is consistent with hyperelasticity described in Hyperelasticity (p. 124).

$$\Phi(C) = \phi(\bar{C}) + U(J) \tag{4.313}$$

where:

J = det (F)

$\bar{C} = J^{\frac{2}{3}}C$ = isochoric part of the right Cauchy-Green deformation tensor C

As is well known, the constitutive equations for hyperelastic material with strain-energy function Φ is given by:

$$S^{2d} = 2\frac{\partial \Phi}{\partial C} \tag{4.314}$$

where:

S^{2d} = second Piola-Kirchhoff stress tensor

The true stress can be obtained as:

$$\sigma = \frac{1}{J} F S^{2d} F^T = \frac{2}{J} F \frac{\partial \Phi}{\partial C} F^T \tag{4.315}$$

Using Equation 4.313 (p. 151) in Equation 4.315 (p. 152) results

$$\sigma = \frac{2}{J} F \frac{\partial \varphi(\bar{C})}{\partial C} F^T + \frac{\partial U(J)}{\partial J} I \tag{4.316}$$

It has been shown elsewhere that $F \frac{\partial \varphi(\bar{C})}{\partial C} F^T$ is deviatoric, therefore Equation 4.316 (p. 152) already assumes the form of deviatoric/pressure decomposition.

Following Simo([327] (p. 939)) and Holzapfel([328] (p. 939)), the viscoelastic constitutive equations, in terms of the second Piola-Kirchhoff stress, is given by

$$\begin{aligned} S^{2d} = & \int_0^t \left[\alpha_\infty^G + \sum_{i=1}^{n_G} \alpha_i^G \exp\left(-\frac{t-\tau}{\tau_i^G} \right) \right] \left(2 \frac{d}{d\tau} \frac{d\Phi}{dC} \right) d\tau \\ & + \int_0^t \left[\alpha_\infty^K + \sum_{i=1}^{n_K} \alpha_i^K \exp\left(-\frac{t-\tau}{\tau_i^G} \right) \right] \left(2 \frac{d}{d\tau} \frac{dU}{dJ} \right) d\tau C^{-1} \end{aligned} \tag{4.317}$$

Denote

$$\bar{S}_i^{2d} = \int_0^t \left[\alpha_\infty^G + \sum_{i=1}^{n_G} \alpha_i^G \exp\left(-\frac{t-\tau}{\tau_i^G} \right) \right] \left(2 \frac{d}{d\tau} \frac{d\Phi}{dC} \right) d\tau \tag{4.318}$$

$$p_i = \int_0^t \left[\alpha_\infty^K + \sum_{i=1}^{n_K} \alpha_i^K \exp\left(-\frac{t-\tau}{\tau_i^G} \right) \right] \left(2 \frac{d}{d\tau} \frac{dU}{dJ} \right) d\tau C^{-1} \tag{4.319}$$

and applying the recursive formula to Equation 4.318 (p. 152) and Equation 4.319 (p. 152) yields,

$$(\bar{S}_i^{2d})_{n+1} = \exp\left(-\frac{\Delta t}{\tau_i^G} \right) (\bar{S}_i^{2d})_n + \alpha_i^G \exp\left(-\frac{\Delta t}{2\tau_i^G} \right) \left[\frac{d\Phi}{dC_{n+1}} - \frac{d\Phi}{dC_n} \right] \tag{4.320}$$

$$(p_i)_{n+1} = \exp\left(-\frac{\Delta t}{\tau_i^K} \right) (p_i)_n + \alpha_i^G \exp\left(-\frac{\Delta t}{2\tau_i^K} \right) \left[\frac{dU}{dJ_{n+1}} - \frac{dU}{dJ_n} \right] \tag{4.321}$$

The above are the updating formulas used in the implementation. Cauchy stress can be obtained using Equation 4.315 (p. 152).

4.9.8. Large-Strain Visco-Anisotropic Hyperelasticity

Large-strain visco-anisotropic hyperelasticity assumes that viscoelastic behavior is isotropic, while the underlying elasticity is anisotropic hyperelasticity. The algorithm is similar to that of viscoelastic hyperelasticity. The strain energy potential is now represented as follows:

$$\Phi(\mathbf{C}) = \Phi(\bar{\mathbf{C}}, \mathbf{A}\otimes\mathbf{A}, \mathbf{B}\otimes\mathbf{B}) + U(J) \tag{4.322}$$

where:

A and **B** are the two unit constitutive material directions, |A| = 1 and |B| = 1. Further information about the anisotropic hyperelastic potential is available in Anisotropic Hyperelasticity (p. 132).

4.9.9. Shift Functions

ANSYS offers the following forms of the shift function:

4.9.9.1. Williams-Landel-Ferry Shift Function
4.9.9.2. Tool-Narayanaswamy Shift Function
4.9.9.3. Tool-Narayanaswamy Shift Function with Fictive Temperature
4.9.9.4. User-Defined Shift Function

The shift function is activated via the **TB**,SHIFT command. For detailed information, see Viscoelasticity in the *Material Reference*.

4.9.9.1. Williams-Landel-Ferry Shift Function

The Williams-Landel-Ferry shift function (Williams [277] (p. 936)) is defined by

$$\log_{10}(A) = \frac{C_2(T - C_1)}{C_3 + T - C_1} \tag{4.323}$$

where:

T = temperature
C_1, C_2, C_3 = material parameters

4.9.9.2. Tool-Narayanaswamy Shift Function

The Tool-Narayanaswamy shift function (Narayanaswamy [110] (p. 926)) is defined by

$$A = \exp\left(\frac{H}{R}\left(\frac{1}{T_r} - \frac{1}{T}\right)\right) \tag{4.324}$$

where:

T_r = material parameter

$\frac{H}{R}$ = material parameter

4.9.9.3. Tool-Narayanaswamy Shift Function with Fictive Temperature

This extension of the Tool-Narayanaswamy shift function includes a fictive temperature. The shift function is defined by

$$A = \exp\left(\frac{H}{R}\left(\frac{1}{T_r} - \frac{X}{T} - \frac{1-X}{T_F}\right)\right) \tag{4.325}$$

where:

T_F = fictive temperature

$X \in [0,1]$ = material parameter

The fictive temperature is given by

$$T_F = \sum_{i=1}^{n_f} C_{fi} T_{fi}$$

where:

n_f = number of partial fictive temperatures

C_{fi} = fictive temperature relaxation coefficients

T_{fi} = partial fictive temperatures

An integrator for the partial fictive temperatures (Markovsky [108] (p. 926)) is given by

$$T_{fi} = \frac{\tau_{fi} T_{fi}^0 + T\Delta t A(T_F^0)}{\tau_{fi} + \Delta t A(T_F^0)}$$

where:

Δt = time increment

τ_{fi} = temperature relaxation times

The superscript 0 represents values from the previous time step.

The fictive temperature model also modifies the volumetric thermal strain model and gives the incremental thermal strain as

$$\Delta\varepsilon^T = \alpha_g(T)\Delta T + \left[\alpha_l(T_F) - \alpha_g(T_F)\right]\Delta T_F$$

where the glass and liquid coefficients of thermal expansion are given by

$$\alpha_g(T) = \alpha_{g0} + \alpha_{g1}T + \alpha_{g2}T^2 + \alpha_{g3}T^3 + \alpha_{g4}T^4$$

$$\alpha_l(T) = \alpha_{l0} + \alpha_{l1}T + \alpha_{l2}T^2 + \alpha_{l3}T^3 + \alpha_{l4}T^4$$

The total thermal strain is given by the sum over time of the incremental thermal strains

$$\varepsilon^T = \sum_t \Delta\varepsilon^T$$

4.9.9.4. User-Defined Shift Function

Other shift functions can be accommodated via the user-provided subroutine UsrShift, described in the *Guide to User-Programmable Features*. The inputs for this subroutine are the user-defined parameters,

the current value of time and temperature, their increments, and the current value of user state variables (if any). The outputs from the subroutine are $\Delta\xi$, $\Delta\xi_{1/2}$ as well as the current value of user state variables.

4.10. Concrete

The concrete material model predicts the failure of brittle materials. Both cracking and crushing failure modes are accounted for. **TB**,CONCR accesses this material model, which is available with the reinforced concrete element SOLID65.

The criterion for failure of concrete due to a multiaxial stress state can be expressed in the form (Willam and Warnke([37] (p. 922))):

$$\frac{F}{f_c} - S \geq 0 \tag{4.326}$$

where:

F = a function (to be discussed) of the principal stress state (σ_{xp}, σ_{yp}, σ_{zp})

S = failure surface (to be discussed) expressed in terms of principal stresses and five input parameters f_t, f_c, f_{cb}, f_1 and f_2 defined in Table 4.4: Concrete Material Table (p. 155)

f_c = uniaxial crushing strength

σ_{xp}, σ_{yp}, σ_{zp} = principal stresses in principal directions

If Equation 4.326 (p. 155) is satisfied, the material will crack or crush.

A total of five input strength parameters (each of which can be temperature dependent) are needed to define the failure surface as well as an ambient hydrostatic stress state. These are presented in Table 4.4: Concrete Material Table (p. 155).

Table 4.4: Concrete Material Table

(Input on TBDATA Commands with TB,CONCR)		
Label	**Description**	**Constant**
f_t	Ultimate uniaxial tensile strength	3
f_c	Ultimate uniaxial compressive strength	4
f_{cb}	Ultimate biaxial compressive strength	5
σ_h^a	Ambient hydrostatic stress state	6
f_1	Ultimate compressive strength for a state of biaxial compression superimposed on hydrostatic stress state σ_h^a	7
f_2	Ultimate compressive strength for a state of uniaxial compression superimposed on hydrostatic stress state σ_h^a	8

However, the failure surface can be specified with a minimum of two constants, f_t and f_c. The other three constants default to Willam and Warnke([37] (p. 922)):

$$f_{cb} = 1.2 f_c \tag{4.327}$$

$$f_1 = 1.45 f_c \tag{4.328}$$

$$f_2 = 1.725 f_c \tag{4.329}$$

However, these default values are valid only for stress states where the condition

$$|\sigma_h| \le \sqrt{3} f_c \tag{4.330}$$

$$\left(\sigma_h = \text{hydrostatic stress state} = \frac{1}{3}(\sigma_{xp} + \sigma_{yp} + \sigma_{zp}) \right) \tag{4.331}$$

is satisfied. Thus condition Equation 4.330 (p. 156) applies to stress situations with a low hydrostatic stress component. All five failure parameters should be specified when a large hydrostatic stress component is expected. If condition Equation 4.330 (p. 156) is not satisfied and the default values shown in Equation 4.327 (p. 155) thru Equation 4.329 (p. 156) are assumed, the strength of the concrete material may be incorrectly evaluated.

When the crushing capability is suppressed with f_c = -1.0, the material cracks whenever a principal stress component exceeds f_t.

Both the function F and the failure surface S are expressed in terms of principal stresses denoted as σ_1, σ_2, and σ_3 where:

$$\sigma_1 = \max(\sigma_{xp}, \sigma_{yp}, \sigma_{zp}) \tag{4.332}$$

$$\sigma_3 = \min(\sigma_{xp}, \sigma_{yp}, \sigma_{zp}) \tag{4.333}$$

and $\sigma_1 \ge \sigma_2 \ge \sigma_3$. The failure of concrete is categorized into four domains:

1. $0 \ge \sigma_1 \ge \sigma_2 \ge \sigma_3$ (compression - compression - compression)

2. $\sigma_1 \ge 0 \ge \sigma_2 \ge \sigma_3$ (tensile - compression - compression)

3. $\sigma_1 \ge \sigma_2 \ge 0 \ge \sigma_3$ (tensile - tensile - compression)

4. $\sigma_1 \ge \sigma_2 \ge \sigma_3 \ge 0$ (tensile - tensile - tensile)

In each domain, independent functions describe F and the failure surface S. The four functions describing the general function F are denoted as F_1, F_2, F_3, and F_4 while the functions describing S are denoted as S_1, S_2, S_3, and S_4. The functions S_i (i = 1,4) have the properties that the surface they describe is continuous while the surface gradients are not continuous when any one of the principal stresses changes sign. The surface will be shown in Figure 4.30: 3-D Failure Surface in Principal Stress Space (p. 158) and Figure 4.32: Failure Surface in Principal Stress Space with Nearly Biaxial Stress (p. 161). These functions are discussed in detail below for each domain.

4.10.1. The Domain (Compression - Compression - Compression)

$$0 \ge \sigma_1 \ge \sigma_2 \ge \sigma_3$$

In the compression - compression - compression regime, the failure criterion of Willam and Warnke([37] (p. 922)) is implemented. In this case, F takes the form

$$F = F_1 = \frac{1}{\sqrt{15}}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]^{\frac{1}{2}} \tag{4.334}$$

and S is defined as

$$S = S_1 = \frac{2r_2(r_2^2 - r_1^2)\cos\eta + r_2(2r_1 - r_2)\left[4(r_2^2 - r_1^2)\cos^2\eta + 5r_1^2 - 4r_1r_2\right]^{\frac{1}{2}}}{4(r_2^2 - r_1^2)\cos^2\eta + (r_2 - 2r_1)^2} \tag{4.335}$$

Terms used to define S are:

$$\cos\ \eta = \frac{2\sigma_1 - \sigma_2 - \sigma_3}{\sqrt{2}\ \left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]^{\frac{1}{2}}} \tag{4.336}$$

$$r_1 = a_0 + a_1\xi + a_2\xi^2 \tag{4.337}$$

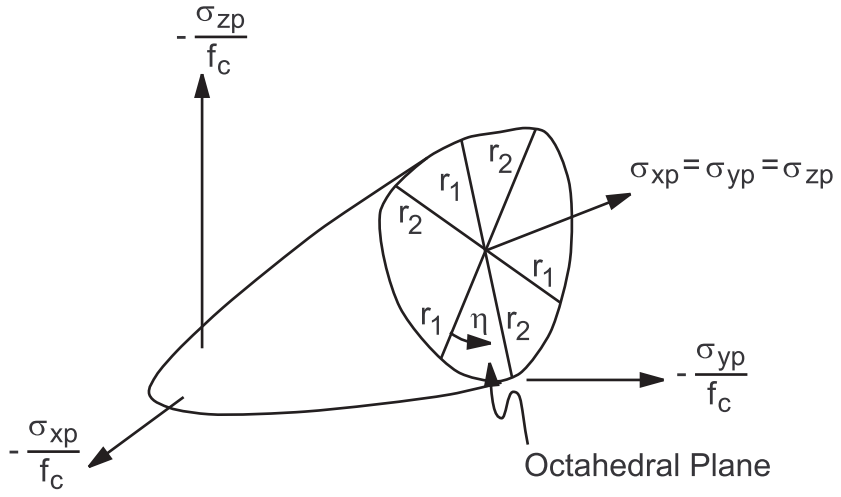
$$r_2 = b_0 + b_1\xi + b_2\xi^2 \tag{4.338}$$

$$\xi = \frac{\sigma_h}{f_c} \tag{4.339}$$

σ_h is defined by Equation 4.331 (p. 156) and the undetermined coefficients a_0, a_1, a_2, b_0, b_1, and b_2 are discussed below.

This failure surface is shown as Figure 4.30: 3-D Failure Surface in Principal Stress Space (p. 158). The angle of similarity η describes the relative magnitudes of the principal stresses. From Equation 4.336 (p. 157), $\eta = 0°$ refers to any stress state such that $\sigma_3 = \sigma_2 > \sigma_1$ (e.g. uniaxial compression, biaxial tension) while $\xi = 60°$ for any stress state where $\sigma_3 > \sigma_2 = \sigma_1$ (e.g. uniaxial tension, biaxial compression). All other multiaxial stress states have angles of similarity such that $0° \leq \eta \leq 60°$. When $\eta = 0°$, S_1 Equation 4.335 (p. 157) equals r_1 while if $\eta = 60°$, S_1 equals r_2. Therefore, the function r_1 represents the failure surface of all stress states with $\eta = 0°$. The functions r_1, r_2 and the angle η are depicted on Figure 4.30: 3-D Failure Surface in Principal Stress Space (p. 158).

Figure 4.30: 3-D Failure Surface in Principal Stress Space



It may be seen that the cross-section of the failure plane has cyclic symmetry about each 120° sector of the octahedral plane due to the range $0° < \eta < 60°$ of the angle of similitude. The function r_1 is determined by adjusting a_0, a_1, and a_2 such that f_t, f_{cb}, and f_1 all lie on the failure surface. The proper values for these coefficients are determined through solution of the simultaneous equations:

$$\begin{Bmatrix} \frac{F_1}{f_c}(\sigma_1 = f_t, \sigma_2 = \sigma_3 = 0) \\ \frac{F_1}{f_c}(\sigma_1 = 0, \sigma_2 = \sigma_3 = -f_{cb}) \\ \frac{F_1}{f_c}(\sigma_1 = -\sigma_h^a, \sigma_2 = \sigma_3 = -\sigma_h^a - f_1) \end{Bmatrix} = \begin{bmatrix} 1 & \xi_t & \xi_t^2 \\ 1 & \xi_{cb} & \xi_{cb}^2 \\ 1 & \xi_1 & \xi_1^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} \tag{4.340}$$

with

$$\xi_t = \frac{f_t}{3f_c},\ \xi_{cb} = -\frac{2f_{cb}}{3f_c},\ \xi_1 = -\frac{\sigma_h^a}{f_c} - \frac{2f_1}{3f_c} \tag{4.341}$$

The function r_2 is calculated by adjusting b_0, b_1, and b_2 to satisfy the conditions:

$$\begin{Bmatrix} \frac{F_1}{f_c}(\sigma_1 = \sigma_2 = 0, \sigma_3 = -f_c) \\ \frac{F_1}{f_c}(\sigma_1 = \sigma_2 = -\sigma_h^a, \sigma_3 = -\sigma_h^a - f_2) \\ \frac{F_1}{f_c} \qquad\qquad 0 \end{Bmatrix} = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{1}{9} \\ 1 & \xi_2 & \xi_2^2 \\ 1 & \xi_0 & \xi_0^2 \\ & & \end{bmatrix} \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \end{Bmatrix} \tag{4.342}$$

ξ_2 is defined by:

$$\xi_2 = -\frac{\sigma_h^a}{f_c} - \frac{f_2}{3f_c} \tag{4.343}$$

and ξ_0 is the positive root of the equation

$$r_2(\xi_0) = a_0 + a_1\xi_0 + a_2\xi_0^2 = 0 \tag{4.344}$$

where a_0, a_1, and a_2 are evaluated by Equation 4.340 (p. 158).

Since the failure surface must remain convex, the ratio r_1 / r_2 is restricted to the range

$$.5 < r_1/r_2 < 1.25 \tag{4.345}$$

although the upper bound is not considered to be restrictive since $r_1 / r_2 < 1$ for most materials (Willam([36] (p. 922))). Also, the coefficients a_0, a_1, a_2, b_0, b_1, and b_2 must satisfy the conditions (Willam and Warnke([37] (p. 922))):

$$a_0 > 0,\ a_1 \le 0,\ a_2 \le 0 \tag{4.346}$$

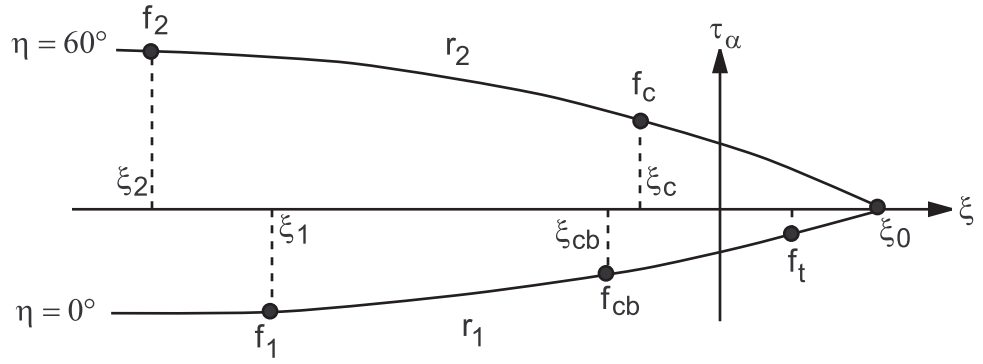
$$b_0 > 0,\ b_1 \le 0,\ b_2 \le 0 \tag{4.347}$$

Therefore, the failure surface is closed and predicts failure under high hydrostatic pressure ($\xi > \xi_2$). This closure of the failure surface has not been verified experimentally and it has been suggested that a von Mises type cylinder is a more valid failure surface for large compressive σ_h values (Willam([36] (p. 922))).

Consequently, it is recommended that values of f_1 and f_2 are selected at a hydrostatic stress level (σ_h^a) in the vicinity of or above the expected maximum hydrostatic stress encountered in the structure.

Equation 4.344 (p. 159) expresses the condition that the failure surface has an apex at $\xi = \xi_0$. A profile of r_1 and r_2 as a function of ξ is shown in Figure 4.31: A Profile of the Failure Surface (p. 159).

Figure 4.31: A Profile of the Failure Surface



As a Function of ξ_α

The lower curve represents all stress states such that $\eta = 0°$ while the upper curve represents stress states such that $\eta = 60°$. If the failure criterion is satisfied, the material is assumed to crush.

4.10.2. The Domain (Tension - Compression - Compression)

$$\sigma_1 \ge 0 \ge \sigma_2 \ge \sigma_3$$

In the regime, F takes the form

$$F = F_2 = \frac{1}{\sqrt{15}}\left[(\sigma_2 - \sigma_3)^2 + \sigma_2^2 + \sigma_3^2\right]^{\frac{1}{2}} \tag{4.348}$$

and S is defined as

$$S = S_2 = \left(1 - \frac{\sigma_1}{f_t}\right)\frac{2p_2(p_2^2 - p_1^2)\cos\eta + p_2(2p_1 - p_2)\left[4(p_2^2 - p_1^2)\cos^2\eta + 5p_1^2 - 4p_1p_2\right]^{\frac{1}{2}}}{4(p_2^2 - p_1^2)\cos^2\eta + (p_2 - 2p_1)^2} \tag{4.349}$$

where cos η is defined by Equation 4.336 (p. 157) and

$$p_1 = a_0 + a_1\chi + a_2\chi^2 \tag{4.350}$$

$$p_2 = b_0 + b_1\chi + b_2\chi^2 \tag{4.351}$$

The coefficients a_0, a_1, a_2, b_0, b_1, b_2 are defined by Equation 4.340 (p. 158) and Equation 4.342 (p. 158) while

$$\chi = \frac{(\sigma_2 + \sigma_3)}{3f_c} \tag{4.352}$$

If the failure criterion is satisfied, cracking occurs in the plane perpendicular to principal stress σ_1.

This domain can also crush. See (Willam and Warnke([37] (p. 922))) for details.

4.10.3. The Domain (Tension - Tension - Compression)

$$\sigma_1 \geq \sigma_2 \geq 0 \geq \sigma_3$$

In the tension - tension - compression regime, F takes the form

$$F = F_3 = \sigma_i;\ i = 1,2 \tag{4.353}$$

and S is defined as

$$S = S_3 = \frac{f_t}{f_c}\left(1 + \frac{\sigma_3}{f_c}\right);\ i = 1,2 \tag{4.354}$$

If the failure criterion for both i = 1, 2 is satisfied, cracking occurs in the planes perpendicular to principal stresses σ_1 and σ_2. If the failure criterion is satisfied only for i = 1, cracking occurs only in the plane perpendicular to principal stress σ_1.

This domain can also crush. See (Willam and Warnke([37] (p. 922))) for details.

4.10.4. The Domain (Tension - Tension - Tension)

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$$

In the tension - tension - tension regimes, F takes the form

$$F = F_4 = \sigma_i;\ i = 1,2,3 \tag{4.355}$$

and S is defined as

$$S = S_4 = \frac{f_t}{f_c} \tag{4.356}$$

If the failure criterion is satisfied in directions 1, 2, and 3, cracking occurs in the planes perpendicular to principal stresses σ_1, σ_2, and σ_3.

If the failure criterion is satisfied in directions 1 and 2, cracking occurs in the plane perpendicular to principal stresses σ_1 and σ_2.

If the failure criterion is satisfied only in direction 1, cracking occurs in the plane perpendicular to principal stress σ_1.

Figure 4.32: Failure Surface in Principal Stress Space with Nearly Biaxial Stress

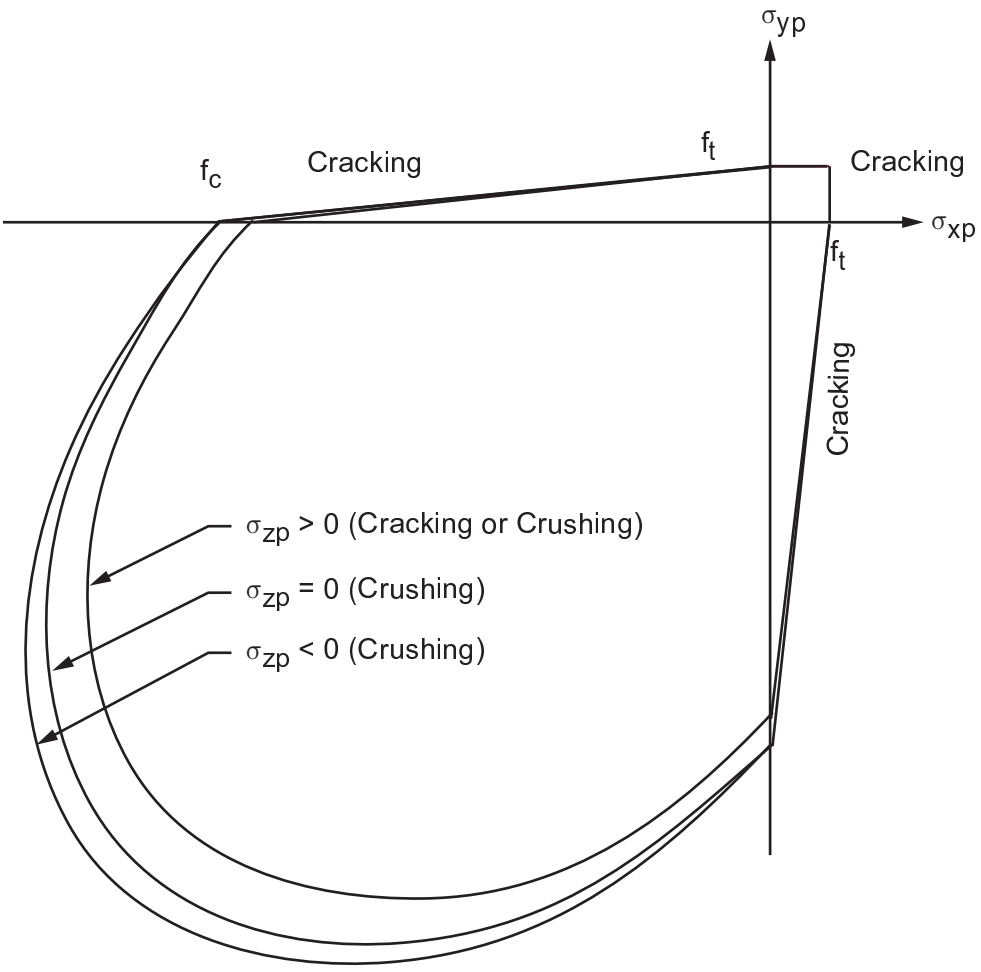


Figure 4.32: Failure Surface in Principal Stress Space with Nearly Biaxial Stress (p. 161) represents the 3-D failure surface for states of stress that are biaxial or nearly biaxial. If the most significant nonzero principal stresses are in the σ_{xp} and σ_{yp} directions, the three surfaces presented are for σ_{zp} slightly greater than zero, σ_{zp} equal to zero, and σ_{zp} slightly less than zero. Although the three surfaces, shown as projections on the σ_{xp} - σ_{yp} plane, are nearly equivalent and the 3-D failure surface is continuous, the mode of material failure is a function of the sign of σ_{zp}. For example, if σ_{xp} and σ_{yp} are both negative and σ_{zp} is slightly positive, cracking would be predicted in a direction perpendicular to the σ_{zp} direction. However, if σ_{zp} is zero or slightly negative, the material is assumed to crush.

4.11. Swelling

The ANSYS program provides a capability of irradiation induced swelling (accessed with **TB**,SWELL). Swelling is defined as a material enlarging volumetrically in the presence of neutron flux. The amount of swelling may also be a function of temperature. The material is assumed to be isotropic and the basic solution technique used is the initial stress method. Swelling calculations are available only through the user swelling subroutine. See User Routines and Non-Standard Uses of the *Advanced Analysis Guide* and the *Guide to ANSYS User Programmable Features* for more details. Input must have C_{72} set to 10. Constants C_{67} through C_{71} are used together with fluence and temperature, as well as possibly strain, stress and time, to develop an expression for swelling rate.

Any of the following three conditions cause the swelling calculations to be bypassed:

1. If $C_{67} \leq 0.$ and $C_{68} \leq 0.$

2. If (input temperature + T_{off}) U $\leq$ 0, where T_{off} = offset temperature (input on **TOFFST** command).

3. If $Fluence_n \leq Fluence_{n-1}$ (n refers to current time step).

The total swelling strain is computed in subroutine USERSW as:

$$\varepsilon_n^{sw} = \varepsilon_{n-1}^{sw} + \Delta\varepsilon^{sw} \tag{4.357}$$

where:

ε_n^{sw} = swelling strain at end of substep n
$\Delta\varepsilon^{sw} = r\Delta f$ = swelling strain increment
r = swelling rate
$\Delta f = f_n - f_{n-1}$ = change of fluence
f_n = fluence at end of substep n (input as VAL1, etc. on the **BFE**,,FLUE command)

For a solid element, the swelling strain vector is simply:

$$\{\varepsilon^{sw}\} = \left\lfloor \varepsilon_n^{sw} \quad \varepsilon_n^{sw} \quad \varepsilon_n^{sw} \quad 0 \quad 0 \quad 0 \right\rfloor^T \tag{4.358}$$

It is seen that the swelling strains are handled in a manner totally analogous to temperature strains in an isotropic medium and that shearing strains are not used.

4.12. Cohesive Zone Material (CZM) Model

Fracture or delamination along an interface between phases plays a major role in limiting the toughness and the ductility of the multi-phase materials, such as matrix-matrix composites and laminated composite structure. This has motivated considerable research on the failure of the interfaces. Interface delamination can be modeled by traditional fracture mechanics methods such as the nodal release technique. Alternatively, you can use techniques that directly introduce fracture mechanism by adopting softening relationships between tractions and the separations, which in turn introduce a critical fracture energy that is also the energy required to break apart the interface surfaces. This technique is called the cohesive zone material (CZM) model . The interface surfaces of the materials can be represented by a special set of interface elements or contact elements, and a CZM model can be used to characterize the constitutive behavior of the interface.

The CZM model consists of a constitutive relation between the traction **T** acting on the interface and the corresponding interfacial separation δ (displacement jump across the interface). The definitions of traction and separation depend on the element and the material model.

The following related topics are available:
4.12.1. Interface Elements
4.12.2. Contact Elements

4.12.1. Interface Elements

For interface elements, the interfacial separation is defined as the displacement jump, δ (that is, the difference of the displacements of the adjacent interface surfaces):

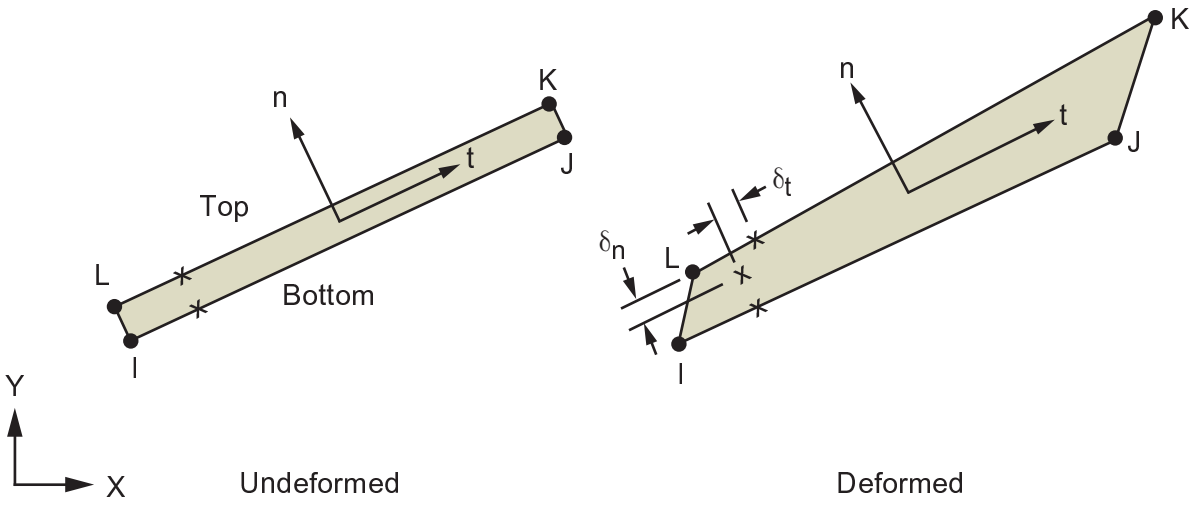
$$\boldsymbol{\delta} = \mathbf{u}^{\text{TOP}} - \mathbf{u}^{\text{BOTTOM}} = \text{interfacial separation} \tag{4.359}$$

The definition of the separation is based on local element coordinate system, Figure 4.33: Schematic of Interface Elements (p. 163). The normal of the interface is denoted as local direction **n**, and the local tangent direction is denoted as **t**. Thus:

$$\delta_n = \mathbf{n} \cdot \boldsymbol{\delta} = \text{normal separation} \tag{4.360}$$

$$\delta_t = \mathbf{t} \cdot \boldsymbol{\delta} = \text{tangential (shear) separation} \tag{4.361}$$

Figure 4.33: Schematic of Interface Elements



The following related topics are available:
4.12.1.1. Material Model - Exponential Behavior
4.12.1.2. Material Model - Bilinear Behavior
4.12.1.3. Viscous Regularization

4.12.1.1. Material Model - Exponential Behavior

An exponential form of the CZM model (input via **TB**,CZM), originally proposed by Xu and Needleman([363] (p. 941)), uses a surface potential:

$$\phi(\boldsymbol{\delta}) = e\sigma_{max}\bar{\delta}_n[1-(1+\Delta_n)e^{-\Delta_n}e^{-\Delta_t^2}] \tag{4.362}$$

where:

$\varphi(\delta)$ = surface potential
e = 2.7182818
σ_{max} = maximum normal traction at the interface (input on **TBDATA** command as C1 using **TB**,CZM)
$\bar{\delta}_n$ = normal separation across the interface where the maximum normal traction is attained with $\delta_t = 0$ (input on **TBDATA** command as C2 using **TB**,CZM)

$\bar{\delta}_t$ = shear separation where the maximum shear traction is attained at $\delta_t = \frac{\sqrt{2}}{2}\bar{\delta}_t$ (input on **TBDATA** command as C3 using **TB**,CZM)

$$\Delta_n = \frac{\delta_n}{\bar{\delta}_n}$$

$$\Delta_t = \frac{\delta_t}{\bar{\delta}_t}$$

The traction is defined as:

$$\mathbf{T} = \frac{\partial \phi(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}} \tag{4.363}$$

or

$$T_n = \frac{\partial \phi(\boldsymbol{\delta})}{\partial \delta_n} \tag{4.364}$$

and

$$T_t = \frac{\partial \phi(\boldsymbol{\delta})}{\partial \delta_t} \tag{4.365}$$

From equations Equation 4.364 (p. 164) and Equation 4.365 (p. 164), we obtain the normal traction of the interface

$$T_n = e\sigma_{max}\Delta_n e^{-\Delta_n} e^{-\Delta_t^2} \tag{4.366}$$

and the shear traction

$$T_t = 2e\sigma_{max}\frac{\bar{\delta}_n}{\bar{\delta}_t}\Delta_t(1+\Delta_n)e^{-\Delta_n}e^{-\Delta_t^2} \tag{4.367}$$

The normal work of separation is:

$$\phi_n = e\sigma_{max}\bar{\delta}_n \tag{4.368}$$

and shear work of separation is assumed to be the same as the normal work of separation, φ_n, and is defined as:

$$\phi_t = \sqrt{e/2}\,\tau_{max}\bar{\delta}_t \tag{4.369}$$

For the 3-D stress state, the shear or tangential separations and the tractions have two components, δ_{t1} and δ_{t2} in the element's tangential plane, and we have:

$$\delta_t = \sqrt{\delta_{t_1}^2 + \delta_{t_2}^2} \tag{4.370}$$

The traction is then defined as:

$$T_{t_1} = \frac{\partial \phi(\boldsymbol{\delta})}{\partial \delta_{t_1}} \tag{4.371}$$

and

$$T_{t_2} = \frac{\partial \phi(\boldsymbol{\delta})}{\partial \delta_{t_2}} \tag{4.372}$$

(In POST1 and POST26 the traction, **T**, is output as SS and the separation, δ, is output as SD.)

The tangential direction t_1 is defined along ij edge of element and the direction t_2 is defined along direction perpendicular to the plane formed by n and t_1. Directions t_1, t_2, and n follow the righthand side rule.

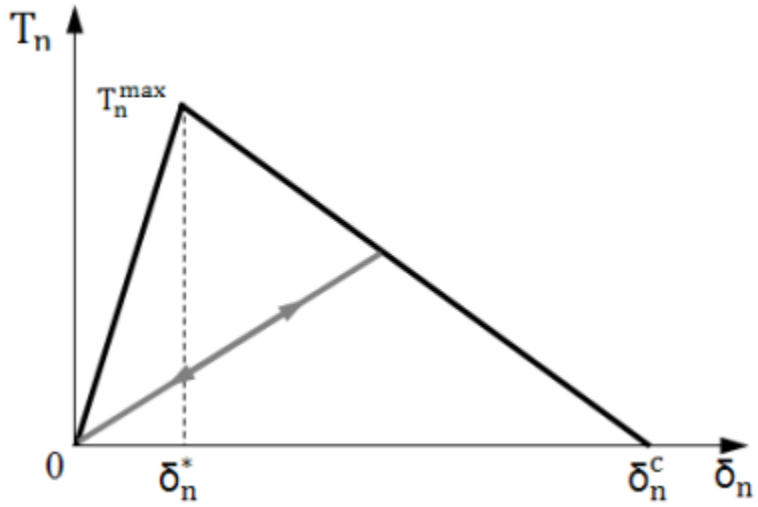
4.12.1.2. Material Model - Bilinear Behavior

The bilinear CZM model (input via **TB**,CZM with *TBOPT* = BILI) can be used with interface elements. The model is based on the model proposed by Alfano and Crisfield [365] (p. 941).

Mode I Dominated Bilinear CZM Model

The Mode I dominated bilinear CZM model assumes that the separation of the material interfaces is dominated by the displacement jump normal to the interface, as shown in the following figure:

Figure 4.34: Mode I Dominated Bilinear CZM Law



The relation between normal cohesive traction T_n and normal displacement jump δ_n can be expressed as:

$$T_n = K_n \delta_n (1 - D_n) \tag{4.373}$$

Where:

K_n = normal cohesive stiffness $\frac{T_n^{max}}{\delta_n^*}$

T_n^{max} = maximum normal cohesive traction σ_{max}
(input via TBDATA command as C1 using TB,CZM)

δ_n^* = normal displacement jump at maximum normal cohesive traction

δ_n^c = normal displacement jump at the completion of debonding
(input via TBDATA command as C2 using TB,CZM)

α = ratio of δ_n^* to δ_n^c (input via TBDATA command as C5 using TB,CZM)

δ_n^{max} = maximum normal displacement jump attained in deformation history
$\max \delta_n(\tau'),$ where $0 \le \tau' \le \tau$

D_n = damage parameter associated with Mode I dominated bilinear cohesive law, defined as

$$D_n = \begin{cases} 0 & \delta_n^{max} \le \delta_n^* \\ \left(\frac{\delta_n^{max} - \delta_n^*}{\delta_n^{max}}\right)\left(\frac{\delta_n^c}{\delta_n^c - \delta_n^*}\right) & \delta_n^* < \delta_n^{max} \le \delta_n^c \\ 1 & \delta_n^{max} > \delta_n^c \end{cases}$$

For Mode I dominated cohesive law, the tangential cohesive traction and tangential displacement jump behavior is assumed to follow the normal cohesive traction and normal displacement jump behavior and is expressed as:

$$T_t = K_t \delta_t (1 - D_n) \tag{4.374}$$

Were:

K_t = tangential cohesive stiffness $\frac{T_t^{max}}{\delta_t^*}$

T_t^{max} = maximum tangential cohesive traction τ_{max}
(input via TBDATA command as C3 using TB,CZM, where
C3 = $-\tau_{max}$ to indicate Mode I dominated fracture)

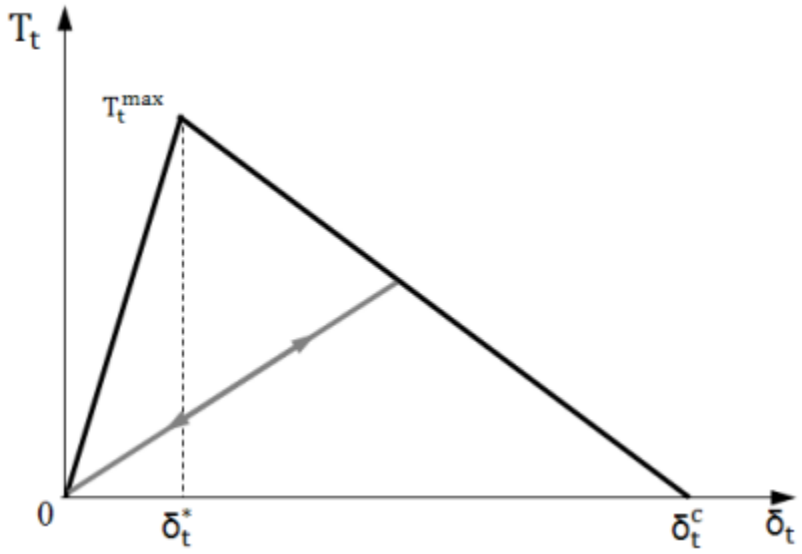
δ_t^* = tangential displacement jump at maximum tangential cohesive traction

δ_t^c = tangential displacement jump at the completion of debonding
(input via TBDATA command as C4 using TB,CZM)

Mode II Dominated Bilinear CZM Model

The Mode II dominated bilinear CZM model assumes that the separation of the material interfaces is dominated by the displacement jump that is tangent to the interface, as shown in the following figure:

Figure 4.35: Mode II Dominated Bilinear CZM Law



The relation between tangential cohesive traction T_t and tangential displacement jump δ_t can be expressed as:

$$T_t = K_t \delta_t (1 - D_t) \tag{4.375}$$

Where:

K_t = tangential cohesive stiffness $\dfrac{T_t^{max}}{\delta_t^*}$

T_t^{max} = maximum tangential cohesive traction τ_{max}
(input via TBDATA command as C3 using TB,CZM)

δ_t^* = tangential displacement jump at maximum tangential cohesive traction

δ_n^c = tangential displacement jump at the completion of debonding
(input via TBDATA command as C4 using TB,CZM)

α = ratio of δ_t^* to δ_t^c (input via TBDATA command as C5 using TB,CZM)

δ_t^{max} = maximum tangential displacement jump attained in deformation history
$\max \delta_t(\tau'),$ where $0 \le \tau' \le \tau$

D_t = damage parameter associated with Mode II/III dominated bilinear cohesive law, defined as

$$D_t = \begin{cases} 0 & \delta_t^{max} \le \delta_t^* \\ \left(\dfrac{\delta_t^{max} - \delta_t^*}{\delta_t^{max}}\right)\left(\dfrac{\delta_t^c}{\delta_t^c - \delta_t^*}\right) & \delta_t^* < \delta_t^{max} \le \delta_t^c \\ 1 & \delta_t^{max} > \delta_t^c \end{cases}$$

For Mode II dominated cohesive law, the normal cohesive traction and normal displacement jump behavior is assumed to follow the tangential cohesive traction and tangential displacement jump behavior and is expressed as:

$$T_n = K_n \delta_n (1 - D_t) \tag{4.376}$$

Where:

K_n = normal cohesive stiffness $\frac{T_n^{max}}{\delta_n^*}$

T_n^{max} = maximum normal cohesive traction σ_{max}
(input via TBDATA command as C1 using TB,CZM, where
C1 = $-\sigma_{max}$ to indicate Mode II/III dominated fracture)

δ_n^* = normal displacement jump at maximum normal cohesive traction

δ_t^c = normal displacement jump at the completion of debonding
(input via TBDATA command as C2 using TB,CZM)

Mixed-Mode Bilinear Cohesive Zone Material Model

For bilinear cohesive law under the mixed-mode fracture, the separation of material interfaces depends on both the normal and tangential components of displacement jumps. To take into account the difference in their contributions to the separation of material interfaces, a non-dimensional effective displacement jump λ for mixed-mode fracture is defined as

$$\lambda = \sqrt{\left(\frac{\delta_n}{\delta_n^c}\right)^2 + \beta^2 \left(\frac{\delta_t}{\delta_t^c}\right)^2} \tag{4.377}$$

where the non-dimensional parameter β (input via the **TBDATA** command as *C6* using **TB**,CZM) assigns different weights to the tangential and normal displacement jumps.

The normal and tangential components of the cohesive tractions are expressed as:

$$\begin{aligned} T_n &= K_n \delta_n (1 - D_m) \\ T_t &= K_t \delta_t (1 - Dm) \end{aligned} \tag{4.378}$$

The damage parameter D_m associated with mixed mode bilinear cohesive law is defined as:

$$D_m = \begin{cases} 0 & \lambda^{max} \le \lambda_{cr} \\ \min(1, d_m) & \lambda^{max} > \lambda_{cr} \end{cases}$$

Where:

$\lambda_{cr} = \frac{\delta_n^*}{\delta_n^c} = \beta \frac{\delta_t^*}{\delta_t^c}$, the value of λ at which the effective traction is maximum

$$d_m = \eta \left(\frac{\lambda^{max} - \lambda_{cr}}{\lambda^{max}} \right)$$

$$\eta = \frac{\delta_n^c}{\delta_n^c - \delta_n^*} = \frac{\delta_t^c}{\delta_t^c - \delta_t^*}$$

$\lambda^{max} = \max \lambda(\tau')$, where $0 \le \tau' \le \tau$

Fracture Mode Identification of a CZM Model

Determining the fracture mode of a CZM model is based on the input data, as follows:

Case	Input on the TBDATA command as follows:
Mode I Dominated	*C1*, *C2*, *C3*, *C4*, *C5* (where *C3* = $-\tau_{max}$)
Mode II/III Dominated	*C1*, *C2*, *C3*, *C4*, *C5* (where *C1* = $-\sigma_{max}$)
Mixed-Mode	*C1*, *C2*, *C3*, *C4*, *C5*, *C6* (where *C1* = σ_{max} and *C3* = τ_{max})

4.12.1.3. Viscous Regularization

To avoid convergence difficulties, a small fictitious viscosity is introduced in cohesive zone material (CZM) models ([416] (p. 944)). The traction definitions are modified by adding viscous damping, as follows:

$$T_n^* = T_n + \zeta \frac{d\Delta_n}{dt}$$
$$T_t^* = T_t + \zeta \frac{d\Delta_t}{dt} \quad (4.379)$$

where:

T_n = normal traction before viscous regularization

T_t = tangential traction before viscous regularization

ζ = damping coefficient (input on the **TBDATA** command as C1 (after defining the CZM model with the viscous regularization option [**TB**,CZM,,,,VREG])

4.12.2. Contact Elements

Delamination with contact elements is referred to as debonding. The interfacial separation is defined in terms of contact gap or penetration and tangential slip distance. The computation of contact and tangential slip is based on the type of contact element and the location of contact detection point. The cohesive zone model can only be used for bonded contact (KEYOPT(12) = 2, 3, 4, 5, or 6) with the augmented Lagrangian method (KEYOPT(2) = 0) or the pure penalty method (KEYOPT(2) = 1). See CONTA174 - 3-D 8-Node Surface-to-Surface Contact (p. 554) for details.

4.12.2.1. Material Model - Bilinear Behavior

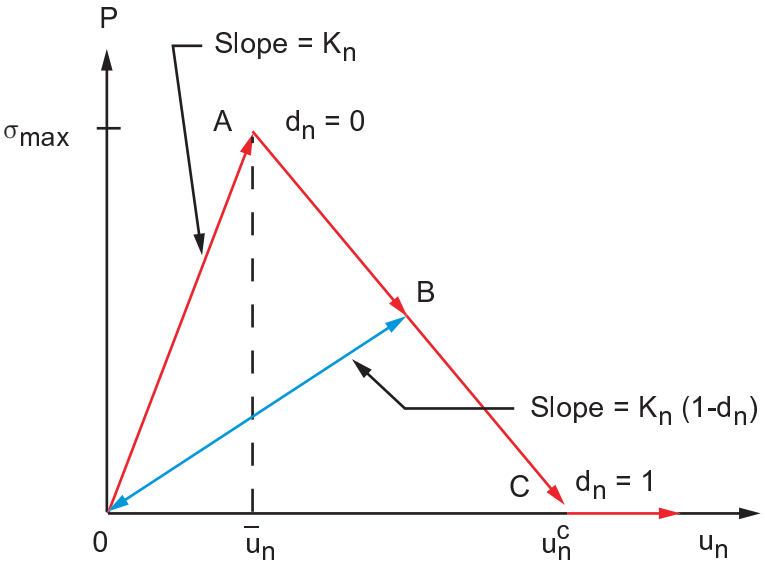
The bilinear cohesive zone material model (input using **TB**,CZM) is based on the model proposed by Alfano and Crisfield [365] (p. 941).

Mode I Debonding

Mode I debonding defines a mode of separation of the interface surfaces where the separation normal to the interface dominates the slip tangent to the interface. The normal contact stress (tension) and contact gap behavior is plotted in Figure 4.36: Normal Contact Stress and Contact Gap Curve for Bilinear Cohesive Zone Material (p. 170). It shows linear elastic loading (*OA*) followed by linear softening (*AC*). The maximum normal contact stress is achieved at point *A*. Debonding begins at point *A* and is completed at point *C* when the normal contact stress reaches zero value; any further separation occurs without any normal contact stress. The area under the curve *OAC* is the energy released due to debonding and is called the critical fracture energy. The slope of the line *OA* determines the contact gap at the maximum

normal contact stress and, hence, characterizes how the normal contact stress decreases with the contact gap, i.e., whether the fracture is brittle or ductile. After debonding has been initiated it is assumed to be cumulative and any unloading and subsequent reloading occurs in a linear elastic manner along line *OB* at a more gradual slope.

Figure 4.36: Normal Contact Stress and Contact Gap Curve for Bilinear Cohesive Zone Material



The equation for curve *OAC* can be written as:

$$P = K_n u_n (1 - d_n) \tag{4.380}$$

where:

P = normal contact stress (tension)
K_n = normal contact stiffness
u_n = contact gap
$\bar{u}_n$ = contact gap at the maximum normal contact stress (tension)
u_n^c = contact gap at the completion of debonding (input on **TBDATA** command as C2 using **TB**,CZM)
d_n = debonding parameter

The debonding parameter for Mode I Debonding is defined as:

$$d_n = \left(\frac{u_n - \bar{u}_n}{u_n} \right) \left(\frac{u_n^c}{u_n^c - \bar{u}_n} \right) \tag{4.381}$$

with $d_n = 0$ for $\Delta_n \leq 1$ and $0 < d_n \leq 1$ for $\Delta_n > 1$.

where:

$$\Delta_n = \frac{u_n}{\bar{u}_n}$$

The normal critical fracture energy is computed as:

$$G_{cn} = \frac{1}{2}\sigma_{max}u_n^c \tag{4.382}$$

where:

σ_{max} = maximum normal contact stress (input on **TBDATA** command as *C1* using **TB**,CZM).

For mode I debonding the tangential contact stress and tangential slip behavior follows the normal contact stress and contact gap behavior and is written as:

$$\tau_t = K_t u_t (1 - d_n) \tag{4.383}$$

where:

τ_t = tangential contact stress
K_t = tangential contact stiffness
u_t = tangential slip distance

Mode II Debonding

Mode II debonding defines a mode of separation of the interface surfaces where tangential slip dominates the separation normal to the interface. The equation for the tangential contact stress and tangential slip distance behavior is written as:

$$\tau_t = K_t u_t (1 - d_t) \tag{4.384}$$

where:

$\bar{u}_t$ = tangential slip distance at the maximum tangential contact stress

u_t^c = tangential slip distance at the completion of debonding (input on **TBDATA** command as C4 using **TB**,CZM)
d_t = debonding parameter

The debonding parameter for Mode II Debonding is defined as:

$$d_t = \left(\frac{u_t - \bar{u}_t}{u_t}\right)\left(\frac{u_t^c}{u_t^c - \bar{u}_t}\right) \tag{4.385}$$

with $d_t = 0$ for $\Delta_t \leq 1$ and $0 < d_t \leq 1$ for $\Delta_t > 1$.

where:

$$\Delta_t = \frac{u_t}{\bar{u}_t}$$

For the 3-D stress state an "isotropic" behavior is assumed and the debonding parameter is computed using an equivalent tangential slip distance:

$$u_t = \sqrt{u_1^2 + u_2^2} \tag{4.386}$$

where:

u_1 and u_2 = slip distances in the two principal directions in the tangent plane

The components of the tangential contact stress are defined as:

$$\tau_1 = K_t u_1 (1 - d_t) \tag{4.387}$$

and

$$\tau_2 = K_t u_2 (1 - d_t) \tag{4.388}$$

The tangential critical fracture energy is computed as:

$$G_{ct} = \frac{1}{2} \tau_{max} u_t^c \tag{4.389}$$

where:

τ_{max} = maximum tangential contact stress (input on **TBDATA** command as C3 using **TB**,CZM).

The normal contact stress and contact gap behavior follows the tangential contact stress and tangential slip behavior and is written as:

$$P = K_n u_n (1 - d_t) \tag{4.390}$$

Mixed-Mode Debonding

In mixed-mode debonding the interface separation depends on both normal and tangential components. The equations for the normal and the tangential contact stresses are written as:

$$P = K_n u_n (1 - d_m) \tag{4.391}$$

and

$$\tau_t = K_t u_t (1 - d_m) \tag{4.392}$$

The debonding parameter is defined as:

$$d_m = \left(\frac{\Delta_m - 1}{\Delta_m} \right) \chi \tag{4.393}$$

with $d_m = 0$ for $\Delta_m \leq 1$ and $0 < d_m \leq 1$ for $\Delta_m > 1$, and Δ_m and χ are defined below.

where:

$\Delta_m = \sqrt{\Delta_n^2 + \Delta_t^2}$ and

$$\chi = \left(\frac{u_n^c}{u_n^c - \bar{u}_n} \right) = \left(\frac{u_t^c}{u_t^c - \bar{u}_t} \right)$$

The constraint on χ that the ratio of the contact gap distances be the same as the ratio of tangential slip distances is enforced automatically by appropriately scaling the contact stiffness value, K_t, as follows:

$$K_t = \left(\frac{\tau_{max} u_n^c}{\sigma_{max} u_t^c} \right) K_n \tag{4.394}$$

For mixed-mode debonding, both normal and tangential contact stresses contribute to the total fracture energy and debonding is completed before the critical fracture energy values are reached for the components. Therefore, a power law based energy criterion is used to define the completion of debonding:

$$\left(\frac{G_n}{G_{cn}}\right)^2+\left(\frac{G_t}{G_{ct}}\right)^2=1 \tag{4.395}$$

where:

$G_n=\int P du_n$ and

$G_t=\int \tau_t du_t$

are, respectively, the normal and tangential fracture energies. Verification of satisfaction of energy criterion can be done during postprocessing of results.

Identifying Debonding Modes

The debonding modes are based on input data:

1. Mode I for normal data (input on **TBDATA** command as *C1*, *C2*, and *C5*).
2. Mode II for tangential data (input on **TBDATA** command as *C3*, *C4*, and *C5*).
3. Mixed mode for normal and tangential data (input on **TBDATA** command as *C1*, *C2*, *C3*, *C4*, *C5* and *C6*).

Artificial Damping

Debonding is accompanied by convergence difficulties in the Newton-Raphson solution. Artificial damping is used in the numerical solution to overcome these problems. For mode I debonding the normal contact stress expression would appear as:

$$P=P^{final}+(P^{initial}-P^{final})e^{\frac{-t}{\eta}} \tag{4.396}$$

where:

$t=t^{final}-t^{initial}=$ time interval

η = damping coefficient (input on **TBDATA** command as *C5* using **TB**,CZM).

The damping coefficient has units of time, and it should be smaller than the minimum time step size so that the maximum traction and maximum separation (or critical fracture energy) values are not exceeded in debonding calculations.

Tangential Slip Under Normal Compression

An option is provided to control tangential slip under compressive normal contact stress for mixed-mode debonding. By default, no tangential slip is allowed for this case, but it can be activated by setting the flag β (input on **TBDATA** command as *C6* using **TB**,CZM) to 1. Settings on β are:

β = 0 (default) no tangential slip under compressive normal contact stress for mixed-mode debonding

β = 1 tangential slip under compressive normal contact stress for mixed-mode debonding

Post Separation Behavior

After debonding is completed the surface interaction is governed by standard contact constraints for normal and tangential directions. Frictional contact is used if friction is specified for contact elements.

Results Output for POST1 and POST26

All applicable output quantities for contact elements are also available for debonding: normal contact stress P (output as PRES), tangential contact stress τ_t (output as SFRIC) or its components τ_1 and τ_2 (output as TAUR and TAUS), contact gap u_n (output as GAP), tangential slip u_t (output as SLIDE) or its components u_1 and u_2 (output as TASR and TASS), etc. Additionally, debonding specific output quantities are also available (output as NMISC data): debonding time history (output as DTSTART), debonding parameter d_n , d_t or d_m (output as DPARAM), fracture energies G_n and G_t (output as DENERI and DENERII).

4.13. Fluid Material Models

Fluid material models define constitutive relations for fluids used in structural analyses. These models are available for hydrostatic fluid elements, HSFLD241 and HSFLD242.

For more details on the fluid material models presented here, see Fluids in the *Material Reference*.

4.13.1. Liquid

Liquid material can be used to model a compressible linear (Newtonian) isotropic elastic non-viscous fluid. Using the stress-strain relationship for linear elastic material as given in Structural Fundamentals (p. 5), the constitutive equation for liquid can be written as:

$$\Delta P = P - P_0 = -K\varepsilon_{volf} \tag{4.397}$$

with

$$\varepsilon_{volf} = \varepsilon_{ijf}\delta_{ij} = \frac{\Delta V_f}{\Delta V_{0f}} \tag{4.398}$$

$$P = -\frac{1}{3}\sigma_{ij}\delta_{ij} \tag{4.399}$$

where:

K = bulk modulus
ε_{volf} = volumetric strain for fluid
ΔV_f = change in fluid volume
V_{0f} = initial fluid volume
P = current fluid pressure
P_0 = initial fluid pressure (use the **IC** command)

The thermal properties of liquid material are defined by coefficient of thermal expansion, which determines the change of fluid volume due to change of temperature:

$$\Delta V_f^{th} = \varepsilon_{volf}^{th} V_{0f} = 3V_{0f}\alpha(T - T_{ref}) \tag{4.400}$$

where:

α = coefficient of thermal expansion
T = current fluid temperature
T_{ref} = reference temperature (use **TREF** or **MP** command)

Initial fluid density (needed when prescribing fluid mass flow rate) can be specified using the **TB**command with *Lab* = FLUID and *TB0PT* = LIQUID.

Change in fluid volume also affects fluid density. If fluid mass is kept fixed, current density for liquid material can defined as:

$$\rho_f = \rho_{0f}\frac{V_{0f}}{V_f} = \rho_{0f}\frac{1}{\left[1+3\alpha(T-T_{tref})-\frac{\Delta P}{K}\right]} \tag{4.401}$$

where:

ρ_{0f} = initial fluid density

This material model is defined through the**TB** command with *Lab* = FLUID and *TB0PT* = LIQUID; in addition, use the **TBDATA** command to define the bulk modulus, K, coefficient of thermal expansion, α, and initial fluid density, ρ_{0f}, as material constants C1, C2, and C3, respectively.

4.13.2. Gas

The Ideal Gas Law is written as:

$$\rho = \frac{P}{RT} \Rightarrow \frac{\partial\rho}{\partial P} = \frac{1}{RT} \tag{4.402}$$

where:

ρ = density
P = pressure
R = universal gas constant
T = temperature

The Ideal Gas Law is used to define the constitutive equation for gas material:

$$V_f = \frac{mRT_t}{P_t} \tag{4.403}$$

where:

m = mass of fluid
R = universal gas constant
$T_t = T_{off} + T$ = total fluid temperature
T_{off} = temperature offset from absolute zero to zero (use **TOFFST** command)
$P_t = P_{ref} + P$ = total fluid pressure
P_{ref} = reference fluid pressure (specified as a real constant PREF)

Current density for gas material is defined in terms of initial density as:

$$\rho_f = \rho_{0f}\frac{P_t T_{0t}}{P_{0t} T_t} \tag{4.404}$$

where:

$T_{0t} = T_{off} + T_{ref}$ = initial total temperature
T_{ref} = reference temperature (use **TREF** or **MP** command)
$P_{0t} = P_{ref} + P_0$ = initial total fluid pressure

To define a gas material, use the**TB** command with *Lab* = FLUID, *TB0PT* = GAS, and initial fluid density, ρ_{0f}, as material constant C1 on the **TBDATA** command. To completely define the initial state of the gas material, also specify a reference pressure, P_{ref}, as a real constant (use the **R** command) for the hydrostatic fluid element, a temperature offset from absolute zero to zero temperature, T_{off}, (use the **TOFFST** command), and a reference temperature, T_{ref} (use the **TREF** or **MP** command).

4.13.3. Pressure-Volume Data

For compressible fluids that do not follow the constitutive equation for liquid or Ideal Gas Law, a pressure-volume curve may be used to define a pressure-volume relationship.

Current fluid density for this material model can be defined as:

$$\rho_f = \rho_{0f} \frac{V_{0f}}{V_f} \tag{4.405}$$

where the current fluid volume is given by the pressure-volume curves based on the current fluid pressure and temperature.

A pressure-volume curve can be defined through the **TB** command with *Lab* = FLUID and *TB0PT* = PVDATA, along with the **TBTEMP** command to define temperature, and the **TBPT** command to define pressure-volume data points for each temperature. The pressure-volume data points must be defined in terms of total pressure and total volume of the fluid in the containing vessel.

Chapter 5: Electromagnetics

The following topics concerning electromagnetic are available:

5.1. Electromagnetic Field Fundamentals

Electromagnetic fields are governed by the following Maxwell's equations (Smythe([150] (p. 929))):

$$\nabla x\{H\} = \{J\} + \left\{\frac{\partial D}{\partial t}\right\} = \{J_s\} + \{J_e\} + \{J_v\} + \left\{\frac{\partial D}{\partial t}\right\} \tag{5.1}$$

$$\nabla x\{E\} = -\left\{\frac{\partial B}{\partial t}\right\} \tag{5.2}$$

$$\nabla \cdot \{B\} = 0 \tag{5.3}$$

$$\nabla \cdot \{D\} = \rho \tag{5.4}$$

where:

∇ x = curl operator

$\nabla \cdot$ = divergence operator

{H} = magnetic field intensity vector

{J} = total current density vector

$\{J_s\}$ = applied source current density vector

$\{J_e\}$ = induced eddy current density vector

$\{J_{vs}\}$ = velocity current density vector

{D} = electric flux density vector (Maxwell referred to this as the displacement vector, but to avoid misunderstanding with mechanical displacement, the name electric flux density is used here.)

t = time

{E} = electric field intensity vector

{B} = magnetic flux density vector

ρ = electric charge density

The continuity equation follows from taking the divergence of both sides of Equation 5.1 (p. 177).

$$\nabla \cdot \left[\{J\} + \left\{ \frac{\partial D}{\partial t} \right\} \right] = 0 \tag{5.5}$$

The continuity equation must be satisfied for the proper setting of Maxwell's equations. Users should prescribe J_s taking this into account.

The above field equations are supplemented by the constitutive relation that describes the behavior of electromagnetic materials. For problems considering saturable material without permanent magnets, the constitutive relation for the magnetic fields is:

$$\{B\} = [\mu]\{H\} \tag{5.6}$$

where:

μ = magnetic permeability matrix, in general a function of {H}

The magnetic permeability matrix [μ] may be input either as a function of temperature or field. Specifically, if [μ] is only a function of temperature,

$$[\mu] = \mu_o \begin{bmatrix} \mu_{rx} & 0 & 0 \\ 0 & \mu_{ry} & 0 \\ 0 & 0 & \mu_{rz} \end{bmatrix} \tag{5.7}$$

where:

μ_o = permeability of free space (input on **EMUNIT** command)
μ_{rx} = relative permeability in the x-direction (input as MURX on **MP** command)

If [μ] is only a function of field,

$$[\mu] = \mu_h \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.8}$$

where:

μ_h = permeability derived from the input B versus H curve (input with **TB**,BH).

Mixed usage is also permitted, e.g.:

$$[\mu] = \begin{bmatrix} \mu_h & 0 & 0 \\ 0 & \mu_o\mu_{ry} & 0 \\ 0 & 0 & \mu_h \end{bmatrix} \tag{5.9}$$

When permanent magnets are considered, the constitutive relation becomes:

$$\{B\} = [\mu]\{H\} + \mu_o\{M_o\} \tag{5.10}$$

where:

$\{M_o\}$ = remanent intrinsic magnetization vector

Rewriting the general constitutive equation in terms of reluctivity it becomes:

$$\{H\} = [\nu]\{B\} - \frac{1}{\nu_o}[\nu]\{M_o\} \tag{5.11}$$

where:

$[\nu]$ = reluctivity matrix = [μ]-1

ν_o = reluctivity of free space $= \frac{1}{\mu_o}$

The constitutive relations for the related electric fields are:

$$\{J\} = [\sigma][\{E\} + \{v\} \times \{B\}] \tag{5.12}$$

$$\{D\} = [\varepsilon]\{E\} \tag{5.13}$$

where:

$$[\sigma] = \begin{bmatrix} \sigma_{xx} & 0 & 0 \\ 0 & \sigma_{yy} & 0 \\ 0 & 0 & \sigma_{zz} \end{bmatrix} = \text{electrical conductivity matrix}$$

$$[\varepsilon] = \begin{bmatrix} \varepsilon_{xx} & 0 & 0 \\ 0 & \varepsilon_{yy} & 0 \\ 0 & 0 & \varepsilon_{zz} \end{bmatrix} = \text{permittivity matrix}$$

$$\{v\} = \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix} = \text{velocity vector}$$

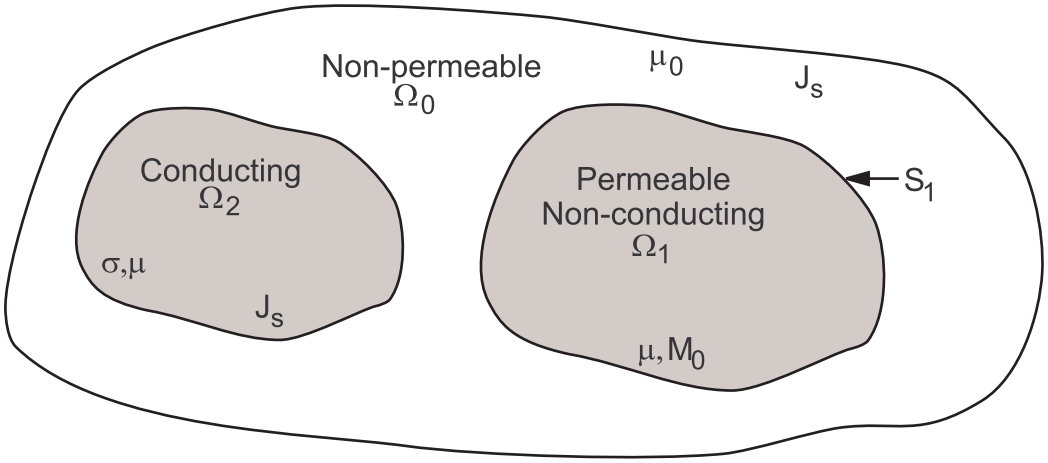
σ_{xx} = conductivity in the x-direction (input as inverse of RSVX on **MP** command)

ε_{xx} = permittivity in the x-direction (input as PERX on **MP** command)

The solution of magnetic field problems is commonly obtained using potential functions. Two kinds of potential functions, the magnetic vector potential and the magnetic scalar potential are used depending on the problem to be solved. Factors affecting the choice of potential include: field dynamics, field dimensionality, source current configuration, domain size and discretization.

The applicable regions are shown below. These will be referred to with each solution procedure discussed below.

Figure 5.1: Electromagnetic Field Regions



where:

Ω_0 = free space region
Ω_1 = nonconducting permeable region
Ω_2 = conducting region
μ = permeability of iron
μ_o = permeability of air
M_o = permanent magnets
S_1 = boundary of W1
σ = conductivity
$\Omega = \Omega_1 + \Omega_2 + \Omega_0$

5.1.1. Magnetic Scalar Potential

The scalar potential method as implemented in SOLID5, SOLID96, and SOLID98 for 3-D magnetostatic fields is discussed in this section. Magnetostatics means that time varying effects are ignored. This reduces Maxwell's equations for magnetic fields to:

$$\nabla x\{H\} = \{J_s\} \tag{5.14}$$

$$\nabla \cdot \{B\} = 0 \tag{5.15}$$

5.1.2. Solution Strategies

In the domain Ω_0 and Ω_1 of a magnetostatic field problem (Ω_2 is not considered for magnetostatics) a solution is sought which satisfies the relevant Maxwell's Equation 5.14 (p. 180) and Equation 5.15 (p. 180) and the constitutive relation Equation 5.10 (p. 178) in the following form (Gyimesi([141] (p. 928)) and Gyimesi([149] (p. 929))):

$$\{H\} = \{H_g\} - \nabla\phi_g \tag{5.16}$$

$$\nabla \cdot [\mu]\nabla\phi_g - \nabla \cdot [\mu]\{H_g\} - \nabla \cdot \mu_o\{M_o\} = \{0\} \tag{5.17}$$

where:

$\{H_g\}$ = preliminary or "guess" magnetic field
ϕ_g = generalized potential

The development of $\{H_g\}$ varies depending on the problem and the formulation. Basically, $\{H_g\}$ must satisfy Ampere's law (Equation 5.14 (p. 180)) so that the remaining part of the field can be derived as the gradient of the generalized scalar potential ϕ_g. This ensures that ϕ_g is singly valued. Additionally, the absolute value of $\{H_g\}$ must be greater than that of $\Delta\phi_g$. In other words, {Hg} should be a good approximation of the total field. This avoids difficulties with cancellation errors (Gyimesi([149] (p. 929))).

This framework allows for a variety of scalar potential formulation to be used. The appropriate formulation depends on the characteristics of the problem to be solved. The process of obtaining a final solution may involve several steps (controlled by the **MAGOPT** solution option).

As mentioned above, the selection of $\{H_g\}$ is essential to the development of any of the following scalar potential strategies. The development of $\{H_g\}$ always involves the Biot-Savart field $\{H_s\}$ which satisfies Ampere's law and is a function of source current $\{J_s\}$. $\{H_s\}$ is obtained by evaluating the integral:

$$\{H_s\} = \frac{1}{4\pi}\int_{volc} \frac{\{J_s\}\times\{r\}}{|\{r\}|^3}\, d(volc) \tag{5.18}$$

where:

$\{J_s\}$ = current source density vector at d(volc)
{r} = position vector from current source to node point
volc = volume of current source

The above volume integral can be reduced to the following surface integral (Gyimesi et al.([173] (p. 930)))

$$\{H_s\} = \frac{1}{4\pi}\int_{surfc} \frac{\{J_s\}}{|\{r\}|}\times d(surfc) \tag{5.19}$$

where:

surfc = surface of the current source

Evaluation of this integral is automatically performed upon initial solution execution or explicitly (controlled by the **BIOT** command). The values of $\{J_s\}$ are obtained either directly as input by:

SOURC36 - Current Source

or indirectly calculated by electric field calculation using:

SOLID5 - 3-D Coupled-Field Solid
LINK68 - Coupled Thermal-Electric Line
SOLID98 - Tetrahedral Coupled-Field Solid
PLANE230 - 2-D Electric Solid
SOLID231 or SOLID232 - 3-D Electric Solids

Depending upon the current configuration, the integral given in Equation 5.19 (p. 181) is evaluated in a closed form and/or a numerical fashion (Smythe([150] (p. 929))).

Three different solution strategies emerge from the general framework discussed above:

Reduced Scalar Potential (RSP) Strategy
Difference Scalar Potential (DSP) Strategy
General Scalar Potential (GSP) Strategy

5.1.2.1. RSP Strategy

Applicability

If there are no current sources ($\{J_s\} = 0$) the RSP strategy is applicable. Also, in general, if there are current sources and there is no iron ($[\mu] = [\mu_o]$) within the problem domain, the RSP strategy is also applicable. This formulation is developed by Zienkiewicz([75] (p. 924)).

Procedure

The RSP strategy uses a one-step procedure (**MAGOPT**,0). Equation 5.16 (p. 180) and Equation 5.17 (p. 180) are solved making the following substitution:

$$\{H_g\} = \{H_s\} \text{ in } \Omega_o \text{ and } \Omega_1 \tag{5.20}$$

Saturation is considered if the magnetic material is nonlinear. Permanent magnets are also considered.

5.1.2.2. DSP Strategy

Applicability

The DSP strategy is applicable when current sources and singly connected iron regions exist within the problem domain ($\{J_s\} \neq \{0\}$) and ($[\mu] \neq [\mu_o]$). A singly connected iron region does not enclose a current. In other words a contour integral of {H} through the iron must approach zero as $u \rightarrow \infty$.

$$\oint \{H\} \cdot \{d\ell\} \rightarrow \{0\} \text{ in } \Omega_1 \text{ as } u \rightarrow \infty \tag{5.21}$$

This formulation is developed by Mayergoyz([119] (p. 927)).

Procedure

The DSP strategy uses a two-step solution procedure. The first step (**MAGOPT**,2) makes the following substitution into Equation 5.16 (p. 180) and Equation 5.17 (p. 180):

$$\{H_g\} = \{H_s\} \text{ in } \Omega_o \text{ and } \Omega_1 \tag{5.22}$$

subject to:

$$\{n\} \times \{H_g\} = \{0\} \text{ on } S_1 \tag{5.23}$$

This boundary condition is satisfied by using a very large value of permeability in the iron (internally set by the program). Saturation and permanent magnets are not considered. This step produces a near zero field in the iron region which is subsequently taken to be zero according to:

$$\{H_1\} = \{0\} \text{ in } \Omega_1 \tag{5.24}$$

and in the air region:

$$\{H_o\} = \{H_s\} - \nabla\phi_g \text{ in } \Omega_o \tag{5.25}$$

The second step (**MAGOPT**,3) uses the fields calculated on the first step as the preliminary field for Equation 5.16 (p. 180) and Equation 5.17 (p. 180):

$$\{H_g\} = \{0\} \text{ in } \Omega_1 \tag{5.26}$$

$$\{H_g\} = \{H_o\} \text{ in } \Omega_o \tag{5.27}$$

Here saturation and permanent magnets are considered. This step produces the following fields:

$$\{H_1\} = -\nabla\phi_g \text{ in } \Omega_1 \tag{5.28}$$

and

$$\{H_o\} = \{H_g\} - \nabla\phi_g \text{ in } \Omega_o \tag{5.29}$$

which are the final results to the applicable problems.

5.1.2.3. GSP Strategy

Applicability

The GSP strategy is applicable when current sources ($\{J_s\} \neq \{0\}$) in conjunction with a multiply connected iron ($[\mu] \neq [\mu_o]$) region exist within the problem domain. A multiply connected iron region encloses some current source. This means that a contour integral of {H} through the iron region is not zero:

$$\oint \{H\} \cdot \{d\ell\} \rightarrow \{0\} \text{ in } \Omega_1 \tag{5.30}$$

where:

- $\cdot$ = refers to the dot product

This formulation is developed by Gyimesi([141] (p. 928), [149] (p. 929), [201] (p. 932)).

Procedure

The GSP strategy uses a three-step solution procedure. The first step (**MAGOPT**,1) performs a solution only in the iron with the following substitution into Equation 5.16 (p. 180) and Equation 5.17 (p. 180):

$$\{H_g\} = \{H_s\} \text{ in } \Omega_o \tag{5.31}$$

subject to:

$$\{n\} \cdot [\mu](\{H_g\} - \nabla\phi_g) = 0 \text{ on } S_1 \tag{5.32}$$

Here S_1 is the surface of the iron air interface. Saturation can optimally be considered for an improved approximation of the generalized field but permanent magnets are not. The resulting field is:

$$\{H_1\} = \{H_s\} - \nabla\phi_g \tag{5.33}$$

The second step (**MAGOPT**,2) performs a solution only in the air with the following substitution into Equation 5.16 (p. 180) and Equation 5.17 (p. 180):

$$\{H_g\} = \{H_s\} \text{ in } \Omega_o \tag{5.34}$$

subject to:

$$\{n\} \times \{H_g\} = \{n\} \times \{H_1\} \text{ in } S_1 \tag{5.35}$$

This boundary condition is satisfied by automatically constraining the potential solution ϕ_g at the surface of the iron to be what it was on the first step (**MAGOPT**,1). This step produces the following field:

$$\{H_o\} = \{H_s\} - \nabla\phi_g \text{ in } \Omega_o \tag{5.36}$$

Saturation or permanent magnets are of no consequence since this step obtains a solution only in air.

The third step (**MAGOPT**,3) uses the fields calculated on the first two steps as the preliminary field for Equation 5.16 (p. 180) and Equation 5.17 (p. 180):

$$\{H_g\} = \{H_1\} \text{ in } \Omega_1 \tag{5.37}$$

$$\{H_g\} = \{H_o\} \text{ in } \Omega_o \tag{5.38}$$

Here saturation and permanent magnets are considered. The final step allows for the total field to be computed throughout the domain as:

$$\{H\} = \{H_g\} - \nabla\phi_g \text{ in } \Omega \tag{5.39}$$

5.1.3. Magnetic Vector Potential

The vector potential method is implemented in PLANE13, PLANE53, SOLID97, and PLANE233 for both 2-D and 3-D electromagnetic fields is discussed in this section. Considering static and dynamic fields and neglecting displacement currents (quasi-stationary limit), the following subset of Maxwell's equations apply:

$$\nabla \times \{H\} = \{J\} \tag{5.40}$$

$$\nabla \times \{E\} = -\frac{\partial B}{\partial t} \tag{5.41}$$

$$\nabla \cdot \{B\} = 0 \tag{5.42}$$

The usual constitutive equations for magnetic and electric fields apply as described by Equation 5.11 (p. 179) and Equation 5.12 (p. 179). Although some restriction on anisotropy and nonlinearity do occur in the formulations mentioned below.

In the entire domain, Ω, of an electromagnetic field problem a solution is sought which satisfies the relevant Maxwell's Equation 5.40 (p. 184) thru Equation 5.41 (p. 184). See Figure 5.1: Electromagnetic Field Regions (p. 180) for a representation of the problem domain Ω.

A solution can be obtained by introducing potentials which allow the magnetic field {B} and the electric field {E} to be expressed as (Biro([120] (p. 927))):

$$\{B\} = \nabla \times \{A\} \tag{5.43}$$

$$\{E\} = -\left\{\frac{\partial A}{\partial t}\right\} - \nabla V \tag{5.44}$$

where:

{A} = magnetic vector potential
V = electric scalar potential

These specifications ensure the satisfaction of two of Maxwell's equations, Equation 5.41 (p. 184) and Equation 5.42 (p. 184). What remains to be solved is Ampere's law, Equation 5.40 (p. 184) in conjunction with the constitutive relations, Equation 5.11 (p. 179), and the divergence free property of current density. Additionally, to ensure uniqueness of the vector potential, the Coulomb gauge condition is employed. The resulting differential equations are:

$$\nabla \times [\nu]\nabla \times \{A\} - \nabla \nu_e \nabla \cdot \{A\} + [\sigma]\left\{\frac{\partial A}{\partial t}\right\} + [\sigma]\nabla V - \{v\} \times [\sigma]\nabla \times \{A\} = \{0\} \text{ in } \Omega_2 \tag{5.45}$$

$$\nabla \cdot \left([\sigma]\left\{\frac{\partial A}{\partial t}\right\} - [\sigma]\nabla V + \{v\} \times [\sigma]\nabla \times \{A\} \right) = \{0\} \text{ in } \Omega_2 \tag{5.46}$$

$$\nabla \times [\nu]\nabla \times \{A\} - \nabla \nu_e \nabla \cdot \{A\} = \{J_s\} + \nabla \times \frac{1}{\nu_o}[\nu]\{M_o\} \text{ in } \Omega_o + \Omega_1 \tag{5.47}$$

where:

$$\nu_e = \frac{1}{3}\text{tr}[\nu] = \frac{1}{3}(\nu(1,1) + \nu(2,2) + \nu(3,3))$$

These equations are subject to the appropriate boundary conditions.

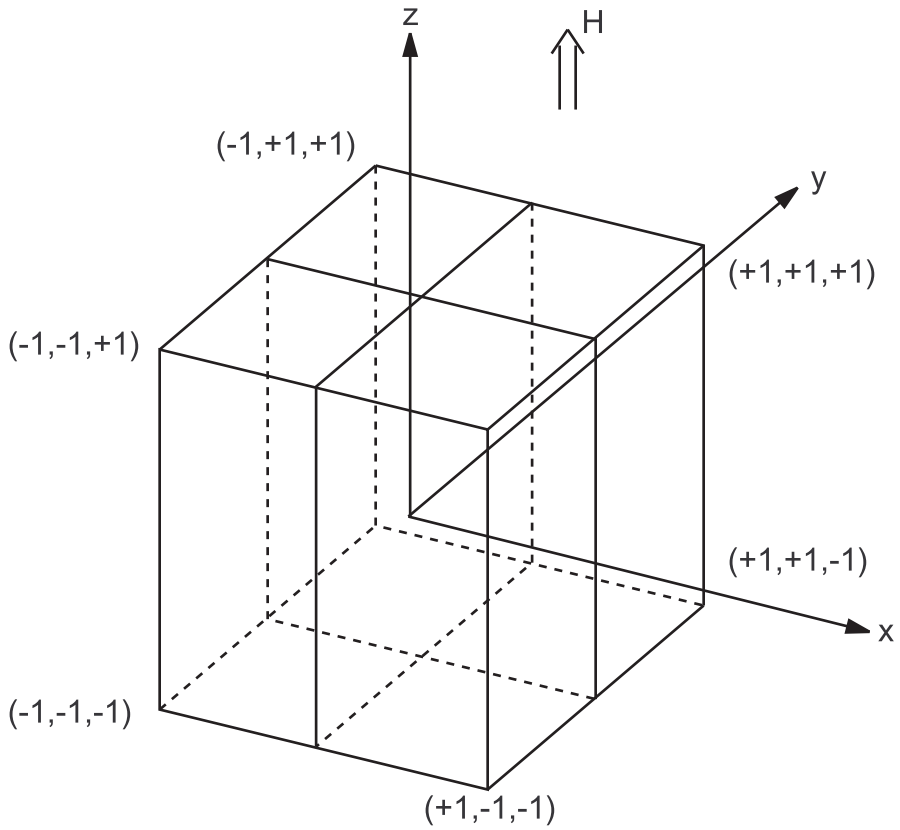
This system of simplified Maxwell's equations with the introduction of potential functions has been used for the solutions of 2-D and 3-D, static and dynamic fields. Silvester([72] (p. 924)) presents a 2-D static formulation and Demerdash([151] (p. 929)) develops the 3-D static formulation. Chari([69] (p. 924)), Brauer([70] (p. 924)) and Tandon([71] (p. 924)) discuss the 2-D eddy current problem and Weiss([94] (p. 925)) and Garg([95] (p. 925)) discuss 2-D eddy current problems which allow for skin effects (eddy currents present in the source conductor). The development of 3-D eddy current problems is found in Biro([120] (p. 927)).

5.1.4. Limitation of the Node-Based Vector Potential

For models containing materials with different permeabilities, the 3-D vector potential formulation is not recommended. The solution has been found (Biro et al. [200.] and Preis et al. [203.]) to be incorrect when the normal component of the vector potential is significant at the interface between elements of different permeability. The shortcomings of the node-based continuous vector potential formulation is demonstrated below.

Consider a volume bounded by planes, x = ± -1, y = ± 1, and z = ± 1. See Figure 5.2: Patch Test Geometry (p. 186). Subdivide the volume into four elements by planes, x = 0 and y = 0. The element numbers are set according to the space quadrant they occupy. The permeability, μ, of the elements is μ_1, μ_2, μ_3, and μ_4, respectively. Denote unit vectors by $\{1_x\}$, $\{1_y\}$, and $\{1_z\}$. Consider a patch test with a known field, $\{H_k\} = \{1_z\}$, $\{B_k\} = \mu\{H_k\}$ changes in the volume according to μ.

Figure 5.2: Patch Test Geometry



Since $\{B_k\}$ is constant within the elements, one would expect that even a first order element could pass the patch test. This is really the case with edge element but not with nodal elements. For example, $\{A\} = \mu \times \{1_y\}$ provides a perfect edge solution but not a nodal one because the normal component of A in not continuous.

The underlying reason is that the partials of a continuous {A} do not exist; not even in a piece-wise manner. To prove this statement, assume that they exist. Denote the partials at the origin by:

$$A_x^+ = \frac{\partial}{\partial y} A_x \text{ for } y > 0;\ A_{\bar{x}} = \frac{\partial}{\partial y} A_x \text{ for } y < 0;$$
$$A_y^+ = \frac{\partial}{\partial x} A_y \text{ for } x > 0;\ A_{\bar{y}} = \frac{\partial}{\partial x} A_y \text{ for } x < 0; \quad (5.48)$$

Note that there are only four independent partials because of A continuity. The following equations follow from $B_k = curl\ A$.

$$A_y^+ - A_x^+ = \mu_1;\ A_{\bar{y}} - A_x^+ = \mu_2$$
$$A_{\bar{y}} - A_{\bar{x}} = \mu_3;\ A_y^+ - A_{\bar{x}} = \mu_4 \quad (5.49)$$

Since the equation system, (Equation 5.49 (p. 186)) is singular, a solution does not exist for arbitrary μ. This contradiction concludes the proof.

5.1.5. Edge-Based Magnetic Vector Potential

The inaccuracy associated with the node-based formulation is eliminated by using the edge-based elements with a discontinuous normal component of magnetic vector potential. The edge-based method is implemented in the 3-D electromagnetic SOLID236 and SOLID237 elements.

The differential equations governing SOLID236 and SOLID237 elements are the following:

$$\nabla \times [\nu] \nabla \times \{A\} + [\sigma]\left(\left\{\frac{\partial A}{\partial t}\right\} + \nabla V\right) + [\varepsilon]\left(\left\{\frac{\partial^2 A}{\partial t^2}\right\} + \nabla\left\{\frac{\partial V}{\partial t}\right\}\right) = 0 \text{ in } \Omega_2 \tag{5.50}$$

$$\nabla \cdot \left([\sigma]\left(\left\{\frac{\partial A}{\partial t}\right\} + \nabla V\right) + [\varepsilon]\left(\left\{\frac{\partial^2 A}{\partial t^2}\right\} + \nabla\left\{\frac{\partial V}{\partial t}\right\}\right)\right) = 0 \text{ in } \Omega_2 \tag{5.51}$$

$$\nabla \times [\nu] \nabla \times \{A\} = \{J_s\} + \nabla \times \frac{1}{\nu_0}[\nu]\{M_0\} \text{ in } \Omega_0 + \Omega_1 \tag{5.52}$$

These equations are subject to the appropriate magnetic and electrical boundary conditions.

The uniqueness of edge-based magnetic vector potential is ensured by the tree gauging procedure (**GAUGE** command) that sets the edge-flux degrees of freedom corresponding to the spanning tree of the finite element mesh to zero.

5.1.6. Harmonic Analysis Using Complex Formalism

In a general dynamic problem, any field quantity, q(r,t) depends on the space, r, and time, t, variables. In a harmonic analysis, the time dependence can be described by periodic functions:

$$q(r,t) = a(r)\cos(\omega t + \phi(r)) \tag{5.53}$$

or

$$q(r,t) = c(r)\cos(\omega t) - s(r)\sin(\omega t) \tag{5.54}$$

where:

r = location vector in space
t = time
ω = angular frequency of time change.
a(r) = amplitude (peak)
ϕ(r) = phase angle
c(r) = measurable field at ωt = 0 degrees
s(r) = measurable field at ωt = -90 degrees

In an electromagnetic analysis, q(r,t) can be the flux density, {B}, the magnetic field, {H}, the electric field, {E}, the current density, J, the vector potential, {A}, or the scalar potential, V. Note, however, that q(r,t) can not be the Joule heat, Q^j, the magnetic energy, W, or the force, F^{jb}, because they include a time-constant term.

The quantities in Equation 5.53 (p. 187) and Equation 5.54 (p. 187) are related by

$$c(r) = a(r)\cos(\phi(r)) \tag{5.55}$$

$$s(r) = a(r)\sin(\phi(r)) \tag{5.56}$$

$$a^2(r) = c^2(r) + s^2(r) \tag{5.57}$$

$$\tan(\phi(r)) = s(r)/c(r) \tag{5.58}$$

In Equation 5.53 (p. 187)) a(r), ϕ(r), c(r) and s(r) depend on space coordinates but not on time. This separation of space and time is taken advantage of to minimize the computational cost. The originally 4 (3 space + 1 time) dimensional real problem can be reduced to a 3 (space) dimensional complex problem. This can be achieved by the complex formalism.

The measurable quantity, q(r,t), is described as the real part of a complex function:

$$q(r,t) = \mathrm{Re}\{Q(r)\exp(j\omega t)\} \tag{5.59}$$

Q(r) is defined as:

$$Q(r) = Q_r(r) + jQ_i(r) \tag{5.60}$$

where:

j = imaginary unit
Re { } = denotes real part of a complex quantity
Q_r(r) and Q_i(r) = real and imaginary parts of Q(r). Note that Q depends only on the space coordinates.

The complex exponential in Equation 5.59 (p. 188) can be expressed by sine and cosine as

$$\exp(j\omega t) = \cos(\omega t) + j\sin(\omega t) \tag{5.61}$$

Substituting Equation 5.61 (p. 188) into Equation 5.59 (p. 188) provides Equation 5.60 (p. 188)

$$q(r,t) = Q_r(r)\cos(\omega t) - Q_i(r)\sin(\omega t) \tag{5.62}$$

Comparing Equation 5.53 (p. 187) with Equation 5.62 (p. 188) reveals:

$$c(r) = Q_r(r) \tag{5.63}$$

$$s(r) = Q_i(r) \tag{5.64}$$

In words, the complex real, Q_r(r), and imaginary, Q_i(r), parts are the same as the measurable cosine, c(r), and sine, s(r), amplitudes.

A harmonic analysis provides two sets of solution: the real and imaginary components of a complex solution. According to Equation 5.53 (p. 187), and Equation 5.63 (p. 188) the magnitude of the real and imaginary sets describe the measurable field at t = 0 and at ωt = -90 degrees, respectively. Comparing Equation 5.54 (p. 187) and Equation 5.63 (p. 188) provides:

$$a(r)^2 = Q_r(r)^2 + Q_i(r)^2 \tag{5.65}$$

$$\tan(\phi(r)) = Q_i(r)/Q_r(r) \tag{5.66}$$

Equation 5.65 (p. 188) expresses the amplitude (peak) and phase angle of the measurable harmonic field quantities by the complex real and imaginary parts.

The time average of harmonic fields such as A, E, B, H, J, or V is zero at point r. This is not the case for P, W, or F because they are quadratic functions of B, H, or J. To derive the time dependence of a quadratic function - for the sake of simplicity - we deal only with a Lorentz force, F, which is product

of J and B. (This is a cross product; but components are not shown to simplify writing. The space dependence is also omitted.)

$$F^{jb}(t) = J(t)B(t) = (J_r\cos(\omega t) - J_i\sin(\omega t))(B_r\cos(\omega t) - B_i\sin(\omega t)) \\ = J_rB_r\cos(\omega t)^2 + J_iB_i\sin(\omega t)^2 - (J_iB_r + J_rB_i)\sin(\omega t)\cos(\omega t) \quad (5.67)$$

where:

F^{jb} = Lorentz Force density (output as FMAG on **PRESOL** command)

The time average of $\cos^2$ and $\sin^2$ terms is 1/2 whereas that of the sin cos term is zero. Therefore, the time average force is:

$$F^{jb} = 1/2(J_rB_r + J_iB_i) \quad (5.68)$$

Thus, the force can be obtained as the sum of "real" and "imaginary" forces. In a similar manner the time averaged Joule power density, Q^j, and magnetic energy density, W, can be obtained as:

$$Q^j = 1/2(J_rE_r + J_iE_i) \quad (5.69)$$

$$W = 1/4(B_rH_r + B_iH_i) \quad (5.70)$$

where:

W = magnetic energy density (output as SENE on **PRESOL** command)

Q^j = Joule Power density heating per unit volume (output as JHEAT on **PRESOL** command)

The time average values of these quadratic quantities can be obtained as the sum of real and imaginary set solutions.

The element returns the integrated value of F^{jb} is output as FJB and W is output as SENE. Q^j is the average element Joule heating and is output as JHEAT. For F and Q^j the 1/2 time averaging factor is taken into account at printout. For W the 1/2 time factor is ignored to preserve the printout of the real and imaginary energy values as the instantaneous stored magnetic energy at t = 0 and at ωt = -90 degrees, respectively. The element force, F, is distributed among nodes to prepare a magneto-structural coupling. The average Joule heat can be directly applied to thermoelectric coupling.

5.1.7. Nonlinear Time-Harmonic Magnetic Analysis

Many electromagnetic devices operate with a time-harmonic source at a typical power frequency. Although the power source is time-harmonic, numerical modeling of such devices can not be assumed as a linear harmonic magnetic field problem in general, since the magnetic materials used in these devices have nonlinear B-H curves. A time-stepping procedure should be used instead. This nonlinear transient procedure provides correct solutions for electromagnetic field distribution and waveforms, as well as global quantities such as force and torque. The only problem is that the procedure is often computationally intensive. In a typical case, it takes about 4-5 time cycles to reach a sinusoidal steady state. Since in each cycle, at least 10 time steps should be used, the analysis would require 40-50 nonlinear solution steps.

In many cases, an analyst is often more interested in obtaining global electromagnetic torque and power losses in a magnetic device at sinusoidal steady state, but less concerned with the actual flux density waveform. Under such circumstances, an approximate time-harmonic analysis procedure may

be pursued. If posed properly, this procedure can predict the time-averaged torque and power losses with good accuracy, and yet at much reduced computational cost.

The basic principle of the present nonlinear time-harmonic analysis is briefly explained next. First of all, the actual nonlinear ferromagnetic material is represented by another fictitious material based on energy equivalence. This amounts to replacing the DC B-H curve with a fictitious or effective B-H curve based on the following equation for a time period cycle T (Demerdash and Gillott([231] (p. 933))):

$$\frac{1}{2}\int_{o}^{B_{eff}} H_m dB_{eff} = \frac{4}{T}\int_{0}^{\frac{T}{4}}\left(\int_{0}^{B} H_m \sin(\omega t) dB\right) dt \tag{5.71}$$

where:

H_m = peak value of magnetic field
B = magnetic flux density
B_{eff} = effective magnetic flux density
T = time period
ω = angular velocity
t = time

With the effective B-H curve, the time transient is suppressed, and the nonlinear transient problem is reduced to a nonlinear time-harmonic one. In this nonlinear analysis, all field quantities are all sinusoidal at a given frequency, similar to the linear harmonic analysis, except that a nonlinear solution has to be pursued.

It should be emphasized that in a nonlinear transient analysis, given a sinusoidal power source, the magnetic flux density B has a non-sinusoidal waveform. While in the nonlinear harmonic analysis, B is assumed sinusoidal. Therefore, it is not the true waveform, but rather represents an approximation of the fundamental time harmonic of the true flux density waveform. The time-averaged global force, torque and loss, which are determined by the approximate fundamental harmonics of fields, are then subsequently approximation to the true values. Numerical benchmarks show that the approximation is of satisfactory engineering accuracy.

5.1.8. Electric Scalar Potential

Neglecting the time-derivative of magnetic flux density $\left\{\frac{\partial B}{\partial t}\right\}$ (the quasistatic approximation), the system of Maxwell's equations (Equation 5.1 (p. 177) through Equation 5.4 (p. 177)) reduces to:

$$\nabla \times \{H\} = \{J\} + \left\{\frac{\partial D}{\partial t}\right\} \tag{5.72}$$

$$\nabla \times \{E\} = \{0\} \tag{5.73}$$

$$\nabla \bullet \{B\} = 0 \tag{5.74}$$

$$\nabla \bullet \{D\} = \rho \tag{5.75}$$

As follows from Equation 5.73 (p. 190), the electric field {E} is irrotational, and can be derived from:

$$\{E\} = -\nabla V \tag{5.76}$$

where:

V = electric scalar potential

In the time-varying electromagnetic field governed by Equation 5.72 (p. 190) through Equation 5.75 (p. 190), the electric and magnetic fields are uncoupled. If only electric solution is of interest, replacing Equation 5.72 (p. 190) by the continuity Equation 5.5 (p. 178) and eliminating Equation 5.74 (p. 190) produces the system of differential equations governing the quasistatic electric field.

Repeating Equation 5.12 (p. 179) and Equation 5.13 (p. 179) without velocity effects, the constitutive equations for the electric fields become:

$$\{J\} = [\sigma]\{E\} \tag{5.77}$$

$$\{D\} = [\varepsilon]\{E\} \tag{5.78}$$

where:

$$[\sigma] = \begin{bmatrix} \frac{1}{\rho_{xx}} & 0 & 0 \\ 0 & \frac{1}{\rho_{yy}} & 0 \\ 0 & 0 & \frac{1}{\rho_{zz}} \end{bmatrix} = \text{electrical conductivity matrix}$$

$$[\varepsilon] = \begin{bmatrix} \varepsilon_{xx} & 0 & 0 \\ 0 & \varepsilon_{yy} & 0 \\ 0 & 0 & \varepsilon_{zz} \end{bmatrix} = \text{permittivity matrix}$$

ρ_{xx} = resistivity in the x-direction (input as RSVX on **MP** command)
ε_{xx} = permittivity in the x-direction (input as PERX on **MP** command)

The conditions for {E}, {J}, and {D} on an electric material interface are:

$$E_{t1} - E_{t2} = 0 \tag{5.79}$$

$$J_{1n} + \frac{\partial D_{1n}}{\partial t} = J_{2n} + \frac{\partial D_{2n}}{\partial t} \tag{5.80}$$

$$D_{1n} - D_{2n} = \rho_s \tag{5.81}$$

where:

E_{t1}, E_{t2} = tangential components of {E} on both sides of the interface
J_{n1}, J_{n2} = normal components of {J} on both sides of the interface
D_{n1}, D_{n2} = normal components of {D} on both sides of the interface
ρ_s = surface charge density

Two cases of the electric scalar potential approximation are considered below.

5.1.8.1. Quasistatic Electric Analysis

In this analysis, the relevant governing equations are Equation 5.76 (p. 190) and the continuity equation (below):

$$\nabla \bullet \left(\{J\} + \left\{ \frac{\partial \{D\}}{\partial t} \right\} \right) = 0 \tag{5.82}$$

Substituting the constitutive Equation 5.77 (p. 191) and Equation 5.78 (p. 191) into Equation 5.82 (p. 192), and taking into account Equation 5.76 (p. 190), one obtain the differential equation for electric scalar potential:

$$-\nabla \bullet ([\sigma]\nabla V) - \nabla \bullet \left([\varepsilon]\nabla \frac{\partial V}{\partial t} \right) = 0 \tag{5.83}$$

Equation 5.83 (p. 192) is used to approximate a time-varying electric field in elements PLANE230, SOLID231, and SOLID232. It takes into account both the conductive and dielectric effects in electric materials. Neglecting time-variation of electric potential Equation 5.83 (p. 192) reduces to the governing equation for steady-state electric conduction:

$$-\nabla \bullet ([\sigma]\nabla V) = 0 \tag{5.84}$$

In the case of a time-harmonic electric field analysis, the complex formalism allows Equation 5.83 (p. 192) to be re-written as:

$$-\nabla \bullet ([\varepsilon]\nabla V) + \frac{j}{\omega} \nabla \bullet ([\sigma]\nabla V) = 0 \tag{5.85}$$

where:

$j = \sqrt{-1}$

ω = angular frequency

Equation 5.85 (p. 192) is the governing equation for a time-harmonic electric analysis using elements PLANE121, SOLID122, and SOLID123.

In a time-harmonic analysis, the loss tangent tan δ can be used instead of or in addition to the electrical conductivity $[\sigma]$ to characterize losses in dielectric materials. In this case, the conductivity matrix $[\sigma]$ is replaced by the effective conductivity $[\sigma^{eff}]$ defined as:

$$[\sigma^{eff}] = [\sigma] + \omega[\varepsilon]\tan\delta \tag{5.86}$$

where:

tan δ = loss tangent (input as LSST on **MP** command)

5.1.8.2. Electrostatic Analysis

Electric scalar potential equation for electrostatic analysis is derived from governing Equation 5.75 (p. 190) and Equation 5.76 (p. 190), and constitutive Equation 5.78 (p. 191):

$$-\nabla \bullet ([\varepsilon]\nabla V) = \rho \tag{5.87}$$

Equation 5.87 (p. 192), subject to appropriate boundary conditions, is solved in an electrostatic field analysis of dielectrics using elements PLANE121, SOLID122, and SOLID123.

5.2. Derivation of Electromagnetic Matrices

The finite element matrix equations can be derived by variational principles. These equations exist for linear and nonlinear material behavior as well as static and transient response. Based on the presence of linear or nonlinear materials (as well as other factors), the program chooses the appropriate Newton-Raphson method. The user may select another method with the (**NROPT** command (see Newton-Raphson Procedure (p. 711))). When transient affects are to be considered a first order time integration scheme must be involved (**TIMINT** command (see Transient Analysis (p. 763))).

5.2.1. Magnetic Scalar Potential

The scalar potential formulations are restricted to static field analysis with partial orthotropic nonlinear permeability. The degrees of freedom (DOFs), element matrices, and load vectors are presented here in the following form (Zienkiewicz([75] (p. 924)), Chari([73] (p. 924)), and Gyimesi([141] (p. 928))):

5.2.1.1. Degrees of freedom

$\{\phi_e\}$ = magnetic scalar potentials at the nodes of the element (input/output as MAG)

5.2.1.2. Coefficient Matrix

$$[K^m] = [K^L] + [K^N] \tag{5.88}$$

$$[K^L] = \int_{vol} (\nabla\{N\}^T)^T [\mu](\nabla\{N\}^T) d(vol) \tag{5.89}$$

$$[K^N] = \int_{vol} \frac{\partial \mu_h}{\partial |H|} (\{H\}^T \nabla\{N\}^T)^T (\{H\}^T \nabla\{N\}^T) \frac{d(vol)}{|H|} \tag{5.90}$$

5.2.1.3. Applied Loads

$$[J_i] = \int_{vol} (\nabla\{N\}^T)^T [\mu](|H_g| + |H_c|) d(vol) \tag{5.91}$$

where:

$\{N\}$ = element shape functions ($\phi = \{N\}^T\{\phi_e\}$)

$$\nabla T = \text{gradient operator} = \left[\frac{\partial}{\partial x}\frac{\partial}{\partial y}\frac{\partial}{\partial z}\right]$$

vol = volume of the element

$\{H_g\}$ = preliminary or "guess" magnetic field (see Electromagnetic Field Fundamentals (p. 177))

$\{H_c\}$ = coercive force vector (input as MGXX, MGYY, MGZZ on **MP** command))

$[\mu]$ = permeability matrix (derived from input material property MURX, MURY, and MURZ (**MP** command) and/or material curve B versus H (accessed with **TB**,BH))(see Equation 5.7 (p. 178), Equation 5.8 (p. 178), and Equation 5.9 (p. 178))

$\frac{d\mu_h}{d|H|}$ = derivative of permeability with respect to magnitude of the magnetic field intensity (derived from the input material property curve B versus H (accessed with **TB**,BH))

The material property curve is input in the form of B values versus H values and is then converted to a spline fit curve of μ versus H from which the permeability terms μ_h and $\frac{d\mu_h}{d|H|}$ are evaluated.

The coercive force vector is related to the remanent intrinsic magnetization vector as:

$$[\mu]\{H_c\} = \mu_o\{M_o\} \tag{5.92}$$

where:

μ_o = permeability of free space (input as MUZRO on **EMUNIT** command)

The Newton-Raphson solution technique (*Option* on the **NROPT** command) is necessary for nonlinear analyses. Adaptive descent is also recommended (*Adaptky* on the **NROPT** command). When adaptive descent is used Equation 5.88 (p. 193) becomes:

$$[K^m] = [K^L] + (1-\xi)[K^N] \tag{5.93}$$

where:

ξ = descent parameter (see Newton-Raphson Procedure (p. 711))

5.2.2. Magnetic Vector Potential

The vector potential formulation is applicable to both static and dynamic fields with partial orthotropic nonlinear permeability. The basic equation to be solved is of the form:

$$[\bar{C}]\{\dot{u}\} + [\bar{K}]\{u\} = \{\bar{J}_i\} \tag{5.94}$$

The terms of this equation are defined below (Biro([120] (p. 927))).

5.2.2.1. Degrees of Freedom

$$\{u\} = \begin{Bmatrix} \{A_e\} \\ \{\nu_e\} \end{Bmatrix} \tag{5.95}$$

where:

$\{A_e\}$ = magnetic vector potentials (input/output as AX, AY, AZ)

$\{\nu_e\}$ = time integrated electric scalar potential ($\nu = \int$ Vdt) (input/output as VOLT)

The VOLT degree of freedom is a time integrated electric potential to allow for symmetric matrices.

5.2.2.2. Coefficient Matrices

$$[\bar{K}] = \begin{bmatrix} [K^{AA}] & [0] \\ [K^{vA}] & [0] \end{bmatrix} \tag{5.96}$$

$$[K^{AA}] = [K^L] + [K^N] + [K^G] \tag{5.97}$$

$$[K^L] = \int_{vol} (\nabla \times [N_A]^T)^T [\nu](\nabla \times [N_A]^T - [N_A][\sigma](\{v\} \times \nabla \times [N_A]^T))d(vol) \tag{5.98}$$

$$[K^G] = \int_{vol} (\nabla \cdot [N_A]^T)^T [\nu](\nabla \cdot [N_A]^T) d(vol) \tag{5.99}$$

$$[K^N] = 2 \int_{vol} \frac{d\nu_h}{d(|B|^2)} (\{B\}^T (\nabla \times [N_A]^T))^T (\{B\}^T (\nabla \times [N_A]^T)) d(vol) \tag{5.100}$$

$$[K^{VA}] = -\int (\nabla [N]^T)^T [\sigma]\{v\} \times \nabla \times [N_A]^T d(vol) \tag{5.101}$$

$$[\bar{C}] = \begin{bmatrix} [C^{AA}] & [C^{Av}] \\ [C^{Av}]^T & [C^{vv}] \end{bmatrix} \tag{5.102}$$

$$[C^{AA}] = \int_{vol} [N_A][\sigma][N_A]^T d(vol) \tag{5.103}$$

$$[C^{Av}] = \int_{vol} [N_A][\sigma]\nabla\{N\}^T d(vol) \tag{5.104}$$

$$[C^{vv}] = \int_{vol} (\nabla\{N\}^T)^T [\sigma]\nabla\{N\}^T d(vol) \tag{5.105}$$

5.2.2.3. Applied Loads

$$\{J_i\} = \begin{Bmatrix} \{J^A\} \\ \{I^t\} \end{Bmatrix} \tag{5.106}$$

$$\{J^A\} = \{J^S\} + \{J^{pm}\} \tag{5.107}$$

$$\{J^S\} = \int_{vol} \{J_s\}[N_A]^T d(vol) \tag{5.108}$$

$$\{J^{pm}\} = \int_{vol} (\nabla x [N_A]^T)^T \{H_c\} d(vol) \tag{5.109}$$

$$\{I^t\} = \int_{vol} \{J_t\}[N_A]^T d(vol) \tag{5.110}$$

where:

$[N_A]$ = matrix of element shape functions for {A}

$(\{A\} = [N_A]^T \{A_e\}; \{A_e\}^T = \lfloor \{A_{xe}\}^T \{A_{ye}\}^T \{A_{ze}\}^T \rfloor)$

[N] = vector of element shape functions for {V} ($V = \{N\}^T\{V_e\}$)

$\{J_s\}$ = source current density vector (input as JS on **BFE** command)

$\{J_t\}$ = total current density vector (input as JS on **BFE** command) (valid for 2-D analysis only)

vol = volume of the element

$\{H_c\}$ = coercive force vector (input as MGXX, MGYY, MGZZ on **MP** command)

ν_o = reluctivity of free space (derived from value using MUZRO on **EMUNIT** command)

$[\nu]$ = partially orthotropic reluctivity matrix (inverse of [μ], derived from input material property curve B versus H (input using **TB**,BH command))

$\frac{d\nu_h}{d(|B|)^2}$ = derivative of reluctivity with respect to the magnitude of magnetic flux squared (derived from input material property curve B versus H (input using **TB**,BH command))
$[\sigma]$ = orthotropic conductivity (input as RSVX, RSVY, RSVZ on **MP** command (inverse)) (see Equation 5.12 (p. 179)).
{v} = velocity vector

The coercive force vector is related to the remanent intrinsic magnetization vector as:

$$\{H_c\} = \frac{1}{\nu_o}[\nu]\{M_o\} \tag{5.111}$$

The material property curve is input in the form of B values versus H values and is then converted to a spline fit curve of ν versus $|B|^2$ from which the isotropic reluctivity terms ν_h and $\frac{d\nu_h}{d(|B|)^2}$ are evaluated.

The above element matrices and load vectors are presented for the most general case of a vector potential analysis. Many simplifications can be made depending on the conditions of the specific problem. In 2-D there is only one component of the vector potential as opposed to three for 3-D problems (AX, AY, AZ).

Combining some of the above equations, the variational equilibrium equations may be written as:

$$\{A_e\}^T([K^{AA}]\{A_e\} + [K^{AV}]\{\nu_e\} + [C^{AA}]d/dt\{A_e\} + [C^{AV}]d/dt\{\nu_e\} - \{J^A\}) = 0 \tag{5.112}$$

$$\{\nu_e\}^T([K^{VA}]\{A_e\} + [K^{VV}]\{\nu_e\} + [C^{VA}]d/dt\{A_e\} + [C^{VV}]d/dt\{\nu_e\} - \{I^t\}) = 0 \tag{5.113}$$

Here T denotes transposition.

Static analyses require only the magnetic vector potential degrees of freedom (KEYOPT controlled) and the K coefficient matrices. If the material behavior is nonlinear then the Newton-Raphson solution procedure is required (*Option* on the **NROPT** command (see Newton-Raphson Procedure (p. 711))).

For 2-D dynamic analyses a current density load of either source ($\{J_s\}$) or total ($\{J_t\}$) current density is valid. J_t input represents the impressed current expressed in terms of a uniformly applied current density. This loading is only valid in a skin-effect analysis with proper coupling of the VOLT degrees of freedom. In 3-D only source current density is allowed. The electric scalar potential must be constrained properly in order to satisfy the fundamentals of electromagnetic field theory. This can be achieved by direct specification of the potential value (using the **D** command) as well as with coupling and constraining (using the **CP** and **CE** commands).

The general transient analysis (**ANTYPE**,TRANS (see Element Reordering (p. 684))) accepts nonlinear material behavior (field dependent $[\nu]$ and permanent magnets (MGXX, MGYY, MGZZ). Harmonic transient analyses (**ANTYPE**,HARMIC (see Harmonic Analysis (p. 780))) is a linear analyses with sinusoidal loads; therefore, it is restricted to linear material behavior without permanent magnets.

5.2.3. Edge-Based Magnetic Vector Potential

The following section describes the derivation of the electromagnetic finite element equations used by SOLID236 and SOLID237 elements.

In an edge-based electromagnetic analysis, the magnetic vector potential {A} is approximated using the edge-based shape functions:

$$\{A\} = [W]^T \{A_e\} \tag{5.114}$$

where:

[W] = matrix of element vector (edge-based) shape functions.

$\{A_e\}$ = edge - flux at the element mid - side nodes (input/output as AZ). Edge - flux is defined as the line integral $= \int_L \{A\}^T d\{l\}$ of the magnetic vector potential along the element edge L.

The electric scalar potential V is approximated using scalar (node-based) element shape functions:

$$V=\{N\}^T \{V_e\} \tag{5.115}$$

where:

{N} = vector of element scalar (node-based) shape functions,

$\{V_e\}$ = electric scalar potential at the element nodes (input/output as VOLT).

Applying the variational principle to the governing electromagnetic equations (see Equation 5.50 (p. 187) - Equation 5.52 (p. 187)), we obtain the system of finite element equations:

$$\begin{bmatrix} [K^{AA}] & [K^{AV}] \\ [K^{VA}] & [K^{VV}] \end{bmatrix} \begin{Bmatrix} \{A_e\} \\ \{V_e\} \end{Bmatrix} + \begin{bmatrix} [C^{AA}] & [C^{AV}] \\ [K^{AV}]^T & [C^{VV}] \end{bmatrix} \begin{Bmatrix} \{\dot{A}_e\} \\ \{\dot{V}_e\} \end{Bmatrix} + \begin{bmatrix} [M^{AA}] & [0] \\ [C^{AV}]^T & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{A}_e\} \\ \{\ddot{V}_e\} \end{Bmatrix} = \begin{Bmatrix} \{J_e^s\} + \{J_e^{pm}\} \\ \{I_e\} \end{Bmatrix} \tag{5.116}$$

where:

$[K^{AA}] = \int_{vol} (\nabla \times [W]^T)^T [\nu](\nabla \times [W]^T) d(vol)$ = element magnetic reluctivity matrix,

$[K^L] = \int_{vol} \left(\left(\left(\nabla \times [W]^T \right) \right)^T [\nu] \left(\nabla \times [W]^T \right) - [W][\sigma] \left(\{v\} \times \nabla \times [W]^T \right) \right) d(vol)$ = element linear magnetic reluctivity matrix,

$[K^N] = 2 \int_{vol} \frac{d\nu_{fi}}{d\left(|B|^2\right)} \left(\{B\}^T \nabla \times [W]^T \right)^T \left(\{B\}^T \nabla \times [W]^T \right) d(vol)$ = element nonlinear magnetic reluctivity matrix,

$[K^{VV}] = \int_{vol} (\nabla \{N\}^T)^T [\sigma](\nabla \{N\}^T) d(vol)$ = element electric conductivity matrix,

$[K^{AV}] = \int_{vol} [W][\sigma](\nabla \{N\}^T) d(vol)$ = element magneto-electric coupling matrix,

$\left[K^{VA}\right] = -\int_{vol}\left(\nabla\{N\}^T\right)^T[\sigma]\{v\}\times\nabla\times[W]^T\,d(vol)$ = element electromagnetic coupling matrix,

$[C^{AA}] = \int_{vol}[W][\sigma][W]^T\,d(vol)$ = element eddy current damping matrix,

$[C^{VV}] = \int_{vol}(\nabla\{N\}^T)^T[\varepsilon](\nabla\{N\}^T)d(vol)$ = element displacement current damping matrix,

$[C^{AV}] = \int_{vol}[W][\varepsilon](\nabla\{N\}^T)d(vol)$ = element magneto-dielectric coupling matrix,

$[M^{AA}] = \int_{vol}[W][\varepsilon][W]^T\,d(vol)$ = element displacement current mass matrix,

$\{J_e^s\}=\int_{vol}[W]^T\{J_s\}d(vol)$ = element source current density vector,

$\{J_e^{pm}\} = \int_{vol}(\nabla x[W]^T)^T\{H_c\}d(vol)$ = element remnant magnetization load vector,

vol = element volume,

$[\nu]$ = reluctivity matrix (inverse of the magnetic permeability matrix input as MURX, MURY, MURZ on **MP** command or derived from the B-H curve input on **TB** command),

$\frac{dv_h}{d(|B|)^2}$ = derivative of reluctivity with respect to the magnitude of magnetic flux squared (derived from the B-H curve (input via **TB**,BH command))

$[\sigma]$ = electrical conductivity matrix (inverse of the electrical resistivity matrix input as RSVX, RSVY, RSVZ on **MP** command),

$[\varepsilon]$= dielectric permittivity (input as PERX, PERY, PERZ on **MP** command) (applicable to a harmonic electromagnetic analysis (KEYOPT(1) = 1) only),

{v} = velocity vector (input as VELO on **BF** command) (applicable to electromagnetic analysis option (KEYOPT(1) = 1) only),

$\{J_s\}$ = source current density vector (input as JS on **BFE** command) (applicable to the stranded conductor analysis option (KEYOPT(1) = 0) only),

$\{H_c\}$ = coercive force vector (input as MGXX, MGYY, MGZZ on **MP** command),

$\{I_e\}$ = nodal current vector (input/output as AMPS).

Equation 5.116 (p. 197) describes the strong coupling between the magnetic edge-flux and the electric potential degrees of freedom is nonsymmetric. It can be made symmetric by either using the weak coupling option (KEYOPT(2) = 1) in static or transient analyses or using the time-integrated electric potential (KEYOPT(2) = 2) in transient or harmonic analyses. In the latter case, the VOLT degree of

freedom has the meaning of the time-integrated electric scalar potential $\int Vdt$, and Equation 5.116 (p. 197) becomes:

$$\begin{bmatrix} [K^{AA}] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{A_e\} \\ \{V_e\} \end{Bmatrix} + \begin{bmatrix} [C^{AA}] & [K^{AV}] \\ [K^{AV}]^T & [K^{VV}] \end{bmatrix} \begin{Bmatrix} \{\dot{A}_e\} \\ \{\dot{V}_e\} \end{Bmatrix} = \begin{Bmatrix} \{J_e^s\} + \{J_e^{pm}\} \\ \{I_e\} \end{Bmatrix} \tag{5.117}$$

5.2.4. Electric Scalar Potential

The electric scalar potential V is approximated over the element as follows:

$$V = \{N\}^T \{V_e\} \tag{5.118}$$

where:

{N} = element shape functions
$\{V_e\}$ = nodal electric scalar potential (input/output as VOLT)

5.2.4.1. Quasistatic Electric Analysis

The application of the variational principle and finite element discretization to the differential Equation 5.83 (p. 192) produces the matrix equation of the form:

$$[C^V]\{\dot{V}_e\} + [K^V]\{V_e\} = \{I_e\} \tag{5.119}$$

where:

$$[K^V] = \int_{VOL} (\nabla\{N\}^T)^T [\sigma^{eff}] (\nabla\{N\}^T) d(VOL) = \text{element electrical conductivity coefficient matrix}$$

$$[C^V] = \int_{VOL} (\nabla\{N\}^T)^T [\varepsilon] (\nabla\{N\}^T) d(VOL) = \text{element dielectric permittivity coefficient matrix}$$

vol = element volume
$[\sigma^{eff}]$ = "effective" conductivity matrix (defined by Equation 5.86 (p. 192))
$\{I_e\}$ = nodal current vector (input/output as AMPS)

Equation 5.119 (p. 199) is used in the finite element formulation of PLANE230, SOLID231, and SOLID232. These elements model both static (steady-state electric conduction) and dynamic (time-transient and time-harmonic) electric fields. In the former case, matrix $[C^V]$ is ignored.

A time-harmonic electric analysis can also be performed using elements PLANE121, SOLID122, and SOLID123. In this case, the variational principle and finite element discretization are applied to the differential Equation 5.85 (p. 192) to produce:

$$(j\omega[C^{vh}] + [K^{vh}])\{V_e\} = \{L_e^n\} \tag{5.120}$$

where:

$$[K^{vh}] = [C^V]$$

$$[C^{vh}] = -\frac{1}{\omega^2}[K^V]$$

$\{L_e^n\}$ = nodal charge vector (input/output as CHRG)

5.2.4.2. Electrostatic Analysis

The matrix equation for an electrostatic analysis using elements PLANE121, SOLID122, and SOLID123 is derived from Equation 5.87 (p. 192):

$$[K^{VS}]\{V_e\} = \{L_e\} \tag{5.121}$$

$[K^{VS}] = \int_{vol} (\nabla\{N\}^T)^T [\varepsilon](\nabla\{N\}^T) d(vol)$ = dielectric permittivity coefficient matrix

$$\{L_e\} = \{L_e^n\} + \{L_e^c\} + \{L_e^{sc}\}$$

$$\{L_e^c\} = \int_{vol} \{\rho\}\{N\}^T d(vol)$$

$$\{L_e^{sc}\} = \int_{s} \{\rho_s\}\{N\}^T d(vol)$$

$\{\rho\}$ = charge density vector (input as CHRGD on **BF** command)
$\{\rho_s\}$ = surface charge density vector (input as CHRGS on **SF** command)

5.3. Electromagnetic Field Evaluations

The basic magnetic analysis results include magnetic field intensity, magnetic flux density, magnetic forces and current densities. These types of evaluations are somewhat different for magnetic scalar and vector formulations. The basic electric analysis results include electric field intensity, electric current densities, electric flux density, Joule heat and stored electric energy.

5.3.1. Magnetic Scalar Potential Results

The first derived result is the magnetic field intensity which is divided into two parts (see Electromagnetic Field Fundamentals (p. 177)); a generalized field $\{H_g\}$ and the gradient of the generalized potential $-\nabla \varphi g$. This gradient (referred to here as $\{H_\varphi\}$) is evaluated at the integration points using the element shape function as:

$$\{H_\phi\} = -\nabla\{N\}^T\{\phi_g\} \tag{5.122}$$

where:

∇T = gradient operator $= \left[\frac{\partial}{\partial x}\frac{\partial}{\partial y}\frac{\partial}{\partial z}\right]$

$\{N\}$ = shape functions
$\{\omega_g\}$ = nodal generalized potential vector

The magnetic field intensity is then:

$$\{H\} = \{H_g\} + \{H_\phi\} \tag{5.123}$$

where:

{H} = magnetic field intensity (output as H)

Then the magnetic flux density is computed from the field intensity:

$$\{B\} = [\mu]\{H\} \tag{5.124}$$

where:

{B} = magnetic flux density (output as B)
[μ] = permeability matrix (defined in Equation 5.7 (p. 178), Equation 5.8 (p. 178), and Equation 5.9 (p. 178))

Nodal values of field intensity and flux density are computed from the integration points values as described in Nodal and Centroidal Data Evaluation (p. 409).

Magnetic forces are also available and are discussed below.

5.3.2. Magnetic Vector Potential Results

The magnetic flux density is the first derived result. It is defined as the curl of the magnetic vector potential. This evaluation is performed at the integration points using the element shape functions:

$$\{B\} = \nabla \times [N_A]^T \{A_e\} \tag{5.125}$$

where:

{B} = magnetic flux density (output as B)
∇ x = curl operator
$[N_A]$ = shape functions
$\{A_e\}$ = nodal magnetic vector potential

Then the magnetic field intensity is computed from the flux density:

$$\{H\} = [\nu]\{B\} \tag{5.126}$$

where:

{H} = magnetic field intensity (output as H)
$[\nu]$ = reluctivity matrix

Nodal values of field intensity and flux density are computed from the integration point value as described in Nodal and Centroidal Data Evaluation (p. 409).

Magnetic forces are also available and are discussed below.

For a vector potential transient analysis current densities are also calculated.

$$\{J_t\} = \{J_e\} + \{J_s\} + \{J_v\} \tag{5.127}$$

where:

$\{J_t\}$ = total current density

$$\{J_e\} = -[\sigma]\left\{\frac{\partial A}{\partial t}\right\} = -[\sigma]\frac{1}{n}\sum_{i=1}^{n}[N_A]^T\{A_e\} \tag{5.128}$$

where:

$\{J_e\}$ = current density component due to {A}
$[\sigma]$ = conductivity matrix
n = number of integration points
$[N_A]$ = element shape functions for {A} evaluated at the integration points
$\{A_e\}$ = time derivative of magnetic vector potential

and

$$\{J_s\} = -[\sigma]\nabla V = [\sigma]\frac{1}{n}\sum_{i=1}^{n}\nabla\{N\}^T\{V_e\} \tag{5.129}$$

where:

$\{J_s\}$ = current density component due to V
∇ = divergence operator
$\{V_e\}$ = electric scalar potential
{N} = element shape functions for V evaluated at the integration points

and

$$\{J_V\} = \{v\} \times \{B\} \tag{5.130}$$

where:

$\{J_v\}$ = velocity current density vector
{v} = applied velocity vector
{B} = magnetic flux density (see Equation 5.125 (p. 201))

5.3.3. Edge-Based Magnetic Vector Potential

The following section describes the results derived from an edge-based electromagnetic analysis using SOLID236 and SOLID237 elements.

The electromagnetic fields and fluxes are evaluated at the integration points as follows:

$$\{B\} = \nabla \times [W]^T\{A_e\} \tag{5.131}$$

$$\{H\}=[\nu]\{B\} \tag{5.132}$$

$$\{E\} = -\nabla\{N\}^T\{V_e\} - [W]^T\left\{\frac{\partial A_e}{\partial t}\right\} + \{v\} \times \{B\} \tag{5.133}$$

$$\{J_c\} = [\sigma]\{E\} \tag{5.134}$$

$$\{J_s\}=\{J_c\}+[\varepsilon]\left\{\frac{\partial E}{\partial t}\right\} \tag{5.135}$$

where:

{B} = magnetic flux density (output as B at the element nodes),

{H} = magnetic field intensity (output as H at the element nodes),

{E} = electric field intensity (output as EF at the element nodes),

$\{J_c\}$ = conduction current density (output as JC at the element nodes and as JT at the element centroid),

$\{J_s\}$ = total (conduction + displacement) current density (output as JS at the element centroid; same as JT in a static or transient analysis),

$\{A_e\}$= edge-flux at the element mid-side nodes (input/output as AZ),

$\{V_e\}$ = electric scalar potential at the element nodes (input/output as VOLT),

[W] = matrix of element vector (edge-based) shape functions,

{N} = vector of element scalar (node-based) shape functions,

$[\nu]$ = reluctivity matrix (inverse of the magnetic permeability matrix (input as MURX, MURY, MURZ on **MP** command or derived from the B-H curve input on **TB** command),

$[\sigma]$ = electrical conductivity matrix (inverse of the electrical resistivity matrix input as RSVX, RSVY, RSVZ on **MP** command),

$[\varepsilon]$ = dielectric permittivity (input as PERX, PERY, PERZ on **MP** command) (applicable to a harmonic electromagnetic analysis (KEYOPT(1) = 1) only).

$\{\nu\}$ = velocity vector (input as VELO on **BF** command) (applicable to electromagnetic analyses (KEYOPT(1) = 1) only)

Nodal values of the above quantities are computed from the integration point values as described in Nodal and Centroidal Data Evaluation (p. 409).

5.3.4. Magnetic Forces

Magnetic forces are computed by elements using the vector potential method (PLANE13, PLANE53, SOLID97, PLANE233, SOLID236 and SOLID237) and the scalar potential method (SOLID5, SOLID96, and SOLID98). Three different techniques are used to calculate magnetic forces at the element level.

5.3.4.1. Lorentz forces

Magnetic forces in current carrying conductors (element output quantity FJB) are numerically integrated using:

$$\{F^{jb}\} = \int_{vol} \{N\}^T (\{J\} \times \{B\}) d(vol) \tag{5.136}$$

where:

{N} = vector of shape functions

For a 2-D analysis, the corresponding electromagnetic torque about +Z is given by:

$$T^{jb} = \{Z\} \cdot \int_{vol} \{r\} \times (\{J\} \times \{B\}) d(vol) \tag{5.137}$$

where:

{Z} = unit vector along +Z axis

{r} = position vector in the global Cartesian coordinate system

In a time-harmonic analysis, the time-averaged Lorentz force and torque are computed by:

$$\{F_{av}^{jb}\} = \frac{1}{2}\int_{vol}\{N\}^T(\{J\}^* \times \{B\})d(vol) \tag{5.138}$$

and

$$T_{av}^{jb} = \{Z\} \cdot \int_{vol}\{r\} \times (\{J\} \times \{B\})d(vol) \tag{5.139}$$

respectively.

where:

{J}* = complex conjugate of {J}

5.3.4.2. Maxwell Forces

The Maxwell stress tensor is used to determine forces on ferromagnetic regions. Depending on whether the magnetic forces are derived from the Maxwell stress tensor using surface or volumetric integration, one distinguishes between the surface and the volumetric integral methods.

5.3.4.2.1. Surface Integral Method

This method is used by PLANE13, PLANE53, SOLID5, SOLID96, SOLID97, SOLID98 elements.

The force calculation is performed on surfaces of air material elements which have a nonzero face loading specified (MXWF on **SF** commands) (Moon([77] (p. 924))). For the 2-D application, this method uses extrapolated field values and results in the following numerically integrated surface integral:

$$\{F^{mx}\} = \frac{1}{\mu_o}\int_s \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{Bmatrix} n_1 \\ n_2 \end{Bmatrix} ds \tag{5.140}$$

where:

$\{F^{mx}\}$ = Maxwell force (output as FMX)

μ_o = permeability of free space (input on **EMUNIT** command)

$$T_{11} = B_x^2 - \frac{1}{2}|B|^2$$

$T_{12} = B_x B_y$

$T_{21} = B_x B_y$

$$T_{22} = B_y^2 - \frac{1}{2}|B|^2$$

3-D applications are an extension of the 2-D case.

For a 2-D analysis, the corresponding electromagnetic torque about +Z axis is given by:

$$T^{mx} = \{Z\} \cdot \frac{1}{\mu_o}\int_s \{r\} \times \left[(\hat{n} \cdot \{B\})\{B\} - \frac{1}{2}(\{B\} \cdot \{B\})\hat{n}\right] ds \tag{5.141}$$

where:

$\hat{n}$ = unit surface normal in the global Cartesian coordinate system

In a time-harmonic analysis, the time-averaged Maxwell stress tensor force and torque are computed by:

$$\{F_{av}^{mx}\} = \frac{1}{2\mu_o}\int_s \left[Re\left\langle (\hat{n}\cdot\{B\}^*)\{B\}\right\rangle - \frac{1}{2}(\{B\}\cdot\{B\}^*)\hat{n}\right] ds \tag{5.142}$$

and

$$T_{av}^{mx} = \{Z\}\cdot\frac{1}{2\mu_o}\int_s \{r\}\times\left[Re\left\langle (\hat{n}\cdot\{B\}^*)\{B\}\right\rangle - \frac{1}{2}(\{B\}\cdot\{B\}^*)\hat{n}\right] ds \tag{5.143}$$

respectively.

where:

{B}* = complex conjugate of {B}
Re{ } = denotes real part of a complex quantity

The FMAGSUM macro is used with this method to sum up Maxwell forces and torques on element component.

5.3.4.2.2. Volumetric Integral Method

This method is used by PLANE233, SOLID236, and SOLID237 elements with KEYOPT(8) = 0.

The Maxwell forces are calculated by the following volumetric integral:

$$\{F_e^{mx}\} = -\int_{vol}[B]^T\{T^{mx}\}d(vol) \tag{5.144}$$

where:

$\{F_e^{mx}\}$ = element magnetic Maxwell forces (output as FMAG at all the element nodes with KEYOPT(7) = 0 or at the element corner nodes only with KEYOPT(7) = 1),

[B] = strain-displacement matrix

$\{T^{mx}\}$ = Maxwell stress vector = $\{T_{11}\ T_{22}\ T_{33}\ T_{12}\ T_{23}\ T_{13}\}^T$

The **EMFT** macro can be used with this method to sum up Maxwell forces and torques.

5.3.4.3. Virtual Work Forces

Electromagnetic nodal forces (including electrostatic forces) are calculated using the virtual work principle. The two formulations currently used for force calculations are the element shape method (magnetic forces) and nodal perturbations method (electromagnetic forces).

5.3.4.3.1. Element Shape Method

Magnetic forces calculated using the virtual work method (element output quantity FVW) are obtained as the derivative of the energy versus the displacement (MVDI on **BF** commands) of the movable part. This calculation is valid for a layer of air elements surrounding a movable part (Coulomb([76] (p. 924))). To determine the total force acting on the body, the forces in the air layer surrounding it can be summed. The basic equation for force of an air material element in the s direction is:

$$F_s = \int_{vol} \{B\}^T \left\{\frac{\partial H}{\partial s}\right\} d(vol) + \int_{vol} \left(\int \{B\}^T \{dH\}\right) \frac{\partial}{\partial s} d(vol) \tag{5.145}$$

where:

F_s = force in element in the s direction

$\left\{\frac{\partial H}{\partial s}\right\}$ = derivative of field intensity with respect to displacements

s = virtual displacement of the nodal coordinates taken alternately to be in the X, Y, Z global directions

vol = volume of the element

For a 2-D analysis, the corresponding electromagnetic torque about +Z axis is given by:

$$T^{VW} = \{Z\} \cdot \frac{1}{\mu_o} \int_{vol} \{r\} \times \left[\frac{1}{2}(\{B\} \cdot \{B\}) \nabla\{s\} - (\{B\} \cdot \nabla\{s\})\{B\}\right] d(vol) \tag{5.146}$$

In a time-harmonic analysis, the time-averaged virtual work force and torque are computed by:

$$\{F_{av}^{VW}\} = \frac{1}{2\mu_o} \int_{vol} \left[\frac{1}{2}(\{B\}^* \cdot \{B\}) \nabla\{s\} - Re\left\langle (\{B\}^* \cdot \nabla\{s\})\{B\}\right\rangle\right] d(vol) \tag{5.147}$$

and

$$T_{av}^{VW} = \{Z\} \cdot \frac{1}{2\mu_o} \int_{vol} \{R\} \times \left[\frac{1}{2}(\{B\}^* \cdot \{B\}) \nabla\{s\} - Re\left\langle (\{B\}^* \cdot \nabla\{s\})\{B\}\right\rangle\right] d(vol) \tag{5.148}$$

respectively.

5.3.4.3.2. Nodal Perturbation Method

This method is used by PLANE121, SOLID122 and SOLID123 elements.

Electromagnetic (both electric and magnetic) forces are calculated as the derivatives of the total element coenergy (sum of electrostatic and magnetic coenergies) with respect to the element nodal coordinates (Gyimesi et al.([346] (p. 940))):

$$F_{xi} = \frac{1}{2} \frac{\partial}{\partial x_i} \left[\int_{vol} (\{d\}^T \{E\} + \{B\}^T \{H\}) d(vol)\right] \tag{5.149}$$

where:

F_{xi} = x-component (y- or z-) of electromagnetic force calculated in node i

x_i = nodal coordinate (x-, y-, or z-coordinate of node i)

vol = volume of the element

Nodal electromagnetic forces are calculated for each node in each element. In an assembled model the nodal forces are added up from all adjacent to the node elements. The nodal perturbation method provides consistent and accurate electric and magnetic forces (using the **EMFT** command macro).

5.3.5. Joule Heat in a Magnetic Analysis

Joule heat is computed by elements using the vector potential method (PLANE13, PLANE53, SOLID97, SOLID236, and SOLID237) if the element has a nonzero resistivity (material property RSVX) and a nonzero current density (either applied J_s or resultant J_t). It is available as the output power loss (output as JHEAT) or as the coupled field heat generation load (**LDREAD**,HGEN).

Joule heat per element is computed as:

1. **Static or Transient Magnetic Analysis**

$$Q^j = \frac{1}{n}\sum_{i=1}^{n}[\rho]\{J_{ti}\}\cdot\{J_{ti}\} \tag{5.150}$$

where:

Q^j = Joule heat per unit volume
n = number of integration points
$[\rho]$ = resistivity matrix (input as RSVX, RSVY, RSVZ on **MP** command)
$\{J_{ti}\}$ = total current density in the element at integration point i

2. **Harmonic Magnetic Analysis**

$$Q^j = \mathrm{Re}\left(\frac{1}{2n}\sum_{i=1}^{n}[\rho]\{J_{ti}\}\cdot\{J_{ti}\}^*\right) \tag{5.151}$$

where:

Re = real component
$\{J_{ti}\}$ = complex total current density in the element at integration point i
$\{J_{ti}\}^*$ = complex conjugate of $\{J_{ti}\}$

5.3.6. Electric Scalar Potential Results

The first derived result in this analysis is the electric field. By definition (Equation 5.76 (p. 190)), it is calculated as the negative gradient of the electric scalar potential. This evaluation is performed at the integration points using the element shape functions:

$$\{E\} = -\nabla\{N\}^T\{V_e\} \tag{5.152}$$

Nodal values of electric field (output as EF) are computed from the integration points values as described in Nodal and Centroidal Data Evaluation (p. 409). The derivation of other output quantities depends on the analysis types described below.

5.3.6.1. Quasistatic Electric Analysis

The conduction current and electric flux densities are computed from the electric field (see Equation 5.77 (p. 191) and Equation 5.78 (p. 191)):

$$\{J\} = [\sigma]\{E\} \tag{5.153}$$

$$\{D\} = ([\varepsilon'] - j[\varepsilon'])\{E\} \tag{5.154}$$

where:

$[\varepsilon'] = [\varepsilon]$

$[\varepsilon''] = \tan\delta[\varepsilon]$

$j = \sqrt{-1}$

Both the conduction current {J} and electric flux {D} densities are evaluated at the integration point locations; however, whether these values are then moved to nodal or centroidal locations depends on the element type used to do a quasistatic electric analysis:

- In a current-based electric analysis using elements PLANE230, SOLID231, and SOLID232, the conduction current density is stored at both the nodal (output as JC) and centoidal (output as JT) locations. The electric flux density vector components are stored at the element centroidal location and output as non-summable miscellaneous items;
- In a charge-based analysis using elements PLANE121, SOLID122, and SOLID123 (harmonic analysis), the conduction current density is stored at the element centroidal location (output as JT), while the electric flux density is moved to the nodal locations (output as D).

The total electric current $\{J_{tot}\}$ density is calculated as a sum of conduction {J} and displacement current $\left\{\frac{\partial D}{\partial t}\right\}$ densities:

$$\{J_{tot}\} = \{J\} + \left\{\frac{\partial D}{\partial t}\right\} \tag{5.155}$$

The total electric current density is stored at the element centroidal location (output as JS). It can be used as a source current density in a subsequent magnetic analysis (**LDREAD**,JS).

The Joule heat is computed from the centroidal values of electric field and conduction current density. In a steady-state or transient electric analysis, the Joule heat is calculated as:

$$Q = \{J\}^T\{E\} \tag{5.156}$$

where:

Q = Joule heat generation rate per unit volume (output as JHEAT)

In a harmonic electric analysis, the Joule heat generation value per unit volume is time-averaged over a one period and calculated as:

$$Q = \frac{1}{2}\mathrm{Re}(\{E\}^T\{J\}^*) \tag{5.157}$$

where:

Re = real component
{E}* = complex conjugate of {E}

The value of Joule heat can be used as heat generation load in a subsequent thermal analysis (**LDREAD**,HGEN).

In a transient electric analysis, the element stored electric energy is calculated as:

$$W = \frac{1}{2}\int_{vol} \{D\}^T \{E\} d(vol) \tag{5.158}$$

where:

W = stored electric energy (output as SENE)

In a harmonic electric analysis, the time-averaged electric energy is calculated as:

$$W = \frac{1}{4}\int_{vol} Re(\{E\}^T \{D\}^*) d(vol) \tag{5.159}$$

5.3.6.2. Electrostatic Analysis

The derived results in an electrostatic analysis are:

Electric field (see Equation 5.152 (p. 207)) at nodal locations (output as EF);
Electric flux density (see Equation 5.154 (p. 208)) at nodal locations (output as D);
Element stored electric energy (see Equation 5.158 (p. 209)) output as SENE

Electrostatic forces are also available and are discussed below.

5.3.7. Electrostatic Forces

Electrostatic forces are determined using the nodal perturbation method (recommended) described in Nodal Perturbation Method (p. 206) or the Maxwell stress tensor described here. This force calculation is performed on surfaces of elements which have a nonzero face loading specified (MXWF on **SF** commands). For the 2-D application, this method uses extrapolated field values and results in the following numerically integrated surface integral:

$$\{F^{mx}\} = \varepsilon_o \int_s \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{Bmatrix} n_1 \\ n_2 \end{Bmatrix} ds \tag{5.160}$$

where:

ε_o = free space permittivity (input as PERX on **MP** command)

$$T_{11} = E_x^2 - \frac{1}{2}|E|^2$$

$T_{12} = E_x E_y$
$T_{21} = E_y E_x$

$$T_{22} = E_y^2 - \frac{1}{2}|E|^2$$

n_1 = component of unit normal in x-direction
n_2 = component of unit normal in y-direction
s = surface area of the element face

$$|E|^2 = E_x^2 + E_y^2$$

3-D applications are an extension of the 2-D case.

5.4. Stranded Coil Analyses

In the magnetic vector potential formulations discussed in previous sections, the source current density {Js} is either known and input as **BFE**,,JS (stranded conductor analysis) or determined from a fully-coupled electromagnetic solution as a sum of eddy J_e (Equation 5.128 (p. 201)) and DC conduction J_s (Equation 5.129 (p. 202)) current densities (massive conductor analysis).

A stranded coil refers to a coil consisting of many fine turns of conducting wires. Because in the low-frequency approximation the cross-section of the wires is small compared to the skin depth, the eddy currents in the wire can be neglected and the magnitude of the current density within the wires can be considered constant. The coil can be energized by an applied voltage or current, or by a controlling electric circuit. The stranded coil analysis can be viewed as a special case of electromagnetic field analysis where the current flow direction in wires is known and determined by the winding, while the magnitude of the current in the coil or voltage drop across the coil can either be imposed or determined from the solution.

The following sections describe the magnetic and electric equations that govern the stranded coil and the different finite element formulations used by the current-technology and legacy electromagnetic elements. The formulations apply to static, transient and harmonic analysis types.

5.4.1. Governing Equations

In a stranded coil analysis, the magnetic field equation

$$\nabla \times \nu \nabla \times \{A\} - \{J\} = 0 \tag{5.161}$$

is coupled to the electric circuit equation

$$\Delta V = RI + \frac{\partial \Phi}{\partial t} \tag{5.162}$$

by the following expressions for the electric current density {J} and the magnetic flux Φ in the coil:

$$\{J\} = \{t\}\frac{N_c}{S_c} I \tag{5.163}$$

$$\Phi = \frac{N_c}{S_c}\int_{V_c} \{t\}^T \cdot \{A\} dV \tag{5.164}$$

where:

{A} = magnetic vector potential,

ν = magnetic reluctivity,

ΔV = voltage drop across the coil,

I = total electric current,

R = total DC resistance of the coil winding,

S_c = coil cross-sectional area,

N_c = number of coil turns,

V_c = coil volume,

{t} = winding direction vector (direction of {J}).

The coupled set of electromagnetic equations governing the stranded coil is obtained by eliminating the current density {J} and magnetic flux Φ from Equation 5.161 (p. 210) and Equation 5.162 (p. 210) using Equation 5.163 (p. 210) and Equation 5.164 (p. 210), respectively. Depending on whether the voltage drop across the coil ΔV or the total electric current I is considered to be an independent electric variable in the stranded coil analysis, one can distinguish between the A-VOLT-EMF and the A-CURR formulations respectively.

5.4.2. A-VOLT-EMF Formulation

This formulation is available with the stranded coil analysis option (KEYOPT(1) = 2) of the current-technology electromagnetic elements PLANE233, SOLID236 and SOLID237.

In addition to the magnetic vector potential ({A}) and the voltage drop across the coil (ΔV) unknowns, the coil electromotive force (E) is introduced as a degree of freedom to prevent the distribution of eddy currents in the wire:

$$E=\frac{N_c}{S_c}\int_{V_c}\{t\}^T\cdot\frac{d\{A\}}{dt}dV \tag{5.165}$$

The corresponding finite element matrix equation can be written as:

$$\begin{bmatrix}[K^{AA}] & [K^{AV}] & -[K^{AV}]\\ [0] & s[K^{VV}] & [0]\\ [0] & [0] & -[K^{VV}]\end{bmatrix}\begin{Bmatrix}\{A_e\}\\ \{\Delta V_e\}\\ \{E_e\}\end{Bmatrix}+\begin{bmatrix}[0] & [0] & [0]\\ s[K^{AV}]^T & [0] & [0]\\ -[K^{AV}]^T & [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\dot{A}_e\}\\ \{\Delta\dot{V}_e\}\\ \{\dot{E}_e\}\end{Bmatrix}=\begin{Bmatrix}\{0\}\\ \{I_e\}\\ \{0\}\end{Bmatrix} \tag{5.166}$$

where:

$[K^{AA}]$ = element magnetic reluctivity matrix defined in Equation 5.116 (p. 197),

$$[K^{AV}]=-\frac{N_c}{S_cR}\int_{vol}[W]\{t\}\{N\}^T\,d(vol)\ =\ \text{element flux linkage matrix}$$

$$[K^{VV}]=\frac{1}{RV_c}\int_{vol}\{N\}\{N\}^T\,d(vol)=\text{element conductivity matrix}$$

S_c = coil cross-sectional area (input as R2 (2-D) or R1 (3-D) on **R** command),

N_c = number of coil turns (input as R3 (2-D) or R2 (3-D) on **R** command),

V_c = coil volume (calculated (2-D) or input as R3 (3-D) on **R** command),

{t} = current direction vector (input as R5 (2-D) or R4-R6 (3-D) on **R** command),

R = total DC resistance of the coil (input as R6 (2-D) or R7 (3-D) on **R** command),

s = symmetry factor of the coil (input as R7 (2-D) or R8 (3-D) on **R** command),

{N} = vector of element scalar (node-based) shape functions,

[W] = {N} (2-D) or matrix of vector (edge-based) shape functions (3-D),

vol = element volume,

$\{I_e\}$ = nodal current vector (input/output as AMPS),

$\{A_e\}$ = nodal magnetic vector potential - Z-component of magnetic vector potential at element nodes (2-D) or edge-flux at element midside nodes (3-D) (input/output as AZ),

$\{\Delta V_e\}$ = nodal voltage drop across the coil (input/output as VOLT),

$\{E_e\}$ = nodal electromotive force in the coil (input/output as EMF).

Note that the VOLT and EMF degrees of freedom should be coupled for each coil using the **CP**,,VOLT and **CP**,,EMF commands.

Equation 5.166 (p. 211) that strongly couples magnetic and electric degrees of freedom in a stranded coil is nonsymmetric. It can be made symmetric by either using the weak coupling option (KEYOPT(2) = 1) in static or transient analyses or, provided the symmetry factor s = 1, by using the time-integrated voltage drop and emf formulation (KEYOPT(2) = 2) in transient or harmonic analyses. In the latter case, the VOLT and EMF degrees of freedom have the meaning of the time-integrated voltage drop across the coil $\int \Delta V dt$ and time-integrated electromotive force $\int E dt$, and Equation 5.166 (p. 211) becomes:

$$\begin{bmatrix} [K^{AA}] & [0] & [0] \\ [0] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{A_e\} \\ \{\int \Delta V_e dt\} \\ \{\int E_e dt\} \end{Bmatrix} + \begin{bmatrix} [0] & [K^{AV}] & -[K^{AV}] \\ s[K^{AV}]^T & s[K^{VV}] & [0] \\ -[K^{AV}]^T & [0] & -[K^{VV}] \end{bmatrix} \begin{Bmatrix} \{\dot{A}_e\} \\ \{\Delta V_e\} \\ \{E_e\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{I_e\} \\ \{0\} \end{Bmatrix} \quad (5.167)$$

With the A-VOLT-EMF formulation where VOLT and EMF are 'true' (not time-integrated) voltage drop and emf respectively, the stranded coil can be node-coupled coupled to the current-based circuit elements (CIRCU124) as they share the same VOLT degree of freedom.

The stranded coil analysis results include:

- Nodal magnetic flux density calculated as in Equation 5.125 (p. 201) (output as B)
- Nodal magnetic field intensity calculated as in Equation 5.126 (p. 201) (output as H)
- Element electric current density calculated as in Equation 5.163 (p. 210) (output as JT or JS)
- Nodal magnetic forces calculated as in Equation 5.144 (p. 205) (output as FMAG)
- Element Joule heat generation rate calculated as in Equation 5.150 (p. 207) (output as JHEAT)

Note that the calculated current density JT (or JS) and the Joule heat generation rate JHEAT are effective in the sense that they are calculated based on the coil cross-sectional area (SC) and coil volume (VC), respectively, and those real constants include the wire and the non-conducting material filling the space between the winding.

5.4.3. A-CURR Formulation

This formulation is available with the voltage-fed (KEYOPT(1) = 2) and the circuit-coupled (KEYOPT(1) = 3) stranded coil analysis options of the legacy electromagnetic elements PLANE53 and SOLID97. With the A-CURR formulation, the unknowns are the components of the magnetic vector potential {A} (AX, AY, AZ) and the coil total current I (CURR).

Assume that a stranded coil has an isotropic and constant magnetic permeability and electric conductivity. Then, by using the magnetic vector potential approach from Electromagnetics (p. 177), the following element matrix equation is derived for the voltage-fed stranded coil analysis:

$$\begin{bmatrix} [0] & [0] \\ [C^{iA}] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{0\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [K^{Ai}] \\ [0] & [K^{ii}] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{i\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{V_o\} \end{Bmatrix} \tag{5.168}$$

where:

{A} = nodal magnetic vector potential vector (input/output as AX, AY, AZ)
{i} = nodal electric current vector (input/output as CURR)
$[K^{AA}]$ = potential stiffness matrix
$[K^{ii}]$ = resistive stiffness matrix
$[K^{Ai}]$ = potential-current coupling stiffness matrix
$[C^{iA}]$ = inductive damping matrix
$\{V_o\}$ = applied voltage drop vector (input/output as VLTG on **BFE**

To couple the A-CURR formulation of the stranded coil to the circuit, you need to use the circuit-coupled stranded coil option with PLANE53 and SOLID97 (KEYOPT(1) = 3) and the stranded coil option of CIRCU124 (KEYOPT(1) = 5). Both these options use the electromotive force degree of freedom EMF in addition to the A and CURR unknowns.

Assuming an isotropic and constant magnetic permeability and electric conductivity, the following element matrix equation is derived for a circuit-coupled stranded coil:

$$\begin{bmatrix} [0] & [0] & [0] \\ [C^{iA}] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{0\} \\ \{0\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [K^{Ai}] & [0] \\ [0] & [K^{ii}] & [K^{ie}] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{i\} \\ \{e\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{0\} \\ \{0\} \end{Bmatrix} \tag{5.169}$$

where:

{e} = nodal electromotive force drop (EMF)
$[K^{ie}]$ = current-emf coupling stiffness

The magnetic flux density {B}, the magnetic field intensity {H}, magnetic forces, and Joule heat can be calculated from the nodal magnetic vector potential {A} using Equation 5.124 (p. 201) and Equation 5.125 (p. 201).

The nodal electric current represents the current in a wire of the stranded coil. Therefore, there is only one independent electric current unknown in each stranded coil. In addition, there is no gradient or flux calculation associated with the nodal electric current vector.

5.5. Inductance, Flux and Energy Computation

Inductance plays an important role in the characterization of magnetic devices, electrical machines, sensors and actuators. The concept of a non-variant (time-independent), linear inductance of wire-like coils is discussed in every electrical engineering book. However, its extension to variant, nonlinear, distributed coil cases is far from obvious.

Time-variance is essential when the geometry of the device is changing: for example actuators, electrical machines. In this case, the inductance depends on a stroke (in a 1-D motion case) which, in turn, depends on time.

Many magnetic devices apply iron for the conductance of magnetic flux. Most iron has a nonlinear B-H curve. Because of this nonlinear feature, two kinds of inductance must be differentiated: differential and secant. The secant inductance is the ratio of the total flux over current. The differential inductance is the ratio of flux change over a current excitation change.

The flux of a single wire coil can be defined as the surface integral of the flux density. However, when the size of the wire is not negligible, it is not clear which contour spans the surface. The field within the coil must be taken into account. Even larger difficulties occur when the current is not constant: for example solid rotor or squirrel-caged induction machines.

The voltage induced in a variant coil can be decomposed into two major components: transformer voltage and motion induced voltage.

The transformer voltage is induced in coils by the rate change of exciting currents. It is present even if the geometry of the system is constant, the coils don't move or expand. To obtain the transformer voltage, the knowledge of flux change (i.e., that of differential flux) is necessary when the exciting currents are perturbed.

The motion induced voltage (sometimes called back-EMF) is related to the geometry change of the system. It is present even if the currents are kept constant. To obtain the motion induced voltage, the knowledge of absolute flux in the coils is necessary as a function of stroke.

Obtaining the proper differential and absolute flux values requires consistent calculations of magnetic absolute and incremental energies and co-energies. The **LMATRIX** command macro provides the absolute flux together with the incremental inductance. For legacy electromagnetic elements, the differential inductance and the absolute flux are provided by the **LMATRIX** command macro. For current-technology elements, the linear perturbation procedure can be used to calculate the differential inductance and the absolute flux using the incremental (IENE) and the co-energy (COEN) element records, respectively.

5.5.1. Differential Inductance Definition

Consider a magnetic excitation system consisting of n coils each fed by a current, I_i. The flux linkage ψ_i of the coils is defined as the surface integral of the flux density over the area multiplied by the number of turns, N_i, of the of the pertinent coil. The relationship between the flux linkage and currents can be described by the secant inductance matrix, $[L_s]$:

$$\{\psi\} = [L_s(t,\{I\})]\{I\} + \{\psi_0\} \tag{5.170}$$

where:

$\{\psi\}$ = vector of coil flux linkages
t = time

{I} = vector of coil currents.
$\{\psi_o\}$ = vector of flux linkages for zero coil currents (effect of permanent magnets)

Main diagonal element terms of $[L_s]$ are called self inductance, whereas off diagonal terms are the mutual inductance coefficients. $[L_s]$ is symmetric which can be proved by the principle of energy conservation.

In general, the inductance coefficients depend on time, t, and on the currents. The time dependent case is called time variant which is characteristic when the coils move. The inductance computation used by the program is restricted to time invariant cases. Note that time variant problems may be reduced to a series of invariant analyses with fixed coil positions. The inductance coefficient depends on the currents when nonlinear magnetic material is present in the domain.

The voltage vector, {U}, of the coils can be expressed as:

$$\{U\} = \frac{\partial}{\partial t}\{\psi\} \tag{5.171}$$

In the time invariant nonlinear case

$$\{U\} = \left(\frac{d[L_s]}{d\{I\}}\{I\} + [L_s]\right)\frac{\partial}{\partial t}\{I\} = [L_d\{I\}]\frac{\partial}{\partial t}\{I\} \tag{5.172}$$

The expression in the bracket is called the differential inductance matrix, $[L_d]$. The circuit behavior of a coil system is governed by $[L_d]$: the induced voltage is directly proportional to the differential inductance matrix and the time derivative of the coil currents. In general, $[L_d]$ depends on the currents, therefore it should be evaluated for each operating point.

5.5.2. Review of Inductance Computation Methods

After a magnetic field analysis, the secant inductance matrix coefficients, L_{sij}, of a coupled coil system could be calculated at postprocessing by computing flux linkage as the surface integral of the flux density, {B}. The differential inductance coefficients could be obtained by perturbing the operating currents with some current increments and calculating numerical derivatives. However, this method is cumbersome, neither accurate nor efficient. A much more convenient and efficient method is offered by the energy perturbation method developed by Demerdash and Arkadan([225] (p. 933)), Demerdash and Nehl([226] (p. 933)) and Nehl et al.([227] (p. 933)). The energy perturbation method is based on the following formula:

$$L_{dij} = \frac{d^2W}{dI_i dI_j} \tag{5.173}$$

where W is the magnetic energy, I_i and I_j are the currents of coils i and j. The first step of this procedure is to obtain an operating point solution for nominal current loads by a nonlinear analysis. In the second step linear analyses are carried out with properly perturbed current loads and a tangent reluctivity tensor, ν_t, evaluated at the operating point. For a self coefficient, two, for a mutual coefficient, four, incremental analyses are required. In the third step the magnetic energies are obtained from the incremental solutions and the coefficients are calculated according to Equation 5.173 (p. 215).

5.5.3. Inductance Computation Method Used

The inductance computation method used by the **LMATRIX** macro is based on Gyimesi and Ostergaard([229] (p. 933)) who revived Smythe's procedure([150] (p. 929)).

The incremental energy W_{ij} is defined by

$$W_{ij} = \frac{1}{2}\int \{\Delta H\}\{\Delta B\}dV \tag{5.174}$$

where $\{\Delta H\}$ and $\{\Delta B\}$ denote the increase of magnetic field and flux density due to current increments, ΔI_i and ΔI_j. The coefficients can be obtained from

$$W_{ij} = \frac{1}{2}L_{dij}\Delta I_i \Delta I_j \tag{5.175}$$

This allows an efficient method that has the following advantages:

1. For any coefficient, self or mutual, only one incremental analysis is required.
2. There is no need to evaluate the absolute magnetic energy. Instead, an "incremental energy" is calculated according to a simple expression.
3. The calculation of incremental analysis is more efficient: The factorized stiffness matrix can be applied. (No inversion is needed.) Only incremental load vectors should be evaluated.

For elements PLANE233, SOLID236 and SOLID237, the linear perturbation procedure can be used to derive the differential inductance from the element incremental energy (output as IENE). The incremental energy is calculated using Equation 5.174 (p. 216), where ΔH and ΔB are the linear perturbation analysis magnetic field and flux densities corresponding to the perturbation current loads ΔIi and ΔIj.

5.5.4. Transformer and Motion Induced Voltages

The absolute flux linkages of a time-variant multi-coil system can be written in general:

$$\{\psi\} = \{\psi\}(\{X\}(t), \{I\}(t)) \tag{5.176}$$

where:

{X} = vector of strokes

The induced voltages in the coils are the time derivative of the flux linkages, according to Equation 5.171 (p. 215). After differentiation:

$$\{U\} = \frac{d\{\psi\}}{d\{I\}}\frac{d\{I\}}{dt} + \frac{d\{\psi\}}{d\{X\}}\frac{d\{X\}}{dt} \tag{5.177}$$

$$\{U\} = [Ld(\{I\}, \{X\})]\frac{d\{I\}}{dt} + \frac{d\{\psi\}}{d\{X\}}\{V\} \tag{5.178}$$

where:

{V} = vector of stroke velocities

The first term is called transformer voltage (it is related to the change of the exciting current). The proportional term between the transformer voltage and current rate is the differential inductance matrix according to Equation 5.172 (p. 215).

The second term is the motion included voltage or back EMF (it is related to the change of strokes). The time derivative of the stroke is the velocity, hence the motion induced voltage is proportional to the velocity.

5.5.5. Absolute Flux Computation

Whereas the differential inductance can be obtained from the differential flux due to current perturbation as described in Differential Inductance Definition (p. 214), Review of Inductance Computation Methods (p. 215), and Inductance Computation Method Used (p. 216). The computation of the motion induced voltage requires the knowledge of absolute flux. In order to apply Equation 5.178 (p. 216), the absolute flux should be mapped out as a function of strokes for a given current excitation ad the derivative $\frac{d\{\psi\}}{d\{X\}}$ provides the matrix link between back EMF and velocity.

The absolute flux is related to the system co-energy by:

$$\{\psi\} = \frac{d\{W'\}}{d\{I\}} \tag{5.179}$$

According to Equation 5.179 (p. 217), the absolute flux can be obtained with an energy perturbation method by changing the excitation current for a given stroke position and taking the derivative of the system co-energy.

The increment of co-energy can be obtained by:

$$\Delta W_i' = \int B \Delta H_i dV \tag{5.180}$$

where:

W_i' = change of co-energy due to change of current Ii
ΔH_i = change of magnetic field due to change of current Ii

The incremental co-energy in Equation 5.180 (p. 217) is output as COEN in a linear perturbation analysis using elements PLANE233, SOLID236 and SOLID237.

5.5.6. Inductance Computations

The differential inductance matrix and the absolute flux linkages of coils can be calculated using the **LMATRIX** command macro for legacy electromagnetic elements and using linear perturbation analysis for the current-technology electromagnetic elements.

The differential inductance computation is based on the energy perturbation procedure using Equation 5.174 (p. 216) and Equation 5.175 (p. 216).

The absolute flux computation is based on the co-energy perturbation procedure using Equation 5.179 (p. 217) and Equation 5.180 (p. 217).

The output can be applied to compute the voltages induced in the coils using Equation 5.178 (p. 216).

5.5.7. Absolute Energy Computation

The absolute magnetic energy is defined by:

$$W_S = \int_0^B \{H\} d\{B\} \tag{5.181}$$

and the absolute magnetic co-energy is defined by:

$$W_C = \int_{-H_c}^{H} \{B\} d\{H\} \tag{5.182}$$

See Figure 5.3: Energy and Co-energy for Non-Permanent Magnets (p. 218) and Figure 5.4: Energy and Co-energy for Permanent Magnets (p. 219) for the graphical representation of these energy definitions. Equations and provide the incremental magnetic energy and incremental magnetic co-energy definitions used for inductance and absolute flux computations.

For legacy electromagnetic elements, the absolute magnetic energy and co-energy can be calculated using the **LMATRIX** command macro. For current-technology electromagnetic elements, the absolute magnetic energy and co-energy is output as MENE and COEN element records, respectively.

Figure 5.3: Energy and Co-energy for Non-Permanent Magnets

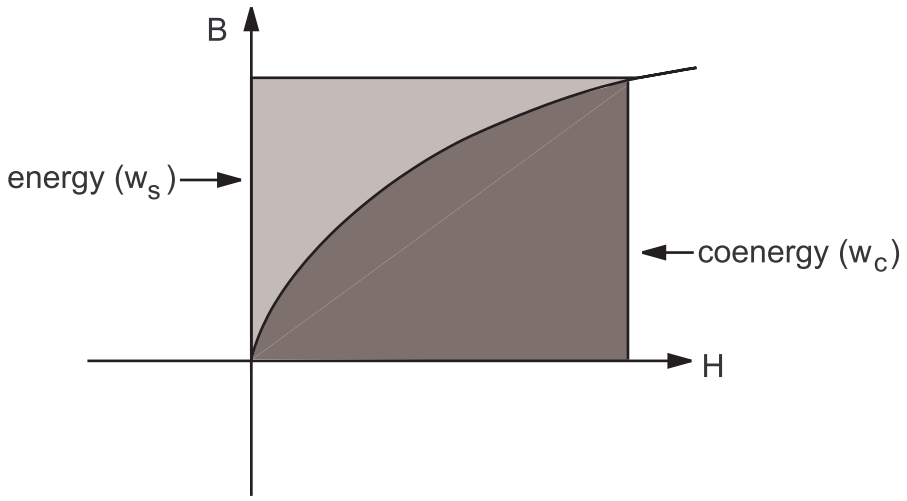
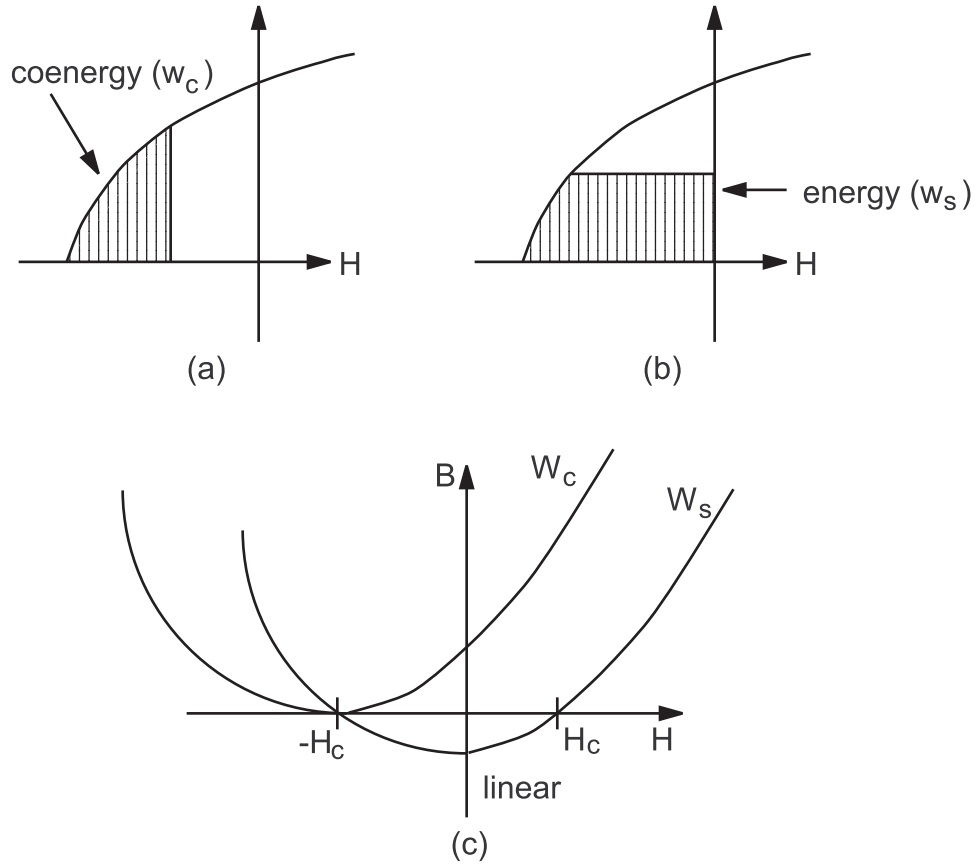


Figure 5.4: Energy and Co-energy for Permanent Magnets



Equation 5.174 (p. 216) and Equation 5.180 (p. 217) provide the incremental magnetic energy and incremental magnetic co-energy definitions used for inductance and absolute flux computations.

In addition to the magnetic energy and co-energy, elements PLANE233, SOLID236 and SOLID237 calculate the apparent energy defined as:

$$W_a = \frac{1}{2}\int_{vol} \{H\}^T \{B\}\, d(vol) \tag{5.183}$$

5.6. Electromagnetic Particle Tracing

Once the electromagnetic field is computed, particle trajectories can be evaluated by solving the equations of motion:

$$m\{a\} = \{F\} = q(\{E\} + \{v\} \times \{B\}) \tag{5.184}$$

where:

m = mass of particle
q = charge of particle
{E} = electric field vector
{B} = magnetic field vector
{F} = Lorentz force vector
{a} = acceleration vector
{v} = velocity vector

The tracing follows from element to element: the exit point of an old element becomes the entry point of a new element. Given the entry location and velocity for an element, the exit location and velocity can be obtained by integrating the equations of motion.

ANSYS particle tracing algorithm is based on Gyimesi et al.([228] (p. 933)) exploiting the following assumptions:

1. No relativistic effects (Velocity is much smaller than speed of light).
2. Pure electric tracing ({B} = {0}), pure magnetic tracing ({E} = {0}), or combined {E-B} tracing.
3. Electrostatic and/or magnetostatic analysis
4. Constant {E} and/or {B} within an element.
5. Quadrangle, triangle, hexahedron, tetrahedron, wedge or pyramid element shapes bounded by planar surfaces.

These simplifications significantly reduce the computation time of the tracing algorithm because the trajectory can be given in an analytic form:

1. parabola in the case of electric tracing
2. helix in the case of magnetic tracing.
3. generalized helix in the case of coupled E-B tracing.

The exit point from an element is the point where the particle trajectory meets the plane of bounding surface of the element. It can be easily computed when the trajectory is a parabola. However, to compute the exit point when the trajectory is a helix, a transcendental equation must be solved. A Newton Raphson algorithm is implemented to obtain the solution. The starting point is carefully selected to ensure convergence to the correct solution. This is far from obvious: about 70 sub-cases are differentiated by the algorithm. This tool allows particle tracing within an element accurate up to machine precision. This does not mean that the tracing is exact since the element field solution may be inexact. However, with mesh refinement, this error can be controlled.

Once a trajectory is computed, any available physical items can be printed or plotted along the path (using the **PLTRAC** command). For example, elapsed time, traveled distance, particle velocity components, temperature, field components, potential values, fluid velocity, acoustic pressure, mechanical strain, etc. Animation is also available.

The plotted particle traces consist of two branches: the first is a trajectory for a given starting point at a given velocity (forward ballistic); the second is a trajectory for a particle to hit a given target location at a given velocity (backward ballistics).

5.7. Capacitance Computation

Capacitance computation is one of the primary goals of an electrostatic analysis. For the definition of ground (partial) and lumped capacitance matrices see Vago and Gyimesi([239] (p. 934)). The knowledge of capacitance is essential in the design of electrostatic devices, Micro Electro Mechanical Systems (MEMS), transmission lines, printed circuit boards (PCB), electromagnetic interference and compatibility (EMI/EMC) etc. The computed capacitance can be the input of a subsequent MEMS analysis by an electrostructural transducer element TRANS126; for theory see TRANS126 - Electromechanical Transducer (p. 513).

To obtain inductance and flux using the **LMATRIX** command macro see Inductance, Flux and Energy Computation (p. 214).

The capacitance matrix of an electrostatic system can be computed (by the **CMATRIX** command macro). The capacitance calculation is based on the energy principle. For details see Gyimesi and Ostergaard([249] (p. 934)) and its successful application Hieke([251] (p. 935)). The energy principle constitutes the basis for inductance matrix computation, as shown in Inductance, Flux and Energy Computation (p. 214).

The electrostatic energy of a linear three electrode (the third is ground) system is:

$$W = \frac{1}{2}C_{11}^{g}V_1^2 + \frac{1}{2}C_{22}^{g}V_2^2 + C_{12}^{g}V_1V_2 \tag{5.185}$$

where:

W = electrostatic energy
V_1 = potential of first electrode with respect to ground
V_2 = potential of second electrode with respect to ground
C_{11}^{g} = self ground capacitance of first electrode
C_{22}^{g} = self ground capacitance of second electrode
C_{12}^{g} = mutual ground capacitance between electrodes

By applying appropriate voltages on electrodes, the coefficients of the ground capacitance matrix can be calculated from the stored static energy.

The charges on the conductors are:

$$Q_1 = C_{11}^{g}V_1 + C_{12}^{g}V_2 \tag{5.186}$$

$$Q_2 = C_{12}^{g}V_1 + C_{22}^{g}V_2 \tag{5.187}$$

where:

Q_1 = charge of first electrode
Q_2 = charge of second electrode

The charge can be expressed by potential differences, too:

$$Q_1 = C_{11}^{\ell}V_1 + C_{12}^{\ell}(V_1 - V_2) \tag{5.188}$$

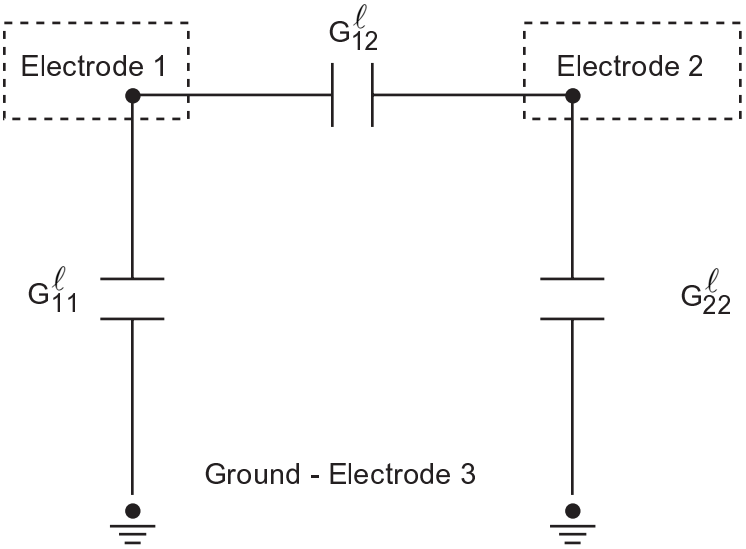
$$Q_2 = C_{22}^{\ell}V_2 + C_{12}^{\ell}(V_2 - V_1) \tag{5.189}$$

where:

C_{11}^{ℓ} = self lumped capacitance of first electrode
C_{22}^{ℓ} = self lumped capacitance of second electrode
C_{12}^{ℓ} = mutual lumped capacitance between electrode

The lumped capacitances can be realized by lumped capacitors as shown in Figure 5.5: Lumped Capacitor Model of Two Conductors and Ground (p. 222). Lumped capacitances are suitable for use in circuit simulators.

Figure 5.5: Lumped Capacitor Model of Two Conductors and Ground



In some cases, one of the electrodes may be located very far from the other electrodes. This can be modeled as an open electrode problem with one electrode at infinity. The open boundary region can be modeled by infinite elements or simply closing the FEM region far enough by an artificial Dirichlet boundary condition. In this case the ground key parameter (*GRNDKEY* on the **CMATRIX** command macro) should be activated. This key assumes that there is a ground electrode at infinity.

The previous case should be distinguished from an open boundary problem without an electrode at infinity. In this case the ground electrode is one of the modeled electrodes. The FEM model size can be minimized in this case, too, by infinite elements. When performing the capacitance calculation, however, the ground key (*GRNDKEY* on the **CMATRIX** command macro) should not be activated since there is no electrode at infinity.

5.8. Conductance Computation

Conductance computation is one of the primary goals of an electrostatic analysis. For the definition of ground (partial) and lumped conductance matrices see Vago and Gyimesi([239] (p. 934)). The knowledge of conductance is essential in the design of electrostatic devices, Micro Electro Mechanical Systems (MEMS), transmission lines, printed circuit boards (PCB), electromagnetic interference and compatibility (EMI/EMC) etc. The computed conductance can be the input of a subsequent MEMS analysis by an electrostructural transducer element TRANS126; for theory see TRANS126 - Electromechanical Transducer (p. 513).

To obtain inductance and flux using the **LMATRIX** command macro see Inductance, Flux and Energy Computation (p. 214).

The conductance matrix of an electrostatic system can be computed (by the **GMATRIX** command macro). The conductance calculation is based on the energy principle. For details see Gyimesi and Ostergaard([249] (p. 934)) and its successful application Hieke([251] (p. 935)). The energy principle constitutes the basis for inductance matrix computation, as shown in Inductance, Flux and Energy Computation (p. 214).

The electrostatic energy of a linear three conductor (the third is ground) system is:

$$W = \frac{1}{2} G_{11}^{g} V_1^2 + \frac{1}{2} G_{22}^{g} V_2^2 + G_{12}^{g} V_1 V_2 \tag{5.190}$$

where:

W = electrostatic energy
V_1 = potential of first conductor with respect to ground
V_2 = potential of second conductor with respect to ground

G_{11}^{g} = self ground conductance of first conductor

G_{22}^{g} = self ground conductance of second conductor

G_{12}^{g} = mutual ground conductance between conductors

By applying appropriate voltages on conductors, the coefficients of the ground conductance matrix can be calculated from the stored static energy.

The currents in the conductors are:

$$I_1 = G_{11}^{g} V_1 + G_{12}^{g} V_2 \tag{5.191}$$

$$I_2 = G_{12}^{g} V_1 + G_{22}^{g} V_2 \tag{5.192}$$

where:

I_1 = current in first conductor
I_2 = current in second conductor

The currents can be expressed by potential differences, too:

$$I_1 = G_{11}^{\ell} V_1 + G_{12}^{\ell} (V_1 - V_2) \tag{5.193}$$

$$I_2 = G_{22}^{\ell} V_2 + G_{12}^{\ell} (V_2 - V_1) \tag{5.194}$$

where:

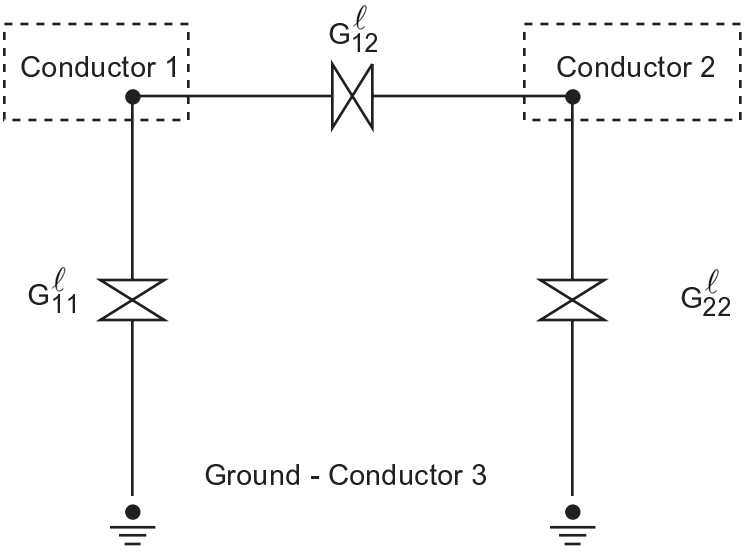
G_{11}^{ℓ} = self lumped conductance of first conductor

G_{22}^{ℓ} = self lumped conductance of second conductor

G_{12}^{ℓ} = mutual lumped conductance between conductors

The lumped conductances can be realized by lumped conductors as shown in Figure 5.6: Lumped Conductor Model of Two Conductors and Ground (p. 224). Lumped conductances are suitable for use in circuit simulators.

Figure 5.6: Lumped Conductor Model of Two Conductors and Ground



5.9. Hall Effect

The Hall effect is available using the electromagnetic analysis option (KEYOPT(1) = 1) of 3-D electromagnetic elements SOLID236 and SOLID237.

The Hall effect analysis is nonlinear and requires at least two iterations to achieve a converged solution. The Newton-Raphson algorithm will automatically be turned on when the Hall constant is specified. An electromagnetic analysis with the Hall effect can be steady-state or transient.

The electric constitutive relation is generalized as follows to include the Hall effect:

$$\{J\} = \left[\sigma^0\right]\left(\{E\} - \{E_H\}\right) \tag{5.195}$$

where:

{*J*} = electric current density vector

$$\left[\sigma^0\right] = \begin{bmatrix} \frac{1}{\rho_{xx}^0} & 0 & 0 \\ 0 & \frac{1}{\rho_{yy}^0} & 0 \\ 0 & 0 & \frac{1}{\rho_{zz}^0} \end{bmatrix} = \text{electric conductivity matrix without a magnetic field} \tag{5.196}$$

ρ_{xx} = electrical resistivity in the X-direction (input as *RSVX* on **MP** command)

{*E*} = electric field intensity vector

$\{E_H\} = -R_H[\{J\} \times \{B\}]$ = Hall field intensity vector

R_H = Hall coefficient (input as *RH* on **MP** command)

$\{B\}=\{B_x, B_y, B_z\}^T$ = magnetic flux density vector

Combining ohmic and Hall conductivity terms, Equation 5.195 (p. 224) can be rewritten using an effective anisotropic and nonsymmetric conductivity:

$$\{J\}=[H]^{-1}\left[\sigma^0\right]\{E\} \tag{5.197}$$

where:

$$[H]=\begin{bmatrix} 1 & -R_H\sigma_{xx}B_z & R_H\sigma_{xx}B_y \\ R_H\sigma_{yy}B_z & 1 & -R_H\sigma_{yy}B_x \\ -R_H\sigma_{zz}B_y & R_H\sigma_{zz}B_x & 1 \end{bmatrix} \tag{5.198}$$

Chapter 6: Heat Flow

The following heat flow topics are available:

6.1. Heat Flow Fundamentals

The following topics concerning heat flow fundamentals are available:

6.1.1. Conduction and Convection

The first law of thermodynamics states that thermal energy is conserved. Specializing this to a differential control volume:

$$\rho c\left(\frac{\partial T}{\partial t}+\{v\}^T\{L\}T\right)+\{L\}^T\{q\}=\ddot{q} \tag{6.1}$$

where:

ρ = density (input as DENS on **MP** command)
c = specific heat (input as C on **MP** command)
T = temperature (=T(x,y,z,t))
t = time

$$\{L\}=\begin{Bmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{Bmatrix}=\text{vector operator}$$

$$\{v\}=\begin{Bmatrix}v_x\\ v_y\\ v_z\end{Bmatrix}=\begin{array}{l}\text{velocity vector for mass transport of heat}\\ \text{(input as VX, VY, VZ on R command,}\\ \text{PLANE55 and SOLID70 only).}\end{array}$$

{q} = heat flux vector (output as TFX, TFY, and TFZ)

$\ddot{q}$ = heat generation rate per unit volume (input on **BF** or **BFE** commands)

It should be realized that the terms {L}T and $\{L\}^T\{q\}$ may also be interpreted as ∇T and $\nabla \cdot \{q\}$, respectively, where ∇ represents the grad operator and $\nabla \cdot$ represents the divergence operator.

Next, Fourier's law is used to relate the heat flux vector to the thermal gradients:

$$\{q\} = -[D]\{L\}T \tag{6.2}$$

where:

$$[D] = \begin{bmatrix} K_{xx} & 0 & 0 \\ 0 & K_{yy} & 0 \\ 0 & 0 & K_{zz} \end{bmatrix} = \text{conductivity matrix}$$

K_{xx}, K_{yy}, K_{zz} = conductivity in the element x, y, and z directions, respectively (input as KXX, KYY, KZZ on **MP** command)

Combining Equation 6.1 (p. 227) and Equation 6.2 (p. 228),

$$\rho c\left(\frac{\partial T}{\partial t} + \{v\}^T\{L\}T\right) = \{L\}^T([D]\{L\}T) + \ddot{q} \tag{6.3}$$

Expanding Equation 6.3 (p. 228) to its more familiar form:

$$\rho c\left(\frac{\partial T}{\partial t} + v_x\frac{\partial T}{\partial x} + v_y\frac{\partial T}{\partial y} + v_z\frac{\partial T}{\partial z}\right) = \ddot{q} + \frac{\partial}{\partial x}\left(K_x\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z\frac{\partial T}{\partial z}\right) \tag{6.4}$$

It will be assumed that all effects are in the global Cartesian system.

Three types of boundary conditions are considered. It is presumed that these cover the entire element.

1. Specified temperatures acting over surface S_1:

   $$T = T^* \tag{6.5}$$

   where T* is the specified temperature (input on **D** command).

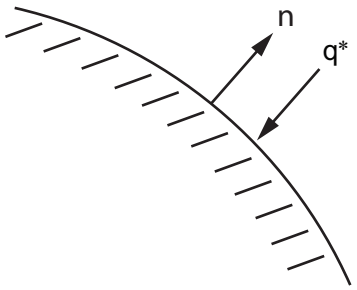
2. Specified heat flows acting over surface S_2:

   $$\{q\}^T\{n\} = -q^* \tag{6.6}$$

   where:

   {n} = unit outward normal vector
   q* = specified heat flow (input on **SF** or **SFE** commands)

Figure 6.1: Specified Heat Flows



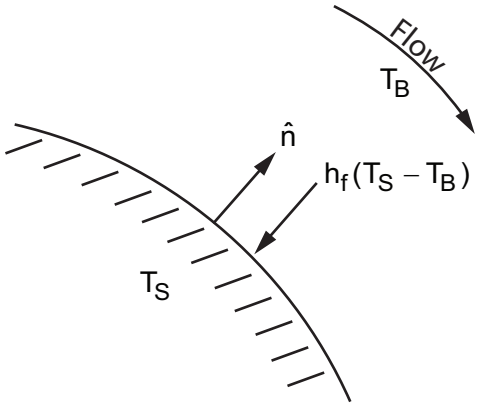
3. Specified convection surfaces acting over surface S_3 (Newton's law of cooling):

$$\{q\}^T\{n\} = h_f(T_S - T_B) \tag{6.7}$$

where:

h_f = film coefficient (input on **SF** or **SFE** commands) Evaluated at $(T_B + T_S)/2$ unless otherwise specified for the element
T_B = bulk temperature of the adjacent fluid (input on **SF** or **SFE** commands)
T_S = temperature at the surface of the model

Figure 6.2: Specified Convection Surfaces



For a fluid flowing past a solid surface, the bulk temperature T_B is equal to the free stream temperature, T_{FS}.

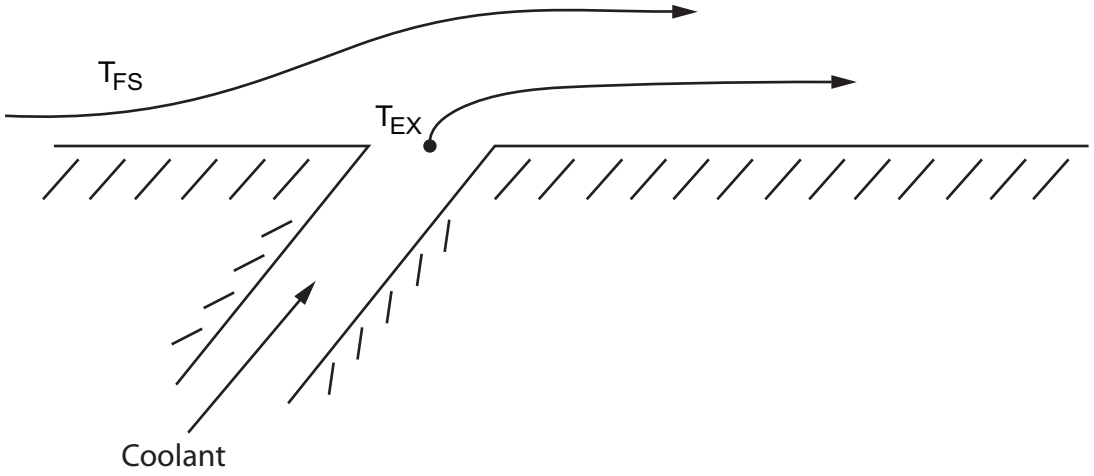
For the case of bleed holes in a solid, film effectiveness (η) accounts for the coolant bleeding through the cooling holes to the external surface of the solid. The bulk temperature T_B is then a combination of the free stream temperature T_{FS} and the temperature of the coolant exiting the bleed hole, T_{EX}:

$$T_B = T_{FS}(1-\eta) + T_{EX}(\eta) \tag{6.8}$$

where $0 < \eta < 1$.

Typically, T_{EX} is obtained from a FLUID116 element using the extra node option for SURF151 or SURF152.

Figure 6.3: Cooling Passages in a Solid



Note that positive specified heat flow is into the boundary (i.e., in the direction opposite of {n}), which accounts for the negative signs in Equation 6.6 (p. 228) and Equation 6.7 (p. 229).

Combining Equation 6.2 (p. 228) with Equation 6.6 (p. 228) and Equation 6.7 (p. 229)

$$\{n\}^T [D]\{L\}T = q^* \tag{6.9}$$

$$\{n\}^T [D]\{L\}T = h_f (T_B - T_s) \tag{6.10}$$

Premultiplying Equation 6.3 (p. 228) by a virtual change in temperature, integrating over the volume of the element, and combining with Equation 6.9 (p. 230) and Equation 6.10 (p. 230) with some manipulation yields:

$$\int_{vol}\left(\rho c\,\delta T\left(\frac{\partial T}{\partial t}+\{v\}^T\{L\}T\right)+\{L\}^T(\delta T)([D]\{L\}T)\right)d(vol)= \int_{S_2}\delta T\,q^* d(S_2)+\int_{S_3}\delta T\,h_f(T_B-T)d(S_3)+\int_{vol}\delta T\,\ddot{q}\,d(vol) \tag{6.11}$$

where:

vol = volume of the element
δT = an allowable virtual temperature (=$\delta T(x,y,z,t)$)

6.1.2. Radiation

Radiant energy exchange between neighboring surfaces of a region or between a region and its surroundings can produce large effects in the overall heat transfer problem. Though the radiation effects generally enter the heat transfer problem only through the boundary conditions, the coupling is especially strong due to nonlinear dependence of radiation on surface temperature.

Extending the Stefan-Boltzmann Law for a system of N enclosures, the energy balance for each surface in the enclosure for a gray diffuse body is given by Siegal and Howell([88] (p. 925)(Equation 8-19)) , which relates the energy losses to the surface temperatures:

$$\sum_{i=1}^{N}\left(\frac{\delta_{ji}}{\varepsilon_i}-F_{ji}\frac{1-\varepsilon_i}{\varepsilon_i}\right)\frac{1}{A_i}Q_i=\sum_{i=1}^{N}(\delta_{ji}-F_{ji})\sigma T_i^4 \tag{6.12}$$

where:

N = number of radiating surfaces
δ_{ji} = Kronecker delta
ε_i = effective emissivity (input on EMIS or **MP** command) of surface i
F_{ji} = radiation view factors (see below)
A_i = area of surface i
Q_i = energy loss of surface i
σ = Stefan-Boltzmann constant (input on STEF or **R** command)
T_i = absolute temperature of surface i

For a system of two surfaces radiating to each other, Equation 6.12 (p. 230) can be simplified to give the heat transfer rate between surfaces i and j as (see Chapman([356] (p. 941))):

$$Q_i = \frac{1}{\left(\frac{1-\varepsilon_i}{A_i\varepsilon_i} + \frac{1}{A_iF_{ij}} + \frac{1-\varepsilon_j}{A_j\varepsilon_j}\right)} \quad \sigma(T_i^4 - T_j^4) \tag{6.13}$$

where:

T_i, T_j = absolute temperature at surface i and j, respectively

If A_j is much greater than A_i, Equation 6.13 (p. 231) reduces to:

$$Q_i = A_i\varepsilon_iF_{ij}'\sigma(T_i^4 - T_j^4) \tag{6.14}$$

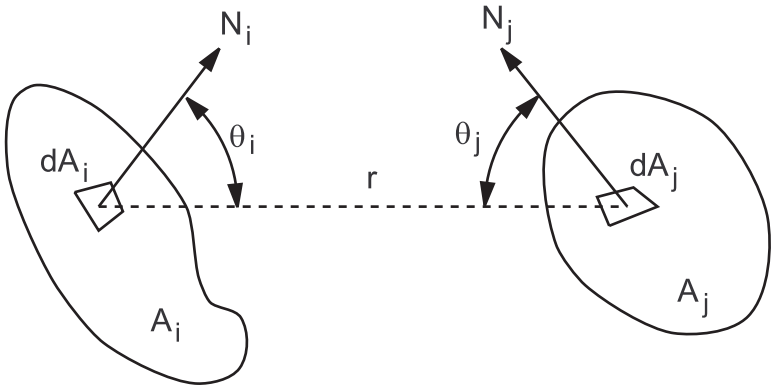
where:

$$F_{ij}' = \frac{Fij}{F_{ij}(1-\varepsilon_i)+\varepsilon_i}$$

6.1.2.1. View Factors

The view factor, F_{ij}, is defined as the fraction of total radiant energy that leaves surface i which arrives directly on surface j, as shown in Figure 6.4: View Factor Calculation Terms (p. 231). It can be expressed by the following equation:

Figure 6.4: View Factor Calculation Terms



$$F_{ij} = \frac{1}{A_i}\int_{A_i}\int_{A_j}\frac{\cos\theta_i \cos\theta_j}{\pi r^2}\,d(A_j)\,d(A_i) \tag{6.15}$$

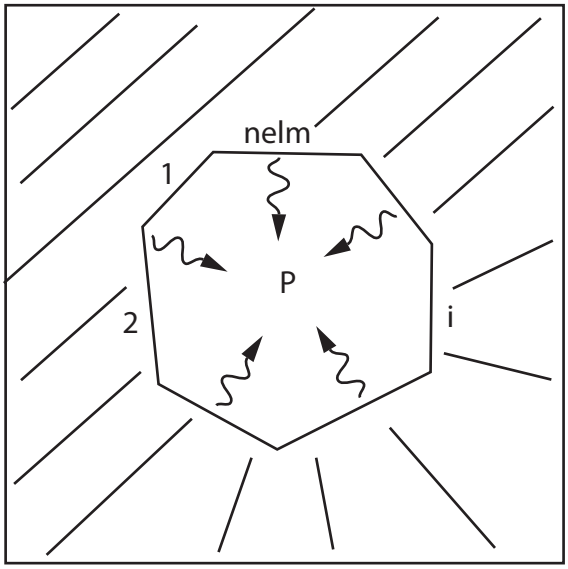
where:

A_i,A_j = area of surface i and surface j
r = distance between differential surfaces i and j
θ_i = angle between N_i and the radius line to surface $d(A_j)$
θ_j = angle between N_j and the radius line to surface $d(A_i)$
N_i,N_j = surface normal of $d(A_i)$ and $d(A_j)$

To ensure a good energy balance, it is important to satisfy both row sum and reciprocity relationships for the view factor matrix (**VFSM** command).

For a perfect enclosure, the row sum must satisfy the following requirement:

$$\sum_{j=1}^{nelm} F_{ij} = 1.0 \;\forall\; i = 1,2,\ldots nelm \text{ (Perfect)}$$

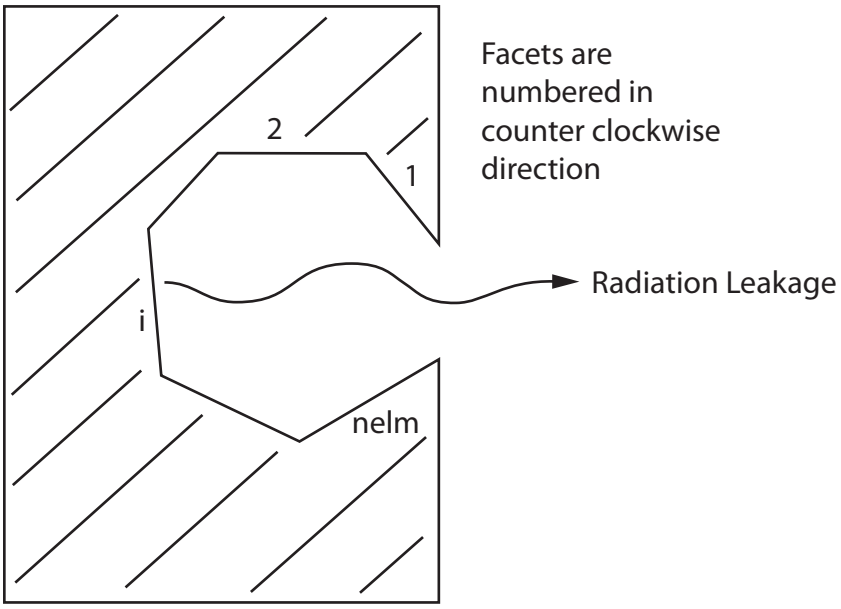


Facets are numbered in counter clockwise direction

where F_{ij} are the view factor matrix values.

For a leaky enclosure, the row sum must satisfy the following requirement:

$$\sum_{j=1}^{nelm} F_{ij} \le 1.0 \;\forall\; i = 1,2,\ldots nelm \text{ (Leaky)}$$



For a perfect enclosure, the following residual must be less than the specified convergence value (input as CONV on the **VFSM** command).

$$R_i' = \left| 1 - \sum_{j=1}^{nelm} F_{ij}' \right| \quad \forall\ i = 1,2,\ldots nelm$$

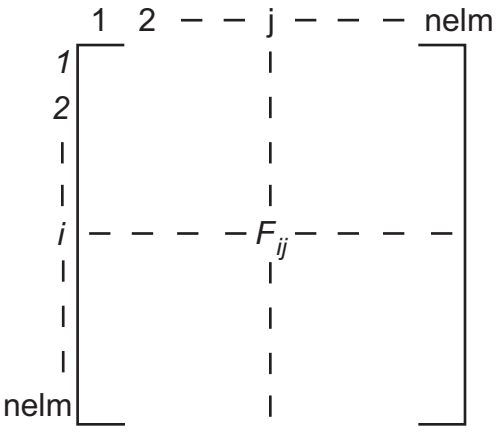
where ' indicates the new view factor values.

$$\begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ i \\ \vdots \\ nelm \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & - & - & j & - & - & - & nelm \end{array} \\ \begin{bmatrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ - & - & - & - & F_{ij}' & - & - & - & - \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{bmatrix} \end{array}$$

For a leaky enclosure, the following residual must be less than the specified convergence value (input as CONV on the **VFSM** command).

$$R_i' = \left| \sum_{j=1}^{nelm} F_{ij} - \sum_{j=1}^{nelm} F_{ij}' \right| \quad \forall\ i = 1,2,\ldots nelm$$

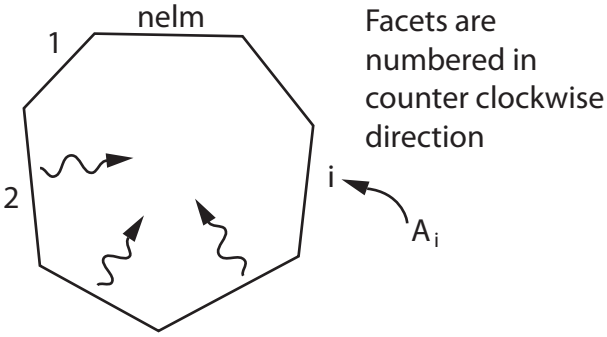
where F_{ij} is the original view factor values.



To ensure a good energy balance, the following reciprocity relationship must also be met:

$$A_iF_{ij}=A_jF_{ji} \quad \forall\ i=1,2,...nelm \text{ and } j=i+1,...nelm$$

where A_i is the area of the ith facet.



6.1.2.2. Radiation Usage

Four methods for analysis of radiation problems are included:

1. Radiation link element LINK31(LINK31 - Radiation Link (p. 430)). For simple problems involving radiation between two points or several pairs of points. The effective radiating surface area, the form factor and emissivity can be specified as real constants for each radiating point.

2. Surface effect elements - SURF151 in 2-D and SURF152 in 3-D for radiating between a surface and a point (SURF151 - 2-D Thermal Surface Effect (p. 532) and SURF152 - 3-D Thermal Surface Effect (p. 533)). The form factor between a surface and the point can be specified as a real constant or can be calculated from the basic element orientation and the extra node location.

3. Radiation matrix method (Radiation Matrix Method (p. 237)). For more generalized radiation problems involving two or more surfaces. The method involves generating a matrix of view factors between radiating surfaces and using the matrix as a superelement in the thermal analysis.

4. Radiosity solver method (Radiosity Solution Method (p. 241)). For generalized problems in 3-D involving two or more surfaces. The method involves calculating the view factor for the flagged radiating surfaces using the hemicube method and then solving the radiosity matrix coupled with the conduction problem.

6.2. Derivation of Heat Flow Matrices

As stated before, the variable T was allowed to vary in both space and time. This dependency is separated as:

$$T = \{N\}^T \{T_e\} \tag{6.16}$$

where:

T = T(x,y,z,t) = temperature
{N} = {N(x,y,z)} = element shape functions
$\{T_e\} = \{T_e(t)\}$ = nodal temperature vector of element

Thus, the time derivatives of Equation 6.16 (p. 235) may be written as:

$$\dot{T} = \frac{\partial T}{\partial t} = \{N\}^T \{\dot{T}_e\} \tag{6.17}$$

δT has the same form as T:

$$\delta T = \{\delta T_e\}^T \{N\} \tag{6.18}$$

The combination {L}T is written as

$$\{L\}T = [B]\{T_e\} \tag{6.19}$$

where:

[B] = {L}{N}T

Now, the variational statement of Equation 6.11 (p. 230) can be combined with Equation 6.16 (p. 235) thru Equation 6.19 (p. 235) to yield:

$$\begin{aligned}&\int_{vol} \rho c\{\delta T_e\}^T \{N\}\{N\}^T \{\dot{T}_e\} d(vol) + \int_{vol} \rho c\{\delta T_e\}^T \{N\}\{v\}^T [B]\{T_e\} d(vol)\\ &+\int_{vol} \{\delta T_e\}^T [B]^T [D][B]\{T_e\} d(vol) = \int_{S_2} \{\delta T_e\}^T \{N\} q^* d(S_2)\\ &+\int_{S_3} \{\delta T_e\}^T \{N\} h_f (T_B - \{N\}^T \{T_e\}) d(S_3) + \int_{vol} \{\delta T_e\}^T \{N\} \ddot{q}\, d(vol)\end{aligned} \tag{6.20}$$

Terms are defined in Heat Flow Fundamentals (p. 227). ρ is assumed to remain constant over the volume of the element. On the other hand, c and $\ddot{q}$ may vary over the element. Finally, $\{T_e\}$, $\{\dot{T}_e\}$, and $\{\delta T_e\}$ are nodal quantities and do not vary over the element, so that they also may be removed from the integral. Now, since all quantities are seen to be premultiplied by the arbitrary vector $\{\delta T_e\}$, this term may be dropped from the resulting equation. Thus, Equation 6.20 (p. 235) may be reduced to:

$$\begin{aligned}&\rho\int_{vol} c\{N\}\{N\}^T d(vol)\{\dot{T}_e\} + \rho\int_{vol} c\{N\}\{v\}^T [B] d(vol)\{T_e\}\\ &+\rho\int_{vol} [B]^T [D][B] d(vol)\{T_e\} = \int_{S_2} \{N\} q^* d(S_2) +\\ &\int_{S_3} T_B h_f \{N\} d(S_3) - \int_{S_3} h_f \{N\}\{N\}^T \{T_e\} d(S_3) + \int_{vol} q\{\ddot{N}\} d(vol)\end{aligned} \tag{6.21}$$

Equation 6.21 (p. 235) may be rewritten as:

$$[C_e^t]\{\dot{T}_e\} + ([K_e^{tm}] + [K_e^{tb}] + [K_e^{tc}])\{T_e\} = \{Q_e\} + \{Q_e^c\} + \{Q_e^g\} \tag{6.22}$$

where:

$[C_e^t] = \rho\int_{vol} c\{N\}\{N\}^T d(vol)$ = element specific heat (thermal damping) matrix

$[K_e^{tm}] = \rho\int_{vol} c\{N\}\{v\}^T [B] d(vol)$ = element mass transport conductivity matrix

$[K_e^{tb}] = \int_{vol} [B]^T [D][B] d(vol)$ = element diffusion conductivity matrix

$[K_e^{tc}] = \int_{S_3} h_f \{N\}\{N\}^T d(S_3)$ = element convection surface conductivity matrix

$\{Q_e^f\} = \int_{S_2} \{N\} q * d(S_2)$ = element mass flux vector

$\{Q_e^c\} = \int_{S_3} T_B h_f \{N\} d(S_3)$ = element convection surface heat flow vector

$\{Q_e^g\} = \int_{vol} \ddot{q}\{N\} d(vol)$ = element heat generation load

Comments on and modifications of the above definitions:

1. $[K_e^{tm}]$ is not symmetric.

2. $[K_e^{tc}]$ is calculated as defined above, for SOLID90 only. All other elements use a diagonal matrix, with the diagonal terms defined by the vector $\int_{S_3} h_f \{N\} d(S_3)$.

3. $[C_e^t]$ is frequently diagonalized, as described in Lumped Matrices (p. 391).

4. If $[C_e^t]$ exists and has been diagonalized and also the analysis is a transient (Key = ON on the **TIMINT** command), $\{Q_e^g\}$ has its terms adjusted so that they are proportioned to the main diagonal terms of $[C_e^t]$. $\{Q_e^j\}$, the heat generation rate vector for Joule heating is treated similarly, if present. This adjustment ensures that elements subjected to uniform heating will have a uniform temperature rise. However, this adjustment also changes nonuniform input of heat generation to an average value over the element.

5. For phase change problems, $[C_e^t]$ is evaluated from the enthalpy curve (Tamma and Namnuru([42] (p. 923))) if enthalpy is input (input as ENTH on **MP** command). This option should be used for phase change problems.

6.3. Heat Flow Evaluations

6.3.1. Integration Point Output

The element thermal gradients at the integration points are:

$$\{a\} = \{L\}T = \left[\frac{\partial T}{\partial x} \quad \frac{\partial T}{\partial y} \quad \frac{\partial T}{\partial z}\right]^T \tag{6.23}$$

where:

{a} = thermal gradient vector (output as TG)
{L} = vector operator
T = temperature

Using shape functions, Equation 6.23 (p. 236) may be written as:

$$\{a\} = [B]\{T_e\} \tag{6.24}$$

where:

[B] = shape function derivative matrix evaluated at the integration points
$\{T_e\}$ = nodal temperature vector of element

Then, the heat flux vector at the integration points may be computed from the thermal gradients:

$$\{q\} = -[D]\{a\} = -[D][B]\{T_e\} \tag{6.25}$$

where:

{q} = heat flux vector (output as TF)
[D] = conductivity matrix (see Equation 6.2 (p. 228))

Nodal gradient and flux vectors may be computed from the integration point values as described in Nodal and Centroidal Data Evaluation (p. 409).

6.3.2. Surface Output

The convection surface output is:

$$q^c = h_f (T_S - T_B) \tag{6.26}$$

where:

q^c = heat flow per unit area due to convection
h_f = film coefficient (input on **SF** or **SFE** commands)
T_S = temperature at surface of model
T_B = bulk temperature of the adjacent fluid (input on **SF** or **SFE** commands)

6.4. Radiation Matrix Method

In the radiation matrix method, for a system of two radiating surfaces, Equation 6.14 (p. 231) can be expanded as:

$$Q_i = \sigma \varepsilon_i F_{ij} A_i (T_i^2 + T_j^2)(T_i + T_j)(T_i - T_j) \tag{6.27}$$

or

$$Q_i = K'(T_i - T_j) \tag{6.28}$$

where:

$$K' = \sigma \varepsilon_i F_{ij} A_i (T_i^2 + T_j^2)(T_i + T_j)$$

K' cannot be calculated directly since it is a function of the unknowns T_i and T_j. The temperatures from previous iterations are used to calculate K' and the solution is computed iteratively.

For a more general case, Equation 6.12 (p. 230) can be used to construct a single row in the following matrix equation:

$$[C]\{Q\} = [D]\{T^4\} \tag{6.29}$$

such that:

$$\text{each row j in } [C] = \left(\frac{\delta_{ji}}{\varepsilon_i} - F_{ji} \frac{1-\varepsilon_i}{\varepsilon_i} \right) \frac{1}{A_i},\ i = 1,2 \ldots N \tag{6.30}$$

$$\text{each row j in } [D] = (\delta_{ji} - F_{ji})\ \sigma,\ i = 1,2 \ldots N \tag{6.31}$$

Solving for {Q}:

$$\{Q\} = [K^{ts}]\{T^4\} \tag{6.32}$$

and therefore:

$$[K^{ts}] = [C]^{-1}[D] \tag{6.33}$$

Equation 6.32 (p. 238) is analogous to Equation 6.12 (p. 230) and can be set up for standard matrix equation solution by the process similar to the steps shown in Equation 6.27 (p. 237) and Equation 6.28 (p. 237).

$$\{Q\} = [K']\{T\} \tag{6.34}$$

[K'] now includes T^3 terms and is calculated in the same manner as in Equation 6.28 (p. 237)). To be able to include radiation effects in elements other than LINK31, MATRIX50 (the substructure element) is used to bring in the radiation matrix. MATRIX50 has an option that instructs the solution phase to calculate [K']. The AUX12 utility is used to create the substructure radiation matrix. AUX12 calculates the effective conductivity matrix, $[K^{ts}]$, in Equation 6.32 (p. 238), as well as the view factors required for finding $[K^{ts}]$. The user defines flat surfaces to be used in AUX12 by overlaying nodes and elements on the radiating edge of a 2-D model or the radiating face of a 3-D model.

6.4.1. View Factor Calculation (2-D): Radiation Matrix Method

Two methods are available in the radiation matrix method to calculate the view factors (**VTYPE** command):

6.4.1.1. Non-Hidden Method

The non-hidden procedure calculates a view factor for every surface to every other surface whether the view is blocked by an element or not. In this procedure, the following equation is used and the integration is performed adaptively.

For a finite element discretized model, Equation 6.15 (p. 232) for the view factor F_{ij} between two surfaces i and j can be written as:

$$F_{ij} = \frac{1}{A_i} \sum_{p=1}^{m} \sum_{q=1}^{n} \left(\frac{\cos\theta_{ip} \cos\theta_{jq}}{\pi r^2} \right) A_{ip} A_{jq} \tag{6.35}$$

where:

m = number of integration points on surface i
n = number of integration points on surface j

When the dimensionless distance between two viewing surfaces D, defined in Equation 6.36 (p. 239), is less than 0.1, the accuracy of computed view factors is known to be poor (Siegal and Howell([88] (p. 925))).

$$D = \frac{d_{min}}{\sqrt{A_{max}}} \tag{6.36}$$

where:

d_{min} = minimum distance between the viewing surfaces A1 and A2
A_{max} = max (A1, A2)

So, the order of surface integration is adaptively increased from order one to higher orders as the value of D falls below 8. The area integration is changed to contour integration when D becomes less than 0.5 to maintain the accuracy. The contour integration order is adaptively increased as D approaches zero.

6.4.1.2. Hidden Method

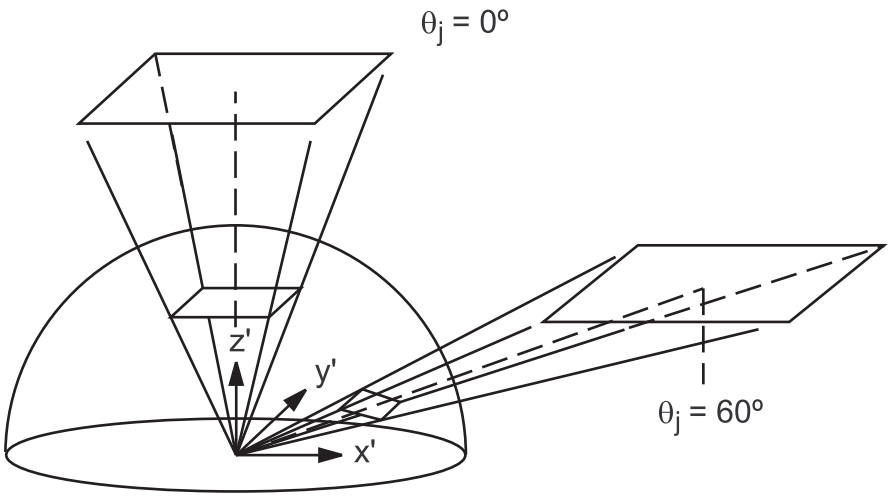
The hidden procedure is a simplified method which uses Equation 6.15 (p. 232) and assumes that all the variables are constant, so that the equation becomes:

$$F_{ij} = \frac{A_j}{\pi r^2} \cos\theta_i \cos\theta_j \tag{6.37}$$

The hidden procedure numerically calculates the view factor in the following conceptual manner. The hidden-line algorithm is first used to determine which surfaces are visible to every other surface. Then, each radiating, or "viewing", surface (i) is enclosed with a hemisphere of unit radius. This hemisphere is oriented in a local coordinate system (x' y' z'), whose center is at the centroid of the surface with the z axis normal to the surface, the x axis is from node I to node J, and the y axis orthogonal to the other axes. The receiving, or "viewed", surface (j) is projected onto the hemisphere exactly as it would appear to an observer on surface i.

As shown in Figure 6.5: Receiving Surface Projection (p. 240), the projected area is defined by first extending a line from the center of the hemisphere to each node defining the surface or element. That node is then projected to the point where the line intersects the hemisphere and transformed into the local system x' y' z', as described in Kreyszig([23] (p. 922))

Figure 6.5: Receiving Surface Projection



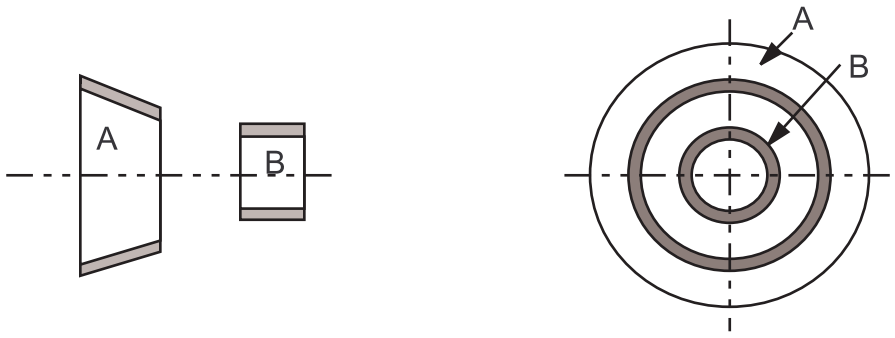
The view factor, F_{ij}, is determined by counting the number of rays striking the projected surface j and dividing by the total number of rays (N_r) emitted by surface i. This method may violate the radiation reciprocity rule, that is, $A_i F_{i-j} \neq A_j F_{j-i}$.

6.4.2. View Factors of Axisymmetric Bodies

When the radiation view factors between the surfaces of axisymmetric bodies are calculated (**GEOM**,1,n command), special logic is used. In this logic, the axisymmetric nature of the body is exploited to reduce the amount of computations. The user, therefore, needs only to build a model in plane 2-D representing the axisymmetric bodies as line "elements."

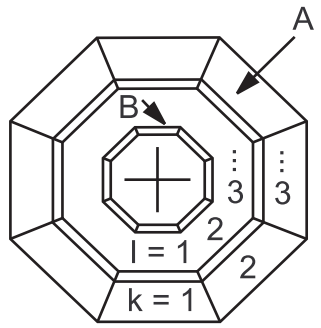
Consider two axisymmetric bodies A and B as shown in Figure 6.6: Axisymmetric Geometry (p. 240).

Figure 6.6: Axisymmetric Geometry



The view factor of body A to body B is calculated by expanding the line "element" model into a full 3-D model of n circumferential segments (**GEOM**,1,n command) as shown in Figure 6.7: End View of Showing n = 8 Segments (p. 241).

Figure 6.7: End View of Showing n = 8 Segments



View factor of body A to B is given by

$$F = \sum_{k=1}^{n} \sum_{\ell=1}^{n} F_{k-\ell} \tag{6.38}$$

where:

$F_{k-\ell}$ = view factor of segment k on body A to segment ℓ on body B

The form factors between the segments of the axisymmetric bodies are computed using the method described in the previous section. Since the coefficients are symmetric, the summation Equation 6.38 (p. 241) may be simplified as:

$$F = n \sum_{\ell=1}^{n} F_{1-\ell} \tag{6.39}$$

Both hidden and non-hidden methods are applicable in the computation of axisymmetric view factors. However, the non-hidden method should be used if and only if there are no blocking surfaces. For example, if radiation between concentric cylinders are considered, the outer cylinder can not see part of itself without obstruction from the inner cylinder. For this case, the hidden method must be used, as the non-hidden method would definitely give rise to inaccurate view factor calculations.

6.4.3. Space Node

A space node may be defined (**SPACE** command) to absorb all energy not radiated to other elements. Any radiant energy not incident on any other part of the model will be directed to the space node. If the model is not a closed system, then the user must define a space node with its appropriate boundary conditions.

6.5. Radiosity Solution Method

In the radiosity solution method for the analysis of gray diffuse radiation between N surfaces, Equation 6.12 (p. 230) is solved in conjunction with the basic conduction problem.

For the purpose of computation it is convenient to rearrange Equation 6.12 (p. 230) into the following series of equations

$$\sum_{j=1}^{N} \left[\delta_{ij} - (1-\varepsilon_i) F_{ij} \right] q_j^o = \varepsilon_i \sigma T_i^4 \tag{6.40}$$

and

$$q_i = q_i^o - \sum_{j=1}^{N} F_{ij} q_j^o \tag{6.41}$$

Equation 6.40 (p. 241) and Equation 6.41 (p. 242) are expressed in terms of the outgoing radiative fluxes (radiosity) for each surface, q_j^o, and the net flux from each surface q_i. For known surface temperatures, T_i, in the enclosure, Equation 6.41 (p. 242) forms a set of linear algebraic equations for the unknown, outgoing radiative flux (radiosity) at each surface. Equation 6.41 (p. 242) can be written as

$$[A]\{q^o\} = \{D\} \tag{6.42}$$

where:

$$A_{ij} = \delta_{ij} - (1-\varepsilon_i)F_{ij}$$

$$q_j^o = \text{radiosity flux for surface } i$$

$$D_i = \varepsilon_i \sigma T_i^4$$

[A] is a full matrix due to the surface to surface coupling represented by the view factors and is a function of temperature due to the possible dependence of surface emissivities on temperature. Equation 6.42 (p. 242) is solved using a Newton-Raphson procedure for the radiosity flux $\{q^o\}$.

When the q^o values are available, Equation 6.41 (p. 242) then allows the net flux at each surface to be evaluated. The net flux calculated during each iteration cycle is under-relaxed, before being updated using

$$q_i^{net} = \phi q_i^{k+1} + (1-\phi) q_i^k \tag{6.43}$$

where:

ϕ = radiosity flux relaxation factor
k = iteration number

The net surface fluxes provide boundary conditions to the finite element model for the conduction process. The radiosity Equation 6.42 (p. 242) is solved coupled with the conduction Equation 6.12 (p. 230) using a segregated solution procedure until convergence of the radiosity flux and temperature for each time step or load step.

The surface temperatures used in the above computation must be uniform over each surface in order to satisfy conditions of the radiation model. In the finite element model, each surface in the radiation problem corresponds to a face or edge of a finite element. The uniform surface temperatures needed for use in Equation 6.42 (p. 242) are obtained by averaging the nodal point temperatures on the appropriate element face.

For open enclosure problems using the radiosity method, an ambient temperature needs to be specified using a space temperature (**SPCTEMP** command) or a space node (**SPCNOD** command), to account for energy balance between the radiating surfaces and the ambient.

6.5.1. View Factor Calculation (3-D): Hemicube Method

For solving radiation problems in 3-D, the radiosity solution method calculates the view factors using the hemicube method as compared to the traditional double area integration method for 3-D geometry. Details related to using the hemicube method for view factor calculation are given in Glass ([272] (p. 936)) and Cohen and Greenberg ([276] (p. 936)). For 2-D and axisymmetric models, see View Factor Calculation (2-D): Radiation Matrix Method (p. 238) and View Factors of Axisymmetric Bodies (p. 240).

The hemicube method is based upon Nusselt's hemisphere analogy. Nusselt's analogy shows that any surface, which covers the same area on the hemisphere, has the same view factor. From this it is evident that any intermediate surface geometry can be used without changing the value of the view factors. In the hemicube method, instead of projecting onto a sphere, an imaginary cube is constructed around the center of the receiving patch. A patch in a finite element model corresponds to an element face of a radiating surface in an enclosure. The environment is transformed to set the center of the patch at the origin with the normal to the patch coinciding with the positive Z axis. In this orientation, the imaginary cube is the upper half of the surface of a cube, the lower half being below the 'horizon' of the patch. One full face is facing in the Z direction and four half faces are facing in the +X, -X, +Y, and -Y directions. These faces are divided into square 'pixels' at a given resolution, and the environment is then projected onto the five planar surfaces. Figure 6.8: The Hemicube (p. 243) shows the hemicube discretized over a receiving patch from the environment.

Figure 6.8: The Hemicube

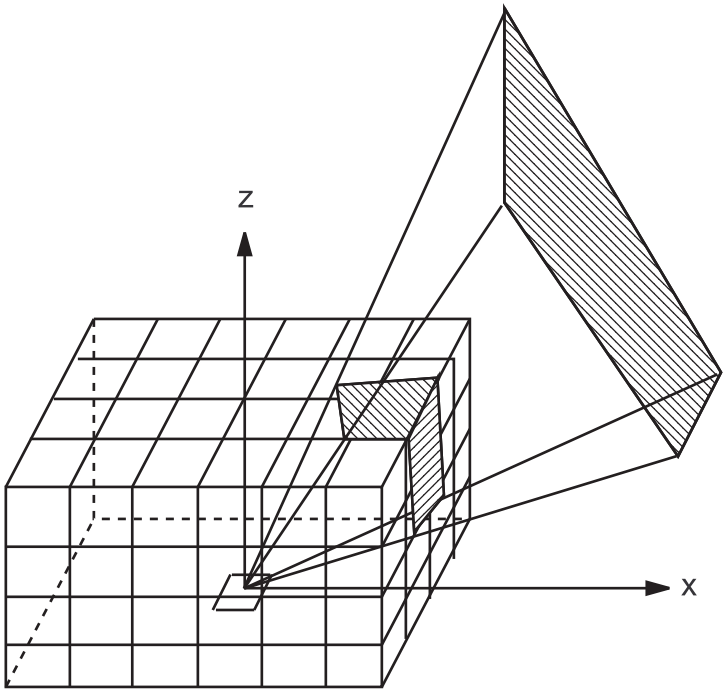
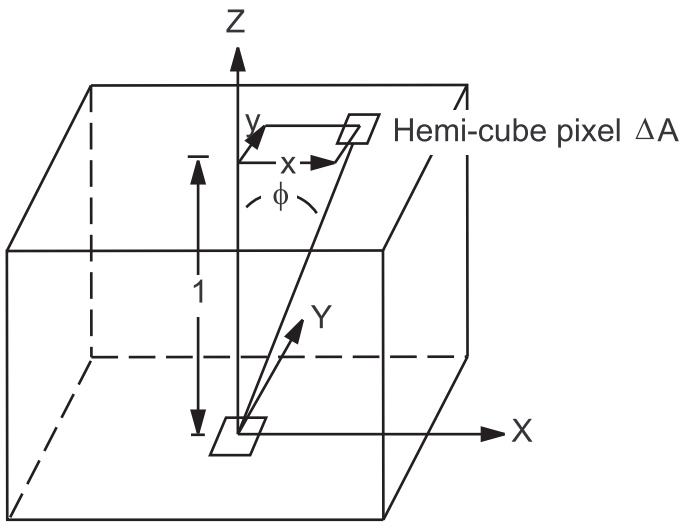


Figure 6.9: Derivation of Delta-View Factors for Hemicube Method



The contribution of each pixel on the cube's surface to the form-factor value varies and is dependent on the pixel location and orientation as shown in Figure 6.9: Derivation of Delta-View Factors for Hemicube Method (p. 244). A specific delta form-factor value for each pixel on the cube is found from modified form of Equation 6.15 (p. 232) for the differential area to differential area form-factor. If two patches project on the same pixel on the cube, a depth determination is made as to which patch is seen in that particular direction by comparing distances to each patch and selecting the nearer one. After determining which patch (j) is visible at each pixel on the hemicube, a summation of the delta form-factors for each pixel occupied by patch (j) determines the form-factor from patch (i) at the center of the cube to patch (j). This summation is performed for each patch (j) and a complete row of N form-factors is found.

At this point the hemicube is positioned around the center of another patch and the process is repeated for each patch in the environment. The result is a complete set of form-factors for complex environments containing occluded surfaces. The overall view factor for each surface on the hemicube is given by:

$$F_{ij} = \sum_{n=1}^{N} \Delta F_n = \frac{\cos\phi_i \cos\phi_j}{\pi r^2} \Delta A_j \tag{6.44}$$

where:

N = number of pixels
ΔF = delta-view factor for each pixel

The hemicube resolution (input on the **HEMIOPT** command) determines the accuracy of the view factor calculation and the speed at which they are calculated using the hemicube method. Default is set to 10. Higher values increase accuracy of the view factor calculation.

Chapter 7: Fluid Flow

Reduced order modeling of fluid flow effects is available in the ANSYS Mechanical APDL program. This chapter deals with the thin film effects which you can model using Mechanical APDL. For detailed fluid flow analysis capabilities, refer to the ANSYS Fluent and ANSYS CFX products.

A thin film is a small gap of fluid between moving surfaces. This thin layer of fluid can alter the structural response of the structure by adding stiffness and/or damping to the system. Movement normal to the gap produces a squeeze film effect (modeled with FLUID136 and FLUID138 elements). Movement tangential to the gap produces a slide film effect (modeled with FLUID139 elements).

The following thin film topics are available:

7.1. Squeeze Film Theory

Reynolds equations known from lubrication technology and theory of rarified gas physics are the theoretical background to analyze fluid structural interactions of microstructures (Blech([337] (p. 940)), Griffin([338] (p. 940)), Langlois([339] (p. 940))). FLUID136 and FLUID138 can be applied to structures where a small gap between two plates opens and closes with respect to time. This happens in the case of accelerometers where the seismic mass moves perpendicular to a fixed wall, in micromirror displays where the mirror plate tilts around a horizontal axis, and for clamped beams such as RF filters where a flexible structure moves with respect to a fixed wall. Other examples are published in literature (Mehner([340] (p. 940))).

FLUID136 and FLUID138 can be used to determine the fluidic response for given wall velocities. Both elements allow for static, harmonic and transient types of analyses. Static analyses can be used to compute damping parameter for low driving frequencies (compression effects are neglected). Harmonic analysis can be used to compute damping and squeeze effects at the higher frequencies. Transient analysis can be used for non-harmonic load functions. Both elements assume isothermal viscous flow.

7.1.1. Flow Between Flat Surfaces

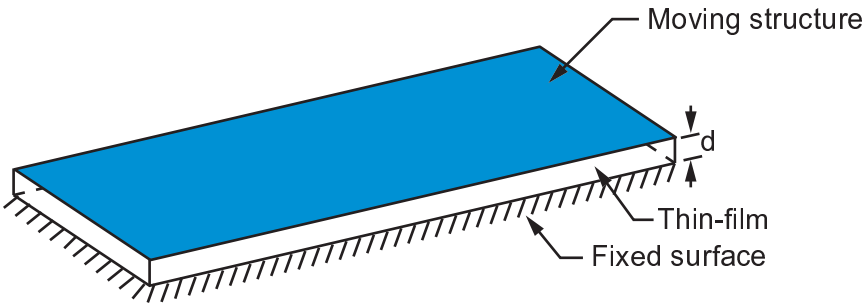
FLUID136 is used to model the thin-film fluid behavior between flat surfaces and is based on the generalized nonlinear Reynolds equation known from lubrication theory.

$$\frac{\partial(d\rho)}{\partial t} = \nabla \cdot \frac{\rho d^3}{12\eta} \nabla P_{abs} \tag{7.1}$$

where:

d = local gap separation
ρ = density
t = time
$\nabla \cdot$ = divergence operator
∇ = gradient operator

η = dynamic viscosity
P_{abs} = absolute pressure



Assuming an ideal gas:

$$\rho = \frac{P_{abs}}{RT} \tag{7.2}$$

where:

R = gas constant
T = temperature

Substituting Equation 7.2 (p. 246) into Equation 7.1 (p. 245) gives:

$$\frac{\partial(dP_{abs})}{\partial t} = \nabla \cdot \frac{P_{abs} d^3}{12\eta} \nabla P_{abs} \tag{7.3}$$

After substituting ambient pressure plus the pressure for the absolute pressure ($P_{abs} = P_0 + P$) this equation becomes:

$$\frac{\partial}{\partial x}\left(1+\frac{P}{P_o}\right)\frac{d^3}{12\eta}\frac{\partial P}{\partial x} + \frac{\partial}{\partial y}\left(1+\frac{P}{P_o}\right)\frac{d^3}{12\eta}\frac{\partial P}{\partial y} = \frac{d}{P_o}\frac{\partial P}{\partial t} + \left(1+\frac{P}{P_o}\right)\frac{\partial d}{\partial t} \tag{7.4}$$

Equation 7.4 (p. 246) is valid for large displacements and large pressure changes (KEYOPT(4) = 1). Pressure and velocity degrees of freedom must be activated (KEYOPT(3) = 1 or 2).

For small pressure changes ($P/P_0 << 1$), Equation 7.4 (p. 246) becomes:

$$\frac{d^3}{12\eta}\left(\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2}\right) = \frac{d}{P_o}\frac{\partial P}{\partial t} + \nu_z \tag{7.5}$$

where ν_z = wall velocity in the normal direction. That is:

$$\nu_z = \frac{\partial d}{\partial t} \tag{7.6}$$

Equation 7.5 (p. 246) is valid for large displacements and small pressure changes (KEYOPT(4) = 0). Pressure and velocity degrees of freedom must be activated (KEYOPT(3) = 1 or 2).

For small displacements ($d/d_0 << 1$) and small pressure changes ($P/P_0 << 1$), Equation 7.5 (p. 246) becomes:

$$\frac{d_o}{P_o}\frac{\partial P}{\partial t} + V_z = \frac{d_o^3}{12\eta}\left(\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2}\right) \tag{7.7}$$

where

d_o = nominal gap.

This equation applies when pressure is the only degree of freedom (KEYOPT(3) = 0).

For incompressible flows (ρ is constant), the generalized nonlinear Reynolds equation (Equation 7.1 (p. 245)) reduces to:

$$\frac{d^3}{12\eta}\left(\frac{\partial^2 P}{\partial x^2}+\frac{\partial^2 P}{\partial y^2}\right)=v_z \tag{7.8}$$

This equation applies for incompressible flow (KEYOPT(4) = 2). Pressure and velocity degrees of freedom must be activated (KEYOPT(3) = 1 or 2).

Reynolds squeeze film equations are restricted to structures with lateral dimensions much larger than the gap separation. Futhermore, viscous friction may not cause a significant temperature change. Continuum theory (KEYOPT(1) = 0) is valid for Knudsen numbers smaller than 0.01.

The Knudsen number Kn of the squeeze film problem can be estimated by:

$$Kn=\frac{L_o P_{ref}}{P_{abs}\, d} \tag{7.9}$$

where:

L_o = mean free path length of the fluid
P_{ref} = reference pressure for the mean free path L_o
$P_{abs} = P_o + P$

For small pressure changes, P_{abs} is approximately equal to P_0 and the Knudsen number can be estimated by:

$$Kn=\frac{L_o P_{ref}}{P_o\, d} \tag{7.10}$$

For systems that operate at Knudsen numbers <0.01, the continuum theory is valid (KEYOPT(1) = 0). The effective viscosity η_{eff} is then equal to the dynamic viscosity η.

For systems which operate at higher Knudsen numbers (KEYOPT(1) = 1), an effective viscosity η_{eff} considers slip flow boundary conditions and models derived from Boltzmann equation. This assumption holds for Knudsen numbers up to 880 (Veijola([342] (p. 940))):

$$\eta_{eff}=\frac{\eta}{1+9.638K_n^{1.159}} \tag{7.11}$$

For micromachined surfaces, specular reflection decreases the effective viscosity at high Knudsen numbers compared to diffuse reflection. Surface accommodation factors, α, distinguish between diffuse reflection ($\alpha = 1$), specular reflection ($\alpha = 0$), and molecular reflection ($0 < \alpha < 1$) of the molecules at the walls of the squeeze film. Typical accommodation factors for silicon are reported between 0.8 and 0.9, those of metal surfaces are almost 1. Different accommodation factors can be specified for each wall by using α_1 and α_2 (input as A1 and A2 on **R** command). α_1 is the coefficient associated with the top moving surface and α_2 is the coefficient associated with the bottom metallic surface. Results for

high Knudsen numbers with accommodation factors (KEYOPT(1) =2) are not expected to be the same as those for high Knudsen numbers without accommodation factors (KEYOPT(1) =1).

The effective viscosity equations for high Knudsen numbers are based on empirical correlations. Fit functions for the effective viscosity of micromachined surfaces are found in Veijola([342] (p. 940)). The effective viscosity is given by the following equation if $\alpha_1 = \alpha_2$:

$$\eta_{eff} = \frac{D\eta}{6Q_1} \tag{7.12}$$

and by the following equation if $\alpha_1 \neq \alpha_2$:

$$\eta_{eff} = \frac{D\eta}{6Q_3} \tag{7.13}$$

where D is the inverse Knudsen number:

$$D = \frac{\sqrt{\pi}}{2K_n}$$

and Q_1, Q_2, and Q_3 are Poiseuille flow rate coefficients:

$$Q_p = Q_p(D, \alpha_1, \alpha_2)$$

for p = 1, 2, or 3.

If both surfaces are the same ($\alpha_1 = \alpha_2$), the Poiseuille flow rate coefficient is given by:

$$Q_1(D, \alpha_1, \alpha_2) = \frac{D}{6} + \frac{1}{\alpha_1^{1.34}\sqrt{\pi}} \ln\left(\frac{1}{D} + 4.1\right)$$
$$+ \frac{\alpha_1}{6.4} + \frac{1.3(1-\alpha_1)}{1+0.08D^{1.83}} + \frac{0.64\alpha_1 D^{.17}}{1+1.12D^{.72}}$$

If the bottom fixed plate is metallic ($\alpha_2 = 1$) and the top moving plate is not metallic ($\alpha_1 \neq 1$), the Poiseuille flow rate coefficient is given by:

$$Q_2(D, \alpha_1, \alpha_2) = \frac{D}{6} + \frac{2-\alpha_1}{\sqrt{\pi}} \ln\left(\frac{1}{D} + 2.18\right) + \frac{\alpha_1}{.642}$$
$$+ \frac{(1-\alpha_1)(D+2.395)}{2+1.12\alpha_1 D} + \frac{1.26+10\alpha_1 D}{1+10.98D} + \frac{e^{-D/5}}{8.77}$$

The general solution valid for arbitrary α_1 and α_2 is a simple linear combination of Q_1 and Q_2:

$$Q_3(D, \alpha_1, \alpha_2) = \left(\frac{\alpha_2 - \alpha_1}{1-\alpha_1}\right) Q_2 + \left(\frac{1-\alpha_2}{1-\alpha_1}\right) Q_1$$

7.1.2. Flow in Channels

FLUID138 can be used to model the fluid flow though short circular and rectangular channels of micrometer size. The element assumes isothermal viscous flow at low Reynolds numbers, the channel length to be small compared to the acoustic wave length, and a small pressure drop with respect to ambient pressure.

In contrast to FLUID116, FLUID138 considers gas rarefaction, is more accurate for channels of rectangular cross sections, allows channel dimensions to be small compared to the mean free path, allows evacuated systems, and considers fringe effects at the inlet and outlet which considerably increase the damping force in case of short channel length. FLUID138 can be used to model the stiffening and damping effects of fluid flow in channels of micro-electromechanical systems (MEMS).

Using continuum theory (KEYOPT(1) = 0) the flow rate Q of channels with circular cross-section (KEYOPT(3) = 0) is given by the Hagen-Poiseuille equation:

$$Q = \frac{r^2}{8\eta}\ \frac{A}{l_c}\Delta P \tag{7.14}$$

Q = flow rate in units of volume/time
r = radius
l_c = channel length
A = cross-sectional area
ΔP = pressure difference along channel length

This assumption holds for low Reynolds numbers (Re < 2300), for l >> r and r >> Lm where Lm is the mean free path at the current pressure.

$$L_m(P_o) = \frac{L_o P_o}{P_o} \tag{7.15}$$

In case of rectangular cross sections (KEYOPT(3) = 1) the channel resistance depends on the aspect ratio of channel. The flow rate is defined by:

$$Q = \frac{8 r_h^2}{\eta \chi}\ \frac{A}{l_c}\Delta P \tag{7.16}$$

where:

r_h = hydraulic radius (defined below)
A = true cross-sectional area (not that corresponding to the hydraulic radius)
χ = so-called friction factor (defined below)

The hydraulic radius is defined by:

$$r_h = \frac{2A}{U} = \frac{2HW}{2(H+W)} \tag{7.17}$$

and the friction factor χ is approximated by:

$$\chi = \left[\frac{1 - 0.63n + 0.052n^5}{3}\frac{(n+1)^2}{32}\right]^{-1} \tag{7.18}$$

where:

H = height of channel
W = width of channel (must be greater than H)
n = H/W

A special treatment is necessary to consider high Knudsen numbers and short channel length (KEYOPT(1) = 1) (Sharipov([343] (p. 940))).

7.2. Slide Film Theory

Slide film damping occurs when surfaces move tangentially with respect to each other. Typical applications of slide film models are damping between fingers of a comb drive and damping between large horizontally moving plates (seismic mass) and the silicon substrate. Slide film damping can be described by a nodal force displacement relationship. FLUID139 is used to model slide film fluid behavior and assumes isothermal viscous flow.

Slide film problems are defined by:

$$\rho \frac{\partial \nu}{\partial t} = \eta \frac{\partial^2 \nu}{\partial z^2} \tag{7.19}$$

where:

P = pressure
ν = plate fluid velocity
η = dynamic viscosity
z = normal direction of the laterally moving plates
t = time

Slide film problems can be represented by a series connection of mass-damper elements with internal nodes where each damper represents the viscous shear stress between two fluid layers and each mass represents its inertial force. The damper elements are defined by:

$$C = \frac{\eta A}{d_i} \tag{7.20}$$

where:

C = damping coefficient
A = actual overlapping plate area
d_i = separation between two internal nodes (not the gap separation)

The mass of each internal node is given by:

$$M = \rho A d_i \tag{7.21}$$

where:

ρ = fluid density

In case of slip flow boundary conditions (KEYOPT(3) = 1) the fluid velocity at the moving plate is somewhat smaller than the plate velocity itself. Slip flow BC can be considered by additional damper elements which are placed outside the slide film whereby the damping coefficient must be:

$$C = \frac{\eta A}{L_m} \tag{7.22}$$

where:

L_m = mean free path length of the fluid at the current pressure

In case of second order slip flow (KEYOPT(3) = 2) the damping coefficient is:

$$C = \left[\frac{L_m}{\eta A} + \frac{d}{\eta A} \left(0.1\ Kn^{0.788} e^{-\frac{Kn}{10}} \right) \right]^{-1} \tag{7.23}$$

where Kn is defined with Equation 7.10 (p. 247)

Note that all internal nodes are placed automatically by FLUID139.

Two node models are sufficient for systems where the operating frequency is below the cut-off frequency which is defined by:

$$f_c = \frac{\eta}{2\pi\rho d^2} \tag{7.24}$$

where:

f_c = cut-off frequency
d = gap separation

In this special case, damping coefficients are almost constant, regardless of the frequency, and inertial effects are negligible. At higher frequencies, the damping ratio increases significantly up to a so-called maximum frequency, which is defined by:

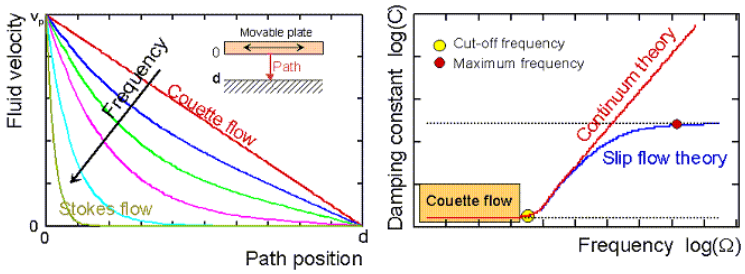
$$f_{max} = \frac{\eta}{2\pi\rho L_m^2} \tag{7.25}$$

where:

f_{max} = maximum frequency

The meaning of both numbers is illustrated below:

Figure 7.1: Flow Theory, Cut-off, and Maximum Frequency Interrelation



In case of large signal damping, the current overlapping plate are as defined by:

$$A_{new} = A_{init} + \frac{dA}{du} (u_n - u_i) \tag{7.26}$$

where:

A_{new} = actual area
A_{init} = initial area
u_i = nodal displacement in operating direction of the first interface node
u_n = nodal displacement of the second interface node

For rectangular plates which move parallel to its edge, the area change with respect to the plate displacement (dA/du) is equal to the plate width. These applications are typical for micro-electromechanical systems as comb drives where the overlapping area changes with respect to deflection.

Chapter 8: Acoustics

The following topics related to acoustics are available:

8.1. Acoustic Fundamentals

The following acoustic fluid fundamentals topics are available:

8.1.1. Governing Equations

In acoustic fluid-structural interaction (FSI) problems, the structural dynamics equation must be considered along with the Navier-Stokes equations of fluid momentum and the flow continuity equation

From the law of conservation of mass law comes the continuity equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v_x)}{\partial x} + \frac{\partial (\rho v_y)}{\partial y} + \frac{\partial (\rho v_z)}{\partial z} = 0 \tag{8.1}$$

where:

v_x, v_y and v_z = components of the velocity vector in the x, y and z directions, respectively
ρ = density
x, y, z = global Cartesian coordinates
t = time

The Navier-Stokes equations are as follows:

$$\frac{\partial \rho v_x}{\partial t} + \frac{\partial (\rho v_x v_x)}{\partial x} + \frac{\partial (\rho v_y v_x)}{\partial y} + \frac{\partial (\rho v_z v_x)}{\partial z} = \rho g_x - \frac{\partial P}{\partial x} + R_x + \frac{\partial}{\partial x}\left(\mu_e \frac{\partial v_x}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_e \frac{\partial v_x}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_e \frac{\partial v_x}{\partial z}\right) + T_x \tag{8.2}$$

$$\frac{\partial \rho v_y}{\partial t} + \frac{\partial (\rho v_x v_y)}{\partial x} + \frac{\partial (\rho v_y v_y)}{\partial y} + \frac{\partial (\rho v_z v_y)}{\partial z} = \rho g_y - \frac{\partial P}{\partial y} + R_y + \frac{\partial}{\partial x}\left(\mu_e \frac{\partial v_y}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_e \frac{\partial v_y}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_e \frac{\partial v_y}{\partial z}\right) + T_y \tag{8.3}$$

$$\frac{\partial \rho v_z}{\partial t}+\frac{\partial(\rho v_x v_z)}{\partial x}+\frac{\partial(\rho v_y v_z)}{\partial y}+\frac{\partial(\rho v_z v_z)}{\partial z}=\rho g_z-\frac{\partial P}{\partial z}$$
$$+R_z+\frac{\partial}{\partial x}\left(\mu_e\frac{\partial v_z}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu_e\frac{\partial v_z}{\partial y}\right)+\frac{\partial}{\partial z}\left(\mu_e\frac{\partial v_z}{\partial z}\right)+T_z \tag{8.4}$$

where:

g_x, g_y, g_z = components of acceleration due to gravity (input on **ACEL** command)
ρ = density
μ_e = effective viscosity (discussed below)
R_x, R_y, R_z = distributed resistances (discussed below)
T_x, T_y, T_z = viscous loss terms (discussed below)

The discretized structural dynamics equation can be formulated using the structural elements as shown in Equation 15.5 (p. 764). The fluid momentum (Navier-Stokes) equations and continuity equations are simplified to get the acoustic wave equation using the following assumptions:

1. The fluid is compressible (density changes due to pressure variations).

2. There is no mean flow of the fluid.

The acoustic wave equation is given by:

$$\nabla\cdot\left(\frac{1}{\rho_0}\nabla p\right)-\frac{1}{\rho_0 c^2}\frac{\partial^2 p}{\partial t^2}+\nabla\cdot\left[\frac{4\mu}{3\rho_0}\nabla\left(\frac{1}{\rho_0 c^2}\frac{\partial p}{\partial t}\right)\right]=-\frac{\partial}{\partial t}\left(\frac{Q}{\rho_0}\right)+\nabla\cdot\left[\frac{4\mu}{3\rho_0}\nabla\left(\frac{Q}{\rho_0}\right)\right] \tag{8.5}$$

where:

c = speed of sound $\left(\sqrt{K/\rho_o}\right)$ in fluid medium (input as SONC on the **MP** command)
ρ_o = mean fluid density (input as DENS on the **MP** command)
K = bulk modulus of fluid
μ = dynamic viscosity (input as VISC on the **MP** command)
p = acoustic pressure (=p(x, y, z, t))
Q = mass source in the continuity equation (input as JS on the **BF** command)
t = time

Since the viscous dissipation has been taken in account using the Stokes hypothesis, Equation 8.5 (p. 254) is referred to as the lossy wave equation for propagation of sound in fluids. The discretized structural Equation 15.5 (p. 764) and the lossy wave Equation 8.5 (p. 254) must be considered simultaneously in FSI problems. The wave equation will be discretized in the next subsection, followed by the derivation of the damping matrix to account for the dissipation at the FSI interface. The acoustic pressure exerting on the structure at the FSI interface will be considered in Derivation of Acoustic Matrices (p. 256) to form the coupling stiffness matrix.

Harmonically varying pressure is given by:

$$p(\vec{r},t)=\mathrm{Re}[p(\vec{r})e^{j\omega t}] \tag{8.6}$$

where:

p = amplitude of the pressure

$j = \sqrt{-1}$

$\omega = 2\pi f$

f = frequency of oscillations of the pressure

Equation 8.5 (p. 254) is reduced to the following inhomogeneous Helmholtz equation:

$$\nabla\cdot(\frac{1}{\rho_0}\nabla p)+\frac{\omega^2}{\rho_0 c^2}p+j\omega\nabla\cdot[\frac{4\mu}{3\rho_0}\nabla(\frac{1}{\rho_0 c^2}p)]=-j\omega(\frac{Q}{\rho_0})+\nabla\cdot[\frac{4\mu}{3\rho_0}\nabla(\frac{Q}{\rho_0})] \tag{8.7}$$

Note that some theories may not be applicable for 2-D acoustic elements using Equation 8.7 (p. 255). See FLUID29 and FLUID129 for available details.

8.1.2. Finite Element Formulation of the Wave Equation

The finite element formulation is obtained by testing wave Equation 8.5 (p. 254) using the Galerkin procedure (Bathe [2] (p. 921)). Equation 8.5 (p. 254) is multiplied by testing function *w* and integrated over the volume of the domain (Zienkiewicz [86] (p. 925)) with some manipulation to yield the following:

$$\begin{aligned}&\iiint_{\Omega_F}\frac{1}{\rho_0 c^2}w\frac{\partial^2 p}{\partial t^2}dv+\iiint_{\Omega_F}\nabla w\cdot(\frac{4\mu}{3\rho_0^2 c^2}\nabla\frac{\partial p}{\partial t})dv+\iiint_{\Omega_F}\nabla w\cdot(\frac{1}{\rho_0}\nabla p)dv\\&-\oiint_{\Gamma_F}w(\frac{1}{\rho_0}+\frac{4\mu}{3\rho_0^2 c^2}\frac{\partial}{\partial t})\hat{n}\cdot\nabla p ds+\oiint_{\Gamma_F}w\frac{4\mu}{3\rho_0^2}\hat{n}\cdot\nabla Q ds\\&=\iiint_{\Omega_F}w\frac{1}{\rho_0}\frac{\partial Q}{\partial t}dv+\iiint_{\Omega_F}\nabla w\cdot(\frac{4\mu}{3\rho_0^2}\nabla Q)dv\end{aligned} \tag{8.8}$$

where:

dv = volume differential of acoustic domain Ω_F

ds = surface differential of acoustic domain boundary Γ_F

$\hat{n}$ = outward normal unit vector to the boundary Γ_F

From the equation of momentum conservation, the normal velocity on the boundary of the acoustic domain is given by:

$$\frac{\partial v_{n,F}}{\partial t}=\hat{n}\cdot\frac{\partial\vec{v}}{\partial t}=-(\frac{1}{\rho_0}+\frac{4\mu}{3\rho_0^2 c^2}\frac{\partial}{\partial t})\hat{n}\cdot\nabla p+\frac{4\mu}{3\rho_0^2}\hat{n}\cdot\nabla Q \tag{8.9}$$

Substituting Equation 8.9 (p. 255) into Equation 8.8 (p. 255) yields the "weak" form of Equation 8.5 (p. 254), given by:

$$\begin{aligned}&\iiint_{\Omega_F}\frac{1}{\rho_0 c^2}w\frac{\partial^2 p}{\partial t^2}dv+\iiint_{\Omega_F}\nabla w\cdot(\frac{4\mu}{3\rho_0^2 c^2}\nabla\frac{\partial p}{\partial t})dv+\iiint_{\Omega_F}\nabla w\cdot(\frac{1}{\rho_0}\nabla p)dv\\&+\oiint_{\Gamma_F}w\frac{\partial v_{n,F}}{\partial t}ds=\iiint_{\Omega_F}w\frac{1}{\rho_0}\frac{\partial Q}{\partial t}dv+\iiint_{\Omega_F}\nabla w\cdot(\frac{4\mu}{3\rho_0^2}\nabla Q)dv\end{aligned} \tag{8.10}$$

The normal acceleration of the fluid particle can be presented using the normal displacement of the fluid particle, given by:

$$\frac{\partial v_{n,F}}{\partial t} = \hat{n} \cdot \frac{\partial^2 \bar{u}_F}{\partial t^2} \tag{8.11}$$

where:

$\bar{u}_F$ = the displacement of fluid particle

After using Equation 8.11 (p. 256), Equation 8.10 (p. 255) is expressed as:

$$\iiint_{\Omega_F} \frac{1}{\rho_0 c^2} w \frac{\partial^2 p}{\partial t^2} dv + \iiint_{\Omega_F} \nabla w \cdot (\frac{4\mu}{3\rho_0^2 c^2} \nabla \frac{\partial p}{\partial t}) dv + \iiint_{\Omega_F} \nabla w \cdot (\frac{1}{\rho_0} \nabla p) dv + \oiint_{\Gamma_F} w \hat{n} \cdot \frac{\partial^2 \bar{u}_F}{\partial t^2} ds$$
$$= \iiint_{\Omega_F} w \frac{1}{\rho_0} \frac{\partial Q}{\partial t} dv + \iiint_{\Omega_F} \nabla w \cdot (\frac{4\mu}{3\rho_0^2} \nabla Q) dv \tag{8.12}$$

8.2. Derivation of Acoustic Matrices

Equation 8.12 (p. 256) contains the fluid pressure p and the structural displacement components $u_{x,F}$, $u_{y,F}$, and $u_{z,F}$ as the dependent variables to solve. The finite element approximating shape functions for the spatial variation of the pressure and displacement components are given by:

$$P = \{N\}^T \{P_e\} \tag{8.13}$$

$$u = \{N'\}^T \{u_e\} \tag{8.14}$$

where:

{N} = element shape function for pressure
{N'} = element shape function for displacements
$\{P_e\}$ = nodal pressure vector
$\{u_e\} = \{u_{xe}\},\{u_{ye}\},\{u_{ze}\}$ = nodal displacement component vectors

From Equation 8.13 (p. 256) and Equation 8.14 (p. 256), the second time derivative of the variables and the virtual change in the pressure can be expressed as follows:

$$\frac{\partial^2 P}{\partial t^2} = \{N\}^T \{\ddot{P}_e\} \tag{8.15}$$

$$\frac{\partial^2}{\partial t^2} \{u\} = \{N'\}^T \{\ddot{u}_e\} \tag{8.16}$$

$$\delta P = \{N\}^T \{\delta P_e\} \tag{8.17}$$

After substituting Equation 8.13 (p. 256) and Equation 8.14 (p. 256) into Equation 8.12 (p. 256), the finite element statement of the wave Equation 8.5 (p. 254) is expressed as:

$$\iiint_{\Omega_F} \frac{1}{\rho_0 c^2} \{N\}\{N\}^T dv \{\ddot{p}_e\} + \iiint_{\Omega_F} \frac{4\mu}{3\rho_0^2 c^2} [\nabla N]^T [\nabla N] dv \{\dot{p}_e\} + \iiint_{\Omega_F} \frac{1}{\rho_0} [\nabla N]^T [\nabla N] dv \{p_e\}$$
$$+ \oiint_{\Gamma_F} \{N\}\{n\}^T \{N'\}^T ds \{\ddot{u}_{e,F}\} = \iiint_{\Omega_F} \frac{1}{\rho_0} \{N\}\{N\}^T dv \{\dot{q}\} + \iiint_{\Omega_F} \frac{4\mu}{3\rho_0^2} [\nabla N]^T [\nabla N] dv \{q\} \tag{8.18}$$

where:

{n} = outward normal vector at the fluid boundary
{q} = nodal mass source vector
$\{\dot{q}\}$ = the first time derivative of nodal mass source vector

Other terms are defined in Acoustic Fundamentals (p. 253). Equation 8.18 (p. 256) can be written in matrix notation to create the following discretized wave equation:

$$[M_F]\{\ddot{p}_e\} + [C_F]\{\dot{p}_e\} + [K_F]\{p_e\} + \bar{\rho}_0[R]^T\{\ddot{u}_{e,F}\} = \{f_F\} \tag{8.19}$$

where:

$[M_F] = \bar{\rho}_0 \iiint_{\Omega_F} \frac{1}{\rho_0 c^2}\{N\}\{N\}^T dv$ = acoustic fluid mass matrix

$[C_F] = \bar{\rho}_0 \iiint_{\Omega_F} \frac{4\mu}{3\rho_0^2 c^2}[\nabla N]^T[\nabla N]dv$ = acoustic fluid damping matrix

$[K_F] = \bar{\rho}_0 \iiint_{\Omega_F} \frac{1}{\rho_0}[\nabla N]^T[\nabla N]dv$ = acoustic fluid stiffness matrix

$[R]^T = \oiint_{\Gamma_F} \{N\}\{n\}^T\{N'\}^T ds$ = acoustic fluid boundary matrix

$\{f_F\} = \bar{\rho}_0 \iiint_{\Omega_F} \frac{1}{\rho_0}\{N\}\{N\}^T dv\{\dot{q}\} + \bar{\rho}_0 \iiint_{\Omega_F} \frac{4\mu}{3\rho_0^2}[\nabla N]^T[\nabla N]dv\{q\}$ = acoustic fluid load vector

$\bar{\rho}_0$ = acoustic fluid mass density constant

8.3. Propagation, Radiation, and Scattering of Acoustic Pressure Waves

The following topics are available are available in this section:
8.3.1. Acoustic Boundary Conditions
8.3.2. Absorbing Boundary Condition (ABC)
8.3.3. Perfectly Matched Layers (PML)
8.3.4. Acoustic Excitation Sources
8.3.5. Sophisticated Acoustic Media

8.3.1. Acoustic Boundary Conditions

For Dirichlet or Neumann boundary conditions, i.e., on a sound-soft boundary with $p = p_0$ (input as PRES on the **D** command) or on a sound-hard boundary with $\hat{n} \cdot \nabla p = 0$, the surface integration will be eliminated.

The Robin boundary condition on impedance boundary Γ_Z is given by:

$$v_{n,F}(\vec{r}) - v_{n,S}(\vec{r}) = Y(\vec{r})p(\vec{r}) \tag{8.20}$$

where:

$v_{n,F}$ = normal velocity of fluid particle on the boundary
$v_{n,S}$ = normal velocity of structure surface
Y = boundary admittance
Z (boundary impedance) = 1/Y (Z is input as IMPD on the **SF** command)

Substituting Equation 8.20 (p. 257) into Equation 8.10 (p. 255) yields:

$$\iint_{\Gamma_Z} w \frac{\partial v_{n,F}}{\partial t} ds = \iint_{\Gamma_Z} w(Y \frac{\partial p}{\partial t} + \frac{\partial v_{n,S}}{\partial t}) ds \tag{8.21}$$

For simplicity, the inviscid fluid will be investigated. The matrix forms in Equation 8.19 (p. 257) are rewritten as:

$$[C_F] = \bar{\rho}_0 \iint_{\Gamma_Z} Y\{N\}\{N\}^T ds \tag{8.22}$$

$$\{f_F\} = -\bar{\rho}_0 \iint_{\Gamma_Z} \{N\}\{N\}^T ds\{\dot{v}_{n,S}\} + \bar{\rho}_0 \iiint_{\Omega_F} \frac{1}{\rho_0}\{N\}\{N\}^T dv\{\dot{q}\} \tag{8.23}$$

In acoustic design, the attenuation coefficient (i.e., the absorption coefficient) is used to define the absorption properties of the material. The attenuation coefficient is the ratio of the absorbed sound power density to the incident sound power density. The ratio is expressed as follows:

$$\alpha = \frac{I_a}{I_{inc}} \tag{8.24}$$

where:

$$I_a = \frac{1}{2} p_a (\hat{n} \cdot \bar{v}_a^*) = \text{absorbed sound power density}$$

$$I_{inc} = \frac{1}{2} p_{inc} (\hat{n} \cdot \bar{v}_{inc}^*) = \text{incident sound power density}$$

Because the module of reflection coefficient is given as $|R| = \sqrt{I_{ref} / I_{inc}} = \sqrt{1-\alpha} \quad (0 \le |R| \le 1)$, the equivalent impedance (assumed Im(z)=0) of the attenuating material is expressed as:

$$z = z_0 \frac{1+\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}} \tag{8.25}$$

where:

$z_0 = \rho_0 c_0$ = sound impedance

The impedance value of the Robin boundary condition can be defined by the sound impedance $z_0 = \rho_0 c_0$ (input as INF on the **SF** command), the attenuation coefficient (input as CONV on the **SF** command), or the general complex impedance (admittance the for modal analysis) (input as IMPD on the **SF** command). The Robin boundary condition can be used for the harmonic response or modal analysis.

8.3.2. Absorbing Boundary Condition (ABC)

Exterior structural acoustics problems typically involve a structure submerged in an infinite, homogeneous, inviscid fluid. For a source-free and inviscid fluid, Equation 8.5 (p. 254) is simplified as follows:

$$\nabla \cdot \left(\frac{1}{\rho_0}\nabla p\right) = \frac{1}{\rho_0 c^2}\frac{\partial^2 p}{\partial t^2} \qquad \text{in } \Omega^+ \tag{8.26}$$

where:

Ω^+ = unbounded region occupied by the fluid

The pressure wave must satisfy the Sommerfeld radiation condition (which states that the waves generated within the fluid are outgoing) at infinity. This is expressed as:

$$\lim_{r\to\infty} r^{(d-1)/2}\left(\frac{\partial p}{\partial r} + \frac{1}{c}\dot{p}\right) = 0 \tag{8.27}$$

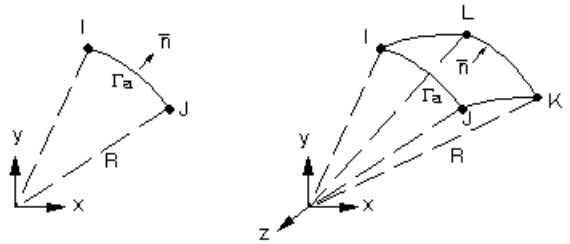
where:

r = distance from the origin

d = dimensionality of the problem (i.e., d =3 or d =2 if Ω^+ is 3-D or 2-D, respectively)

A primary difficulty encountered when using finite elements to model an infinite medium arises from the need to satisfy the Sommerfeld radiation condition, as expressed in Equation 8.27 (p. 259). A typical approach to this problem involves truncating the unbounded domain Ω^+ by introducing an absorbing (artificial) boundary Γ_a at some distance from the structure (Equation 8.5 (p. 254)).

Figure 8.1: Absorbing Boundary



The homogeneous Helmholtz equation (Equation 8.26 (p. 259)) is then solved in the annular region Ω^+, which is bounded by the absorbing boundary Γ_a. For the resulting problem in Ω^+ to be well-posed, an appropriate condition must be specified on Γ_a. To this end, the following second-order conditions are used on Γ_a (Kallivokas et al. [218] (p. 933)):

In two dimensions:

$$P_n + \gamma P_n = -\frac{1}{c}\ddot{P} + \left(\frac{1}{2}\kappa - \frac{\gamma}{c}\right)P + \frac{1}{2}cP\lambda\lambda + \left(\frac{1}{8}\kappa^2 c + \frac{1}{2}\kappa\gamma\right)P \tag{8.28}$$

where:

n = outward normal to Γ_a

P_n = pressure derivative in the normal direction

$P_{\lambda\lambda}$ = pressure derivative along Γ_a
k = curvature of Γ_a
γ = stability parameter

In three dimensions:

$$\dot{P}_n + \gamma P_n = -\frac{1}{c}\ddot{P} + \left(H - \frac{\gamma}{c}\right)\dot{P} + H\gamma P + \frac{c}{2\sqrt{EG}}\left[\left(\sqrt{\frac{G}{E}}P_u\right)_u + \left(\sqrt{\frac{G}{E}}P_v\right)_v\right] + \frac{c}{2}\left(H^2 - K\right)P \tag{8.29}$$

where:

n = outward normal
u and v = orthogonal curvilinear surface coordinates (e.g., the meridional and polar angles in spherical coordinates)
P_u, P_v = pressure derivatives in the Γ_a surface directions
H and K = mean and Gaussian curvature, respectively
E and G = usual coefficients of the first fundamental form

Following a Galerkin based procedure, Equation 8.26 (p. 259) is multiplied by testing function *w* and integrated over the annular domain Ω^f. By using the divergence theorem on the resulting equation it can be shown that:

$$\iiint_{\Omega_F} \frac{1}{\rho_0 c^2} w \frac{\partial^2 p}{\partial t^2} dv + \iiint_{\Omega_F} \nabla w \cdot \left(\frac{1}{\rho_0}\nabla p\right) dv - \iint_{\Gamma_a} w\left(\frac{1}{\rho_0}\hat{n}\cdot\nabla p\right) ds - \iint_{\Gamma_F} w\left(\frac{1}{\rho_0}\hat{n}\cdot\nabla p\right) ds = 0 \tag{8.30}$$

Upon discretization of Equation 8.30 (p. 260), the first term on the left hand side will yield the mass matrix of the fluid, while the second term will yield the stiffness matrix. The following finite element approximations for quantities on the absorbing boundary Γ_a placed at a radius R are introduced:

$$p(x,t) = \{N\}^T\{p_e\}, \qquad q^{(1)}(x,t) = \{N\}^T\{q_e^{(1)}\}, \qquad q^{(2)}(x,t) = \{N\}^T\{q_e^{(2)}\} \tag{8.31}$$

where:

{N} = vectors of shape functions
p, $q^{(1)}$, $q^{(2)}$ = unknown nodal values (p is output as degree of freedom PRES. $q^{(1)}$ and $q^{(2)}$ are solved for but not output.)

Taking the nodal shape function as the testing function, the element stiffness and damping matrices can be further reduced. The two dimensional case is reduced to:

$$[K^{2D}] = \frac{1}{8R}\begin{bmatrix} 4[M_L] & 4R^2[M_L] & -[M_L] \\ 4R^2[M_L] & -4R^2[M_L] & 0 \\ -[M_L] & 0 & [M_L] \end{bmatrix} \tag{8.32}$$

$$[C^{2D}] = \frac{1}{8c}\begin{bmatrix} 8[M_L] & 0 & 0 \\ 0 & 4R^2[M_L] & 0 \\ 0 & 0 & [M_L] \end{bmatrix} \tag{8.33}$$

where:

$$[M_L] = \int_{\Gamma_L} \{N\}\{N\}^T dl$$

dl = arc-length differential

These matrices are 6 x 6 in size, having 2 nodes per element with 3 degrees of freedom per node (p, $q^{(1)}$, $q^{(2)}$). (See FLUID129 for details.)

The three dimensional case is reduced to:

$$[K^{3D}] = \frac{1}{2R}\begin{bmatrix} 2[M_S] & R^2[K_S] \\ R^2[K_S] & -R^2[K_S] \end{bmatrix} \tag{8.34}$$

$$[C^{3D}] = \frac{1}{2c}\begin{bmatrix} 2[M_S] & 0 \\ 0 & -R^2[K_S] \end{bmatrix} \tag{8.35}$$

where:

$$[M_S] = \frac{\bar{\rho}_0}{\rho_0}\iint_{\Gamma_S} \{N\}\{N\}^T d\hat{s}$$

$$[K_S] = \frac{\bar{\rho}_0}{\rho_0}\iint_{\Gamma_S} \{\nabla^S N\}^T \{\nabla^S N\} ds$$

∇^S = surface differential operator

ds = area differential

For the three dimensional case, the surface element can be a quadratic or triangle element with corner and mid-nodes that have 2 degrees of freedom per node (p, q) (Barry et al. [217] (p. 933)). (See FLUID130 for details.)

For the axisymmetric case:

$$[K_a^{2D}] = \frac{\pi}{R}\begin{bmatrix} 2[M_A] & R^2[M_A] \\ R^2[M_A] & -R^2[M_A] \end{bmatrix} \tag{8.36}$$

$$[C_a^{2D}] = \frac{\pi}{c}\begin{bmatrix} 2[M_A] & 0 \\ 0 & -R^2[M_A] \end{bmatrix} \tag{8.37}$$

where:

$$[M_A] = \int_{\Gamma_L} \{N\}\{N\}^T r dl$$

r = radius

These matrices are 4x4 in size with 2 nodes per element and 2 degrees of freedom per node (p, q).

The local absorbing boundary conditions in this section are applicable to modal, harmonic response, or transient analyses.

8.3.3. Perfectly Matched Layers (PML)

Perfectly matched layers are artificial ansiotropic materials that absorb all incoming waves without any reflections, except for the gazing wave that parallels the PML interface in the propagation direction. PMLs are currently constructed by the propagation of acoustic waves in the media for harmonic response analysis (Zhao [406] (p. 944)).

In the frequency domain, the governing lossless and source-free momentum and conservation are given by:

$$\nabla p = -j\omega\rho_0\vec{v} \tag{8.38}$$

$$\nabla \cdot \vec{v} = -\frac{1}{\rho_0 c^2} j\omega p \tag{8.39}$$

The anisotropic PML material is assumed as follows:

$$\bar{\bar{\rho}} = \rho_0[\Lambda] \tag{8.40}$$

$$\bar{\bar{\zeta}} = \frac{1}{\rho_0 c^2}[\Lambda] \tag{8.41}$$

where:

$[\Lambda]$ = diagonal tensor diag{s_x, s_y, s_z}, and s_x, s_y, s_z are complex constants.

In order to satisfy the vector operation, the governing acoustic equations for PMLs are written as:

$$\nabla p = -j\omega\bar{\bar{\rho}} \cdot \vec{v} \tag{8.42}$$

$$\nabla \cdot (\bar{\bar{\zeta}}^{-1} \cdot \vec{v}) = -j\omega p \tag{8.43}$$

If the modified operator ∇_s is expressed as:

$$\nabla_S = \frac{1}{s_x}\frac{\partial}{\partial x}\hat{x} + \frac{1}{s_y}\frac{\partial}{\partial y}\hat{y} + \frac{1}{s_z}\frac{\partial}{\partial z}\hat{z} \tag{8.44}$$

Then the wave equation for a PML with a modified differential operator is expressed as:

$$\nabla_S \cdot (\frac{1}{\rho_0}\nabla_S p) + \frac{\omega^2}{\rho_0 c^2} p = 0 \tag{8.45}$$

The interface between the PML and the real acoustic medium can be rendered reflectionless for all frequencies and angles of incidence of a wave propagating from the acoustic medium to the PML if the elements of $[\Lambda]$ are defined as follows:

$$s_i = 1 + \frac{\sigma_i}{j\omega} \quad (i = 1, 2, 3) \tag{8.46}$$

where:

σ_i = attenuation constant

Multiplying $s_x s_y s_z$ with Equation 8.45 (p. 262) results in the final wave equation needed to construct the PML:

$$\nabla \cdot (\frac{1}{\rho_0}[\Psi] \cdot \nabla p) + \frac{\omega^2}{\rho_0 c^2} s_x s_y s_z p = 0 \tag{8.47}$$

where:

$[\Psi]$ = diagonal tensor diag$\{s_y s_z/s_x, s_z s_x/s_y, s_x s_y/s_z\}$

By using the Galerkin method and setting the pressure to zero (input as PRES on the **D** command) on the backed boundary, the "weak" formulation of an acoustic pressure wave in PML is expressed as:

$$\iiint_{\Omega_{PML}} \{\nabla w \cdot (\frac{1}{\rho_0}[\Psi] \cdot \nabla p) - \frac{\omega^2}{\rho_0 c^2} s_x s_y s_z w p\} dv = 0 \tag{8.48}$$

Equation 8.48 (p. 263) can be written in matrix notation to produce the following discretized wave equation:

$$\{-\omega^2 [M_{PML}] + j\omega [C_{PML}] j + [K_{PML}]\}\{p_e\} = 0 \tag{8.49}$$

where:

$$[M_{PML}] = \frac{\bar{\rho}_0}{\rho_0} \iiint_{\Omega_{PML}} \frac{S_{Re}}{c^2} \{N\}\{N\}^T dv = \text{ PML mass matrix}$$

$$[C_{PML}] = \frac{\bar{\rho}_0}{\omega \rho_0} \iiint_{\Omega_{PML}} [\nabla N]^T ([\Psi_{Im}][\nabla N]) dv - \frac{\omega \bar{\rho}_0}{\rho_0} \iiint_{\Omega_{PML}} \frac{S_{Im}}{c^2} \{N\}\{N\}^T dv = \text{ PML damping matrix}$$

$$[K_{PML}] = \frac{\bar{\rho}_0}{\rho_0} \iiint_{\Omega_{PML}} [\nabla N]^T ([\Psi_{Re}][\nabla N]) dv = \text{ PML stiffness matrix}$$

S_{Re} = real part of complex constant $s_x s_y s_z$
S_{Im} = imaginary part of complex constant $s_x s_y s_z$
$[\Psi_{Re}]$ = real part of complex tensor $[\Psi]$
$[\Psi_{Im}]$ = imaginary part of complex tensor $[\Psi]$

The PML material is defined using FLUID30, FLUID220, and FLUID221 elements with KEYOPT(4) = 1. Since the PML material is constructed in Cartesian coordinates, the edges of the 3-D PML region must be aligned with the axes of the global Cartesian coordinate system. More than one 1-D PML region may exist in a finite element model. The element coordinate system (**ESYS** command) uniquely identifies each PML region. A parabolic attenuation distribution is used to minimize numerical reflections in the PML. ANSYS, Inc. recommends using at least three PML element layers to obtain satisfactory accuracy. Some buffer elements between the PML region and objects should be utilized. Since a PML acts as an infinite open domain, any boundary conditions and material properties need to be carried over into the PML region. The pressure must be set to zero (input as PRES on the **D** command) on the exterior surface of the PML, excluding Neumann boundaries (symmetric planes in the model). Any kinds of excitation sources are prohibited in the PML region. Refer to Perfectly Matched Layers (PML) in the *Mechanical APDL High-Frequency Electromagnetic Analysis Guide* for more information about using PMLs.

8.3.4. Acoustic Excitation Sources

Sound excitation sources are a fundamental part of an acoustic analysis. Various excitation sources, such as a monopole or plane wave can be imposed in simulations. The following methods can be used to introduce sound excitation sources:

- Specified pressure or normal velocity on the domain boundary
- Mass source in the wave Equation 8.5 (p. 254)
- Analytic wave sources:
  - Incident pressure applied on the exterior boundary (Robin boundary condition)
  - Incident pressure applied on the PML interface
  - Solve the wave equation with pure scattered pressure as a variable (Pure Scattered Pressure Formulation (p. 280))

8.3.4.1. Specified Pressure or Normal Velocity on the Domain Boundary

Specified pressure (input as PRES on the **D** command) or normal velocity on the domain boundary offers a straightforward method to introduce the excitation source for harmonic response analysis (normal acceleration for transient analysis) (input as SHLD on the **SF** command) or the velocity for harmonic response analysis (acceleration for transient analysis) (input as EF on the **BF** command) (Equation 8.10 (p. 255)) .

8.3.4.2. Mass Source in the Wave Equation

Mass source in the wave Equation 8.5 (p. 254) (input as JS on the **BF** command) can be used to introduce excitations for a pressure wave.

A point mass source is given by:

$$Q(x,y,z) = Q_p \delta(x - x_s)\delta(y - y_s)\delta(z - z_s) \tag{8.50}$$

where:

Q_p = magnitude of point mass source with unit [kg/s]
(xs, ys, zs) = source origin
δ = Dirac delta function

A line mass source is given by:

$$Q(x,y,z) = Q_l(z)\delta(x - x_s)\delta(y - y_s) \tag{8.51}$$

where:

Q_l = magnitude of line mass source with unit [kg/(ms)]

A surface mass source is given by:

$$Q(x,y,z) = Q_s(x,y)\delta(z - z_s) \tag{8.52}$$

where:

Q_s = magnitude of surface mass source with unit [kg/(m^2s)]

The volumetric mass source Q in Equation 8.8 (p. 255) has the unit [kg/(m^3s)].

Depending on how the load is applied on the finite element, the program will interpret the load as a point, line, surface, or volumetric mass source. For a transient analysis, the mass source rate ($\partial Q / \partial t$) in Equation 8.8 (p. 255) should be defined.

8.3.4.3. Analytic Wave Sources

The various incident wave sources are defined using the **AWAVE** command for harmonic response analysis.

A time-harmonic incident planar wave (input as PLAN on the **AWAVE** command) is given by:

$$p_{inc} = p_{inc,0} e^{j\phi} e^{-j\vec{k}_i \cdot (\vec{r} - \vec{r}_s)} = \bar{p}_{inc,0} e^{-j\vec{k}_i \cdot (\vec{r} - \vec{r}_s)} \tag{8.53}$$

where:

$P_{inc,0}$= magnitude of incident planar wave

ϕ = initial phase angle of incident planar wave (usually ignored)

$\vec{r}_i$ = source origin (usually ignored)

$\vec{k}_i$ = wave vector of incident planar wave

The particle velocity is given by:

$$v_{n,f} = \frac{1}{\omega \rho_0} \hat{n} \cdot \vec{k}_i p_{inc} \tag{8.54}$$

A time-harmonic radiated monopole pressure (input as MONO on the **AWAVE** command) is introduced by:

$$p = \bar{p}_0 \frac{e^{-j\vec{k} \cdot (\vec{r} - \vec{r}_s)}}{|\vec{r} - \vec{r}_s|} \tag{8.55}$$

where:

$\vec{k} = k\hat{R}$ = wave vector

$\hat{R} = (\vec{r} - \vec{r}_s)/|\vec{r} - \vec{r}_s|$ = radial unit vector

k = wave number

If the normal velocity is along the $\hat{R}$ direction, the velocity of particle is given by:

$$v_{n,f} = -j \frac{1}{\omega \rho_0} \left(jk + \frac{1}{|\vec{r} - \vec{r}_s|}\right) \bar{p}_0 \frac{e^{-j\vec{k} \cdot (\vec{r} - \vec{r}_s)}}{|\vec{r} - \vec{r}_s|} \tag{8.56}$$

Assuming that the monopole is a pulsating sphere (r=a) with surface pressure p_a, the pressure can be expressed as:

$$p = ap_a e^{jka} \frac{e^{-j\vec{k}\cdot(\vec{r}-\vec{r}_s)}}{|\vec{r}-\vec{r}_s|} \tag{8.57}$$

If the normal velocity on the pulsating sphere surface is equal to v_a, the pressure is expressed as:

$$p = \frac{v_a \rho_0 ace^{jka} ka}{ka - j} \frac{e^{-j\vec{k}\cdot(\vec{r}-\vec{r}_s)}}{|\vec{r}-\vec{r}_s|} \tag{8.58}$$

Other applicable incident sources (Whitaker [[408] (p. 944)]) are also defined by the **AWAVE** command.

To apply the incident wave to the model, the splitting pressure should be introduced on the exterior excitation surface when the total pressure is solved by FEM. The total pressure consists of the incident pressure p^{inc} and scattered pressure p^{sc}. This can be expressed as:

$$p = p^{inc} + p^{sc} \tag{8.59}$$

For a planar wave, the incident wave and scattered wave can be written respectively as:

$$p^{inc} = p_0^{inc} e^{-j\vec{k}_i\cdot(\vec{r}-\vec{r}_s)} \tag{8.60}$$

$$p^{sc} = p_0^{sc} e^{-j\vec{k}_s\cdot(\vec{r}-\vec{r}_s)} \tag{8.61}$$

where:

$\vec{k}_s$ = scattered wave vector

Assuming the normal scattered wave vector is in the opposite direction of the normal incident wave vector, this can be expressed as:

$$\hat{n}\cdot\vec{k}_s = -\hat{n}\cdot\vec{k}_i \tag{8.62}$$

After substituting Equation 8.54 (p. 265) into the boundary integration of Equation 8.8 (p. 255), this can be expressed as:

$$\iint_{\Gamma_{ext}} \frac{1}{\rho_0}\nabla p\cdot\hat{n}ds = \iint_{\Gamma_{ext}} \frac{1}{\rho_0}\nabla(p^{inc}+p^{sc})\cdot\hat{n}ds = j\omega \iint_{\Gamma_{ext}} \frac{\hat{n}\cdot\hat{k}_i}{\rho_0 c} wpds - j2\omega \iint_{\Gamma_{ext}} \frac{\hat{n}\cdot\hat{k}_i}{\rho_0 c} wp^{inc}ds \tag{8.63}$$

where:

$\hat{k}_i$ = unit incident wave vector

The damping matrix and source vector in Equation 8.19 (p. 257) for lossless and source-free fluid with the planar incident wave on the exterior boundary is given by:

$$[C_F] = -\iint_{\Gamma_{ext}} \frac{\hat{n}\cdot\hat{k}_i}{\rho_0 c}\{N\}\{N\}^T ds \tag{8.64}$$

$$[f_F] = -j2\omega \iint_{\Gamma_{ext}} \frac{\hat{n}\cdot\hat{k}_i}{\rho_0 c}\{N\}p^{inc}ds \tag{8.65}$$

A similar procedure can be used for a spherical incident wave, but the scattered wave is assumed to be a planar wave in the $-\hat{k}_i$ direction. The damping matrix and source vector in Equation 8.19 (p. 257) for lossless and source-free fluid with the spherical incident wave on the exterior boundary is given by:

$$[C_F] = -\iint_{\Gamma_{ext}} \frac{\hat{n}\cdot\hat{R}}{\rho_0 c}\{N\}\{N\}^T ds \tag{8.66}$$

$$[f_F] = -j\omega \iint_{\Gamma_{ext}} \left(\frac{2}{\rho_0 c} + \frac{1}{j\omega\rho_0 |\bar{r} - \bar{r}_s|}\right)\{N\}p^{inc}\hat{n}\cdot\hat{R}ds \tag{8.67}$$

If we assume that the incident wave perpendicularly projects on the transparent exterior surface of the model, the analogous derivation can be achieved with the Robin boundary condition (Equation 8.20 (p. 257)), using the wave admittance Y_0 as the impedance boundary. Since the incoming incident wave is propagating into the model, the Robin boundary condition is expressed as:

$$\hat{n}'\cdot\bar{v}_f^{inc} = Y_0 p^{inc} \tag{8.68}$$

where:

$\hat{n}'$ = inward unit normal vector of boundary

Assuming that the scattered wave has the same wave admittance as the incident wave, the Robin boundary condition for the outgoing scattered wave is expressed as:

$$\hat{n}\cdot\bar{v}_f^{sc} = Y_0 p^{sc} \tag{8.69}$$

where:

$\hat{n}$ = outward unit normal vector of boundary

Substituting Equation 8.68 (p. 267) into the boundary integration of Equation 8.8 (p. 255) and utilizing Equation 8.69 (p. 267) leads to:

$$\iint_{\Gamma_{ext}} \frac{1}{\rho_0}\nabla p\cdot\hat{n}ds = -j\omega\iint_{\Gamma_{ext}} Y_0 wpds + j2\omega\iint_{\Gamma_{ext}} wv_n^{inc}ds \tag{8.70}$$

The damping matrix and source vector in Equation 8.19 (p. 257) for lossless and source-free fluid with the incident excitation on the impedance boundary is expressed as:

$$[C_F] = \iint_{\Gamma_{ext}} Y_0\{N\}\{N\}^T ds \tag{8.71}$$

$$[f_F] = j2\omega\iint_{\Gamma_{ext}} \{N\}\{N\}^T ds\{v_n^{inc}\} \tag{8.72}$$

The above equations are useful for the analysis of the wave scattering and the pipe structure, respectively.

The planar scattered pressure wave propagating along the opposite direction of the incident wave is assumed to be on the exterior surface of the model, while the incident wave projects into the simulation domain from outside of the model. Obviously, this assumption leads to a numerical error once the non-planar scattered wave occurs. Hence, the PML is used to absorb the arbitrary scattered outgoing waves.

Assuming that the PML region is a scattered pressure region and that other regions are total pressure regions, the "weak" forms for the total pressure and the scattered pressure are respectively written as:

$$\iiint_{\Omega_F} \{\nabla w \cdot (\frac{1}{\rho_0} \nabla p^{tot} - \frac{\omega^2}{\rho_0 c^2} w p^{tot}\} dv - \oiint_{\Gamma_S} \frac{1}{\rho_0} w \hat{n} \cdot \nabla p^{tot} ds = 0 \tag{8.73}$$

In PML scattered pressure region:

$$\iiint_{\Omega_{PML}} \{\nabla w \cdot (\frac{1}{\rho_0} [\Psi] \cdot \nabla p^{sc}) - \frac{\omega^2}{\rho_0 c^2} s_x s_y s_z w p^{sc}\} dv = 0 \tag{8.74}$$

Total pressure can be split into scattered pressure and incident pressure on the interface between the general acoustic region and the PML region. This can be expressed as:

$$p_I^{tot} = p_I^{inc} + p_I^{sc} \tag{8.75}$$

The degrees of freedom on nodes with PML attributes are scattered pressure. The surface integration of scattered pressure on the interface will be cancelled out in Equation 8.73 (p. 268) and Equation 8.74 (p. 268). Assuming that the inside model is source free, the general matrix notation in the normal acoustic region, including the pressure on interface nodes, can be written as:

$$\{-\omega^2 [M_F] + j\omega [C_F] + [K_F]\}\{p\} = 0 \tag{8.76}$$

Substituting Equation 8.75 (p. 268) into Equation 8.76 (p. 268) leads to the matrix equation for the interior total pressure and the interface scattered pressure, expressed as:

$$\begin{aligned} &\left(-\omega^2 \begin{bmatrix} M_{tt} & M_{tI} \\ M_{It} & M_{II} \end{bmatrix} + j\omega \begin{bmatrix} C_{tt} & C_{tI} \\ C_{It} & C_{II} \end{bmatrix} + \begin{bmatrix} K_{tt} & K_{tI} \\ K_{It} & K_{II} \end{bmatrix}\right) \begin{Bmatrix} p \\ p_I^{sc} \end{Bmatrix} \\ &= -\left(-\omega^2 \begin{bmatrix} M_{tt} & M_{tI} \\ M_{It} & M_{II} \end{bmatrix} + j\omega \begin{bmatrix} C_{tt} & C_{tI} \\ C_{It} & C_{II} \end{bmatrix} + \begin{bmatrix} K_{tt} & K_{tI} \\ K_{It} & K_{II} \end{bmatrix}\right) \begin{Bmatrix} 0 \\ p_I^{inc} \end{Bmatrix} \end{aligned} \tag{8.77}$$

The matrix equation for whole domain is expressed as:

$$\begin{pmatrix} \bar{K}_{tt} & \bar{K}_{tI} & 0 \\ \bar{K}_{It} & \bar{K}_{II} & 0 \\ 0 & 0 & \bar{K}_{ss} \end{pmatrix} \begin{Bmatrix} p \\ p_I^{sc} \\ p^{sc} \end{Bmatrix} = \{f^{inc}\} \tag{8.78}$$

where:

$\bar{K}_{tt}$ = complex stiffness matrix for interior total pressure from Equation 8.77 (p. 268)

$\bar{K}_{tI}, \bar{K}_{It}$ = complex stiffness matrix on interface from Equation 8.77 (p. 268)

$\bar{K}_{II}$ = complex stiffness matrix for interface scattered pressure from Equation 8.77 (p. 268)

$\bar{K}_{ss}$ = complex stiffness matrix for PML interior scattered pressure from Equation 8.49 (p. 263)

f^{inc} = complex source vector from Equation 8.77 (p. 268)

8.3.5. Sophisticated Acoustic Media

The following topics are available in this section:

8.3.5.1. Non-uniform Acoustic Media

In non-uniform acoustic media the mass density and sound speed vary with the spatial position. Wave Equation 8.5 (p. 254) in lossless media is rewritten as:

$$\nabla\cdot\left(\frac{1}{\rho_0(\vec{r})}\nabla p\right)-\frac{1}{\rho_0(\vec{r})c^2(\vec{r})}\frac{\partial^2 p}{\partial t^2}=-\frac{\partial}{\partial t}\left(\frac{Q}{\rho_0(\vec{r})}\right) \tag{8.79}$$

According to the ideal gas law, the equation of state and the speed of sound in an ideal gas are respectively expressed as:

$$c^2(\vec{r})=\gamma RT(\vec{r}) \tag{8.80}$$

$$P_{state}(\vec{r})=\rho(\vec{r})RT(\vec{r}) \tag{8.81}$$

where:

$$\gamma=\frac{C_P}{C_V}=\text{ specific heat ratio}$$

C_P = heat coefficient at a constant pressure per unit mass (input as C on the **MP** command)
C_V = heat coefficient at a constant volume per unit mass (input as CVH on the **MP** command)
R = gas constant
T = temperature
P_{state} = the absolute pressure of the gas measured in atmospheres

Assuming density ρ_0 (input as DENS on the **MP** command) and sound speed c_0 (input as SONC on the **MP** command) at the reference temperature T_0 (input on **TREF** command), and the reference static pressure (input as P_{sref} on **R** command) the density and sound speed in media are cast as follows:

$$c(\vec{r})=c_0\sqrt{\frac{T(\vec{r})}{T_0}} \tag{8.82}$$

$$\rho(\vec{r})=\frac{P_{state}(\vec{r})}{T(\vec{r})}\frac{\rho_0 T_0}{P_{state,0}} \tag{8.83}$$

The spatial temperature is input as TEMP on **BF** command and the nodal static pressure is input as CHRGD on the **BF** command.

8.3.5.2. Equivalent Fluid of Perforated Materials

8.3.5.2.1. Johnson-Chapoux-Allard Model

Assuming that the skeleton of the perforated material is rigid, the perforated material can be approximated using the Johnson-Champoux-Allard equivalent fluid model, which uses the complex effective density and velocity (Allard [407] (p. 944)). The wave equation with complex effective density and velocity is given by:

$$\nabla \cdot (\frac{1}{\rho_{eff}} \nabla p_a) + \frac{\omega^2}{\rho_{eff} c_{eff}^2} p_a = 0 \tag{8.84}$$

The effective density is given by:

$$\rho = \frac{\sigma\varphi}{j\omega}\left[1 + \frac{j4\alpha_\infty^2 \eta\omega\rho_0}{\varphi^2 \Lambda^2 \sigma^2}\right]^{1/2} + \rho_0 \alpha_\infty \tag{8.85}$$

where:

ω = angular frequency
σ = fluid resistivity
Φ = porosity
α_∞ = tortuosity
Λ = viscous characteristic length
ρ_0 = density of fluid
η = dynamic (shear) viscosity

The effective bulk modulus is given by:

$$K(\omega) = \frac{P_0 \gamma}{\gamma - (\gamma - 1)\left[\frac{8\eta}{j\omega\rho_0 P_{rt} \Lambda'^2}\left(1 + \frac{\Lambda'^2}{16}\frac{j\omega\rho_0 P_{rt}}{\eta}\right)^{1/2} + 1\right]^{-1}} \tag{8.86}$$

where:

γ = specific heat ratio
P_0 = static reference pressure (input as PSREF on the **R** command)
P_{rt}=$C_p\eta/\kappa$ = Prandtl number
C_p = specific heat
κ = thermal conductivity
Λ' = thermal characteristic length

The material coefficients used in the Johnson-Champoux-Allard model are input with the **TBDATA** command for the **TB**,PERF material as well as through **MP** commands.

The complex effective velocity is given by:

$$c_{eff} = \sqrt{K/\rho_{eff}} \tag{8.87}$$

The sound impedance is given by:

$$Z_c = \sqrt{\rho_{eff} K} = \rho_{eff} c_{eff} \tag{8.88}$$

The "weak" form of the Johnson-Champoux-Allard equivalent fluid model is given by:

$$-\omega^2 \iiint_{\Omega_F} \frac{1}{\rho_{eff} c_{eff}^2} w p_a dv + \iiint_{\Omega_F} \nabla w \cdot (\frac{1}{\rho_{eff}} \nabla p_a) dv - \oiint_{\Gamma_F} w \frac{1}{\rho_{eff}} \nabla p_a \cdot d\vec{s} = 0 \tag{8.89}$$

The matrix notation is expressed as:

$$\{-\omega^2[\bar{M}_F] + j\omega[\bar{C}_F] + [\bar{K}_F]\}\{p\} = \{f_F\} \tag{8.90}$$

where:

$$[\bar{M}_F] = \iiint_{\Omega_F} \frac{1}{\rho_{eff} c_{eff}^2}\{N\}\{N\}^T dv = [\bar{M}_r] + j[\bar{M}_i]$$

$$[\bar{K}_F] = \iiint_{\Omega_F} \frac{1}{\rho_{eff}}[\nabla N]^T[\nabla N]dv = [\bar{K}_r] + j[\bar{K}_i]$$

$$[\bar{C}_F] = \iint_{\Gamma_Z} Y\{N\}\{N\}^T ds = [\bar{C}_r] + j[\bar{C}_i]$$

$$[f_F] = -j\omega \iint_{\Gamma_v} \{N\}\{N\}^T ds\{v_{n,f}\}$$

The final matrix equation is expressed as:

$$\{-\omega^2[M_F] + j\omega[C_F] + [K_F]\}\{p\} = \{f_F\} \tag{8.91}$$

where:

$$[M_F] = [\bar{M}_{F,r}]$$

$$[K_F] = [\bar{K}_{F,r}] - \omega[\bar{C}_{F,i}]$$

$$[C_F] = [\bar{C}_{F,r}] - \omega[\bar{M}_{F,i}] + \frac{1}{\omega}[\bar{K}_{F,i}]$$

8.3.5.2.2. Delany-Bazley and Miki Models

Instead of inputting five parameters in the Johnson-Champoux-Allard model, only one parameter, fluid resistivity, is required in the Delany-Bazley and Miki models. In general, complex impedance is defined by:

$$Z_c = R + jX \tag{8.92}$$

The complex propagation constant s is defined by

$$\gamma = \alpha + j\beta \tag{8.93}$$

where:

R = resistance
X = reactance
α = attenuation constant
β = phase constant

In both the Delany-Bazley and the Miki models, variables are approximated by the functions:

$$R = \rho_0 c_0 \left\{1 + a\left(\frac{f}{\sigma}\right)^b\right\} \tag{8.94}$$

$$X = -\rho_0 c_0 \left\{ c \left(\frac{f}{\sigma} \right)^d \right\} \tag{8.95}$$

$$\alpha = \frac{\omega}{c_0} q \left(\frac{f}{\sigma} \right)^r \tag{8.96}$$

$$\beta = \frac{\omega}{c_0} \left\{ 1 + s \left(\frac{f}{\sigma} \right)^t \right\} \tag{8.97}$$

where:

f = frequency
σ = fluid resistivity

Coefficients are provided in the following table.

Table 8.1: Coefficients of the Approximation Functions for Delany-Bazley and Miki Models

Coefficient	Delany-Bazley	Miki
a	0.0497	0.0699
b	-0.754	-0.632
c	0.0758	0.107
d	-0.732	-0.632
q	0.169	0.160
r	-0.618	-0.618
s	0.0858	0.109
t	-0.700	-0.618

The proposed working range for the Delany-Bazley model is $0.01 < \frac{f}{\sigma} < 1.00$. The Miki model can be extended to $0.01 < \frac{f}{\sigma}$.

8.3.5.2.3. Complex Media Properties

The complex effective mass density and sound velocity in Equation 8.84 (p. 270) may be given by:

$$\rho_{eff} = \rho_r + j\rho_i \tag{8.98}$$

$$c_{eff} = c_r + jc_i \tag{8.99}$$

The Helmholtz equation with complex mass density and sound velocity is directly solved with the defined parameters.

The complex effective mass density and sound velocity are input with the **TBDATA** command for the **TB**,PERF material.

8.3.5.3. Impedance Sheet Approximation

Analogous to Equation 8.20 (p. 257), the boundary condition across an impedance sheet in the acoustic domain is given by:

$$v_{n,F^+}(\vec{r}) - v_{n,F^-}(\vec{r}) = Y(\vec{r})p(\vec{r}) \tag{8.100}$$

where:

$v_{n,F^+}(\vec{r})$ = normal velocity above the impedance sheet

$v_{n,F^-}(\vec{r})$ = normal velocity below the impedance sheet

$Y(\vec{r})$ = admittance on the impedance sheet

$p(\vec{r})$= pressure on the impedance sheet

Assuming that the acoustic domain is divided into Ω^+ and Ω^- with the impedance sheet on the surface, substituting Equation 8.100 (p. 273) into Equation 8.10 (p. 255) casts the damping matrix as Equation 8.22 (p. 258). For a thin perforated layer, the sheet impedance (input as IMPD on the **BF** command) is given by:

$$Z = Z_c \frac{Z_L + jZ_c \tan(kd)}{Z_c + jZ_L \tan(kd)} \tag{8.101}$$

where:

Z_L = load impedance

d = thickness of perforated layer

$Z_c = \rho_{eff} c_{eff}$ = sound impedance of the perforated material defined by Equation 8.88 (p. 270)

$k = \frac{\omega}{c_{eff}}$ = complex wave number of the perforated material (c_{eff} is defined by Equation 8.87 (p. 270))

For a thin perforated material layer backed with a rigid wall, the impedance is given by:

$$Z = -jZ_c \cot(kd) \tag{8.102}$$

8.3.5.4. Viscous-Thermal Media

An acoustic waves propagating in viscous-thermal media have a complex propagating constant in the frequency domain. The attenuation of the acoustic wave is proportional to the shear and bulk viscosity and the thermal conduction coefficient of the media.

8.3.5.4.1. Boundary Layer Impedance (BLI) Model

In a rigid-walled, bounded domain, the interaction between he acoustic movement and both the diffusion of heat and the diffusion of the shear wave leads to a reactive and absorbing process on the walls. The entropic and vortical perturbations diffuse into the rigid wall in a direction normal to the boundary and decay to zero depending on the thickness of the boundary layers. A hybrid numerical and analytic solution using the boundary layer impedance (BLI) model is proposed with the following assumptions ([414] (p. 944)):

- The thicknesses of the boundary layers are much smaller than the dimensions of the numerical domain, while the pressure variation is constant over the boundary layers' thicknesses.
- The acoustic pressure is approximated by its zero expansion in the boundary layers and its flow is tangential to the wall.
- Spatial variations of both velocity and temperature in the normal direction are much greater than spatial variations in the tangential directions.

The fundamental Helmholtz governing equation and boundary condition in the BLI model is given by:

$$(\nabla^2 + k_a^2)p = 0 \quad \text{in the bulk domain}$$
$$\frac{\partial p}{\partial n} = jk_0\beta p \quad \text{on the boundaries} \tag{8.103}$$

where:

$$k_a \approx k_0(1-(j/2)k_0 l_{vh})$$

$$\beta = k_0\left[\frac{k_w^2}{k_0^2 k_v} + \frac{(\gamma-1)}{k_h}\right]$$

$k_0 = \omega / c_0$
$l_{vh} = l_v + (\gamma-1)l_h$
k_w = the transverse wavenumber

$$k_v = (1-j)\sqrt{k_0/(2l'_v)}$$

$$k_h = (1-j)\sqrt{k_0/(2l_h)}$$

$$l_v = (\frac{4}{3}\mu + \eta)/(\rho_0 c_0)$$

$l'_v = \mu / (\rho_0 c_0)$
$l_h = \lambda / (\rho_0 c_0 C_p)$
$\gamma = C_p / C_v$ = the specific heat ratio
ω = the angular frequency
ρ_0 = the mass density of the fluid
c_0 = the sound speed of the fluid
μ = the dynamic viscosity
η = the bulk viscosity
λ = the thermal conductivity
C_p = the heat coefficient at constant pressure per unit mass
C_v = the heat coefficient at constant volume per unit mass

8.3.5.4.2. Low Reduced Frequency (LRF) Model

The low reduced frequency (LRF) model is derived from simplified Navier-Stokes equations with the following assumptions ([415] (p. 944)):

- The acoustic wavelength is much greater than the length scale of the geometry
- The acoustic wavelength is much greater than the boundary layer thickness

The Helmholtz wave equation with modified density and bulk module for the LRF model is as follows:

$$\nabla \cdot (\frac{1}{\hat{\rho_0}} \nabla p) + \frac{\omega^2}{\hat{K_0}} p = 0 \quad \text{in the bulk domain} \tag{8.104}$$

where:

$\hat{\rho_0} = -\frac{\rho_0}{B(s/l)} =$ the modified density

$\hat{K_0} = K_0 \frac{n(s\sigma/l)}{\gamma} =$ the modified bulk module

ρ_0 = the mass density

K_0 = the modified bulk module

l = the length scale

$s = l\sqrt{\frac{\rho_0 \omega}{\mu}} =$ the shear wave number

$\sigma = C_p \eta / \kappa$ = Prandtl number

γ = specific heat ratio

The viscous and thermal effects are taken into account by the function B(s/l) and n(sσ/l), respectively.

The analytic solution of both the functions B(s/l) and n(sσ/l) can be derived for three particular cases: thin layer, tube with rectangular cross-section, and tube with circular cross section.

A thin layer is specified by thickness h. The functions B(s/l) and n(sσ/l) are derived by:

$$B\left(\frac{s}{l}\right) = \frac{\tanh(s\sqrt{i})}{s\sqrt{i}} - 1 \tag{8.105}$$

$$n\left(\frac{s\sigma}{l}\right) = \left(1 + \frac{\gamma - 1}{\gamma} D\left(\frac{s\sigma}{l}\right)\right)^{-1} \tag{8.106}$$

where:

$$D\left(\frac{s\sigma}{l}\right) = \frac{\tanh(s\sigma\sqrt{i})}{s\sigma\sqrt{i}} - 1$$

A tube with a rectangular cross-section is specified by width w and height h. The functions B(s/l) and n(sσ/l) are derived by:

$$B\left(\frac{s}{l}\right) = -\frac{64i(s^2/l^2)}{w^2 h^2} \sum_{m=1,3,5\ldots} \sum_{n=1,3,5\ldots} \frac{\sin^2(\beta_m w/2)\sin^2(\beta_n h/2)}{\beta_m^2 \beta_n^2 (\beta_m^2 + \beta_n^2 + i(s^2/l^2))} \tag{8.107}$$

$$n\left(\frac{s\sigma}{l}\right) = \left(1 + \frac{\gamma - 1}{\gamma} D\left(\frac{s\sigma}{l}\right)\right)^{-1} \tag{8.108}$$

where:

$$D\left(\frac{s\sigma}{l}\right) = -\frac{64i(s^2\sigma^2/l^2)}{w^2h^2}\sum_{m=1,3,5\ldots}\sum_{n=1,3,5\ldots}\frac{\sin^2(\beta_m w/2)\sin^2(\beta_n h/2)}{\beta_m^2\beta_n^2(\beta_m^2+\beta_n^2+i(s^2\sigma^2/l^2))}$$

A tube with a circular cross section is specified by radius R. The functions B(s/l) and n(sσ/l) are derived by:

$$B\left(\frac{s}{l}\right) = \frac{J_2(is\sqrt{i})}{J_0(is\sqrt{i})} \tag{8.109}$$

$$n\left(\frac{s\sigma}{l}\right) = \left(1+\frac{\gamma-1}{\gamma}D\left(\frac{s\sigma}{l}\right)\right)^{-1} \tag{8.110}$$

where:

$$D\left(\frac{s\sigma}{l}\right) = \frac{J_2(is\sigma\sqrt{i})}{J_0(is\sigma\sqrt{i})}$$

J_0 = the zero-order Bessel function
J_2 = the second-order Bessel function

8.4. Acoustic Fluid-Structural Interaction (FSI)

The following topics are available in this section:

8.4.1. Coupled Acoustic Fluid-Structural System with an Unsymmetric Matrix Equation

The coupling conditions on the interface between the acoustic fluid and the structure are given by:

$$\bar{\bar{\sigma}}(\bar{u}_S)\vec{n} + p\vec{n} = 0 \quad \text{on } \Gamma_I \tag{8.111}$$

$$\vec{n}\cdot\bar{u}_S - \vec{n}\cdot\bar{u}_F = 0 \quad \text{on } \Gamma_I \tag{8.112}$$

where:

$\bar{\bar{\sigma}}(\bar{u}_S)$ = solid stress tensor
p = acoustic pressure
$\bar{u}_S$ = displacement in solid
$\bar{u}_F$ = displacement in acoustic fluid
$\vec{n}$ = outward normal unit vector of fluid domain

Equation 8.111 (p. 276) is a kinetic condition relating the solid stress to the pressure imposed on the interface by sound. Equation 8.112 (p. 276) is a kinematic condition that assumes that there is no friction between the solid and acoustic fluid on the interface.

In order to completely describe the FSI problem, the fluid pressure load acting at the interface is added to Equation 15.5 (p. 764). This effect is included in FLUID29, FLUID30, FLUID220, and FLUID221 only if KEYOPT(2) $\neq$ 1. Hence, the structural equation is rewritten as:

$$([M_F]+[S_F])\ddot{p}+[C_F]\dot{p}+[K_F]p+\bar{\rho}_0[R]^T\ddot{u}=[f_F] \tag{8.113}$$

The fluid pressure load vector $\{f_e^{pr}\}$ at the interface S is obtained by integrating the pressure over the area of the surface as follows:

$$\{f_e^{pr}\}=\iint_{\Gamma_i}\{N'\}p\bar{n}ds \tag{8.114}$$

where:

{N'} = shape functions employed to discretize the displacement components u, v, and w (obtained from the structural element).

Substituting the finite element approximating function for pressure given by Equation 8.16 (p. 256) into Equation 8.114 (p. 277) leads to:

$$\{f_e^{pr}\}=\iint_{\Gamma_i}\{N'\}\{N\}^T\{n\}ds\{p_e\} \tag{8.115}$$

By comparing the integral in Equation 8.115 (p. 277) with the matrix definition of $[R]^T$ in Equation 8.19 (p. 257), the following relation becomes clear:

$$\{f_e^{pr}\}=[R]\{p_e\} \tag{8.116}$$

Substituting Equation 8.116 (p. 277) into Equation 8.113 (p. 277) results in the dynamic elemental equation of the structure, expressed as:

$$[M_S]\{\ddot{u}_e\}+[C_S]\{\dot{u}_e\}+[K_S]\{u_e\}-[R]\{p_e\}=\{f_S\} \tag{8.117}$$

Equation 8.19 (p. 257) and Equation 8.117 (p. 277) describe the complete finite element discretized equations for the FSI problem. These equations are written in assembled form as:

$$\begin{bmatrix}[M_S] & 0\\ \bar{\rho}_0[R]^T & [M_F]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}_e\}\\ \{\ddot{p}_e\}\end{Bmatrix}+\begin{bmatrix}[C_S] & 0\\ 0 & [C_F]\end{bmatrix}\begin{Bmatrix}\{\dot{u}_e\}\\ \{\dot{p}_e\}\end{Bmatrix}+\begin{bmatrix}[K_S] & -[R]\\ 0 & [K_F]\end{bmatrix}\begin{Bmatrix}\{u_e\}\\ \{p_e\}\end{Bmatrix}=\begin{Bmatrix}f_S\\ f_F\end{Bmatrix} \tag{8.118}$$

The acoustic fluid element in an FSI problem will generate all the submatrices with a superscript F in addition to the coupling submatrices $\bar{\rho}_0$, $[R]^T$, and [R]. Submatrices with a superscript S will be generated by the compatible structural element used in the model.

Assuming that the actual surface is at an elevation η relative to the mean surface in z-direction, the pressure for a sloshing (free) surface is given by:

$$p=\rho_F g\eta \tag{8.119}$$

By utilizing the definition of velocity and the momentum conservation equation in addition to Equation 8.119 (p. 277), pressure can be expressed as:

$$\frac{1}{\rho_F}\frac{\partial p}{\partial z}=-\frac{1}{\rho_F g}\ddot{p} \tag{8.120}$$

The surface integration of the "weak" form (Equation 8.19 (p. 257)) on the sloshing surface is given by:

$$\iint_{\Gamma_{SL}} \frac{1}{\rho_F} w\hat{n} \cdot \nabla p ds = -\iint_{\Gamma_{SL}} \frac{1}{\rho_F g} w\ddot{p} ds \tag{8.121}$$

The acoustic fluid matrix equation with sloshing effect is expressed as:

$$([M_F] + [S_F])\ddot{p} + [C_F]\dot{p} + [K_F]p + \bar{\rho}_0 [R]^T \ddot{u} = [f_F] \tag{8.122}$$

where:

$$S_F = \frac{\bar{\rho}_0}{g} \iint_{\Gamma_{SL}} \frac{1}{\rho_F} \{N\}\{N\}^T ds = \text{acoustic sloshing mass matrix}$$

Substituting Equation 8.122 (p. 278) into Equation 8.118 (p. 277) yields:

$$\begin{bmatrix} [M_S] & 0 \\ \bar{\rho}_0 [R]^T & [M_F] + [S_F] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}_e\} \\ \{\ddot{p}_e\} \end{Bmatrix} + \begin{bmatrix} [C_S] & 0 \\ 0 & [C_F] \end{bmatrix} \begin{Bmatrix} \{\dot{u}_e\} \\ \{\dot{p}_e\} \end{Bmatrix} + \begin{bmatrix} [K_S] & -[R] \\ 0 & [K_F] \end{bmatrix} \begin{Bmatrix} \{u_e\} \\ \{p_e\} \end{Bmatrix} = \begin{Bmatrix} f_S \\ f_F \end{Bmatrix} \tag{8.123}$$

If the impedance boundary is exerted on the FSI interface (input as IMPD on the **SF** command), the coupling condition expressed in Equation 8.112 (p. 276) is rewritten as:

$$\vec{n} \cdot \vec{u}_F - \vec{n} \cdot \vec{u}_S = -j\frac{Y}{\omega} p \qquad \text{on } \Gamma_I \tag{8.124}$$

Substituting Equation 8.124 (p. 278) into Equation 8.19 (p. 257) yields:

$$[M_F]\{\ddot{p}_e\} + ([C_F] + [C_{FSI}])\{\dot{p}_e\} + [K_F]\{p_e\} + \bar{\rho}_0 [R]^T \{\ddot{u}_e\} = \{f_F\} \tag{8.125}$$

where:

$$[C_{FSI}] = \bar{\rho}_0 \iint_{\Gamma_{FSI}} Y\{N\}\{N\}^T ds = \text{acoustic damping matrix on FSI interface}$$

Damping matrix $[C_{FSI}]$ on the impedance FSI interface has been shown to be the same as the damping matrix in Equation 8.22 (p. 258). Therefore, the coupling matrix expressed in Equation 8.123 (p. 278) can still be the final matrix equation.

8.4.2. Coupled Acoustic Fluid-Structural System with Symmetric Matrix Equation for Full Harmonic Analysis

The matrix in Equation 8.123 (p. 278) has been shown to be unsymmetric. Solving Equation 8.123 (p. 278) may consume more computer resources and time than solving the symmetric matrix equation. For the frequency domain, assume that:

$$p = \dot{q} = j\omega q \tag{8.126}$$

Substituting Equation 8.126 (p. 278) into Equation 8.117 (p. 277) and Equation 8.19 (p. 257) yields:

$$[M_S]\{\ddot{u}_e\} + [C_S]\{\dot{u}_e\} + [K_S]\{u_e\} - j\omega[R]\{q_e\} = \{f_S\} \tag{8.127}$$

$$j\omega([M_F] + [S_F])\{\ddot{q}_e\} + j\omega[C_F]\{\dot{q}_e\} + j\omega[K_F]\{q_e\} + (j\omega)(j\omega)\bar{\rho}_0 [R]^T \{u_e\} = \{f_F\} \tag{8.128}$$

Dividing coupled Equation 8.128 (p. 278) by $-j\omega\bar{\rho}_0$ yields the acoustic matrix equation, written as:

$$-\frac{1}{\bar{\rho}_0}([M_F]+[S_F])\{\ddot{q}_e\}-\frac{1}{\bar{\rho}_0}[C_F]\{\dot{q}_e\}-\frac{1}{\bar{\rho}_0}[K_F]\{q_e\}-(j\omega)[R]^T\{u_e\}=\frac{j}{\omega\bar{\rho}_0}\{f_F\} \tag{8.129}$$

The coupled matrix equation is given by:

$$\left(-\omega^2\begin{bmatrix}[M_S] & 0\\ 0 & -\frac{[M_F]}{\bar{\rho}_0}-\frac{[S_F]}{\bar{\rho}}\end{bmatrix}+j\omega\begin{bmatrix}[C_S] & -[R]\\ -[R]^T & -\frac{[C_F]}{\bar{\rho}_0}\end{bmatrix}+\begin{bmatrix}[K_S] & 0\\ 0 & -\frac{[K_F]}{\bar{\rho}_0}\end{bmatrix}\right)\begin{Bmatrix}\{u_e\}\\ \{p_e\}\end{Bmatrix}=\begin{Bmatrix}f_S\\ \frac{jf_F}{\omega\bar{\rho}_0}\end{Bmatrix} \tag{8.130}$$

After solving Equation 8.130 (p. 279), the pressure is obtained using Equation 8.126 (p. 278).

8.4.3. Coupled Acoustic Fluid-Structural System with Symmetric Matrix Equation for Lossless Modal Analysis

Transformation Equation 8.126 (p. 278) cannot be used for the modal analysis because of the involvement of frequency. Thus, a displacement potential Φ is introduced. This leads to:

$$\vec{u}_F=\frac{\bar{\rho}_0}{\rho_F}\nabla\varphi \tag{8.131}$$

The momentum and continuity equations in lossless and source free acoustic media are respectively given by:

$$\frac{1}{\rho_F}\nabla p=\omega^2\vec{u}_F \qquad \text{in } \Omega_F \tag{8.132}$$

$$\frac{1}{\rho_F c^2}p+\nabla\cdot\vec{u}_F=0 \qquad \text{in } \Omega_F \tag{8.133}$$

The acoustic fluid-structural coupling conditions are rewritten as:

$$\bar{\bar{\sigma}}(\vec{u}_S)\vec{n}+p\vec{n}=0 \qquad \text{on } \Gamma_I \tag{8.134}$$

$$\vec{n}\cdot\vec{u}_S-\hat{n}\cdot\frac{\bar{\rho}_0}{\rho_F}\nabla\varphi=0 \qquad \text{on } \Gamma_I \tag{8.135}$$

Equation 8.132 (p. 279) is substituted into Equation 8.132 (p. 279) multiplied by $\bar{\rho}_0\nabla w$. Taking the integration over the acoustic fluid domain, Equation 8.132 (p. 279) is rewritten as:

$$\bar{\rho}_0\iiint_{\Omega_F}\nabla w\cdot(\frac{1}{\rho_F}\nabla p)dv=\omega^2\bar{\rho}_0^2\iiint_{\Omega_F}\nabla w\cdot(\frac{1}{\rho_F}\nabla\varphi)dv \qquad \text{in } \Omega_F \tag{8.136}$$

The matrix notation of Equation 8.136 (p. 279) is given by:

$$-\omega^2\bar{\rho}_0[K_F]\{\varphi_e\}+[K_F]^T\{p_e\}=0 \qquad \text{in } \Omega_F \tag{8.137}$$

Equation 8.132 (p. 279) is substituted into Equation 8.133 (p. 279) and multiplied by $\bar{\rho}_0 q$. Taking the integration over the acoustic fluid domain, Equation 8.133 (p. 279) is rewritten as:

$$\bar{\rho}_0 \iiint_{\Omega_F} w\left(\frac{1}{\rho_F c^2}\right)pdv + \bar{\rho}_0^2 \iiint_{\Omega_F} w\nabla\cdot\left(\frac{1}{\rho_F}\nabla\varphi\right)dv = 0 \qquad \text{in } \Omega_F \tag{8.138}$$

By utilizing Green's identity and Equation 8.138 (p. 280), Equation 8.138 (p. 280) becomes:

$$\bar{\rho}_0 \iiint_{\Omega_F} \frac{1}{\rho_F c^2} wpdv - \bar{\rho}_0^2 \iiint_{\Omega_F} \nabla w\cdot\left(\frac{1}{\rho_F}\nabla\varphi\right)dv + \bar{\rho}_0 \iint_{\Gamma_I} w\hat{n}\cdot\bar{u}_S ds = 0 \qquad \text{in } \Omega_F \tag{8.139}$$

The matrix notation of Equation 8.139 (p. 280) is given by:

$$\frac{1}{\bar{\rho}_0}[M_F]\{p_e\} - [K_F]\{\varphi_e\} + [R]^T\{u_e\} = 0 \qquad \text{in } \Omega_F \tag{8.140}$$

The surface integration on the sloshing surface can be expressed as:

$$\bar{\rho}_0 \iint_{\Gamma_{SI}} w\hat{n}\cdot\bar{u}_F ds = \bar{\rho}_0 \iint_{\Gamma_{SI}} wu_{F,z} ds = \bar{\rho}_0 \iint_{\Gamma_{SI}} \frac{1}{\rho_F g} wpds \qquad \text{in } \Omega_F \tag{8.141}$$

Adding the matrix notation of Equation 8.141 (p. 280) into Equation 8.140 (p. 280) yields:

$$\frac{1}{\bar{\rho}_0}([M_F]+[S_F])\{p_e\} - [K_F]\{\varphi_e\} + [R]^T\{u_e\} = 0 \qquad \text{in } \Omega_F \tag{8.142}$$

Assembling Equation 8.122 (p. 278), Equation 8.137 (p. 279), and Equation 8.142 (p. 280) together yields the following lossless symmetric eigen matrix equation:

$$-\omega^2\begin{bmatrix} M_S & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \bar{\rho}_0 K_F \end{bmatrix}\begin{Bmatrix} \{u_e\} \\ \{p_e\} \\ \{\varphi_e\} \end{Bmatrix} + \begin{bmatrix} K_S & -R & 0 \\ -R^T & -\frac{1}{\bar{\rho}_0}M_F - \frac{1}{\bar{\rho}_0}S_F & K_F \\ 0 & K_F^T & 0 \end{bmatrix}\begin{Bmatrix} \{u_e\} \\ \{p_e\} \\ \{\varphi_e\} \end{Bmatrix} = 0 \tag{8.143}$$

It should be noted that an auxiliary degree of freedom, velocity potential Φ, is introduced to derive the symmetric eigen equation. It should also be noted that the mass matrix of Equation 8.143 (p. 280) is related to the velocity potential rather than pressure.

8.5. Pure Scattered Pressure Formulation

An alternative method to simulate the pressure wave excited by the analytic incident wave sources (defined using the **AWAVE** command), is to use scattered pressure as the solved variable. Pure scattered pressure formulation may be used in the following situations:

- The scattered pressure is much smaller than the incident pressure wave, and the scattered properties are under the investigation.
- The analytic wave sources defined using the **AWAVE** command are inside the model for real acoustic devices, such as an enclosed-back loudspeaker.

When the scattered pressure is investigated as the degree of freedom, the total pressure is split into incident pressure and scattered pressure. This can be expressed as:

$$p^{tot} = p^{inc} + p^{sc} \tag{8.144}$$

The pressure boundary conditions are also changed for the scattered pressure.

For the Dirichlet boundary ($p = p_0$):

$$p^{sc} = p_0 - p^{inc} \tag{8.145}$$

For the Neumann boundary $\left(\frac{\partial p}{\partial n} = 0\right)$:

$$\hat{n} \cdot \nabla p^{sc} = -\hat{n} \cdot \nabla p^{inc} \tag{8.146}$$

The boundary of each decomposed domain is assumed to consist of the truncation surface Γ_{ext}, impedance surface Γ_Z, and FSI interface Γ_{FSI}. This can be expressed as:

$$\Gamma_S = \Gamma_{ext} + \Gamma_Z + \Gamma_{FSI} \tag{8.147}$$

The "weak" form for the scattered pressure derived from Equation 8.19 (p. 257) is written as:

$$\iiint_{\Omega} \{\nabla w \cdot (\frac{1}{\rho} \nabla p^{sc}) - \frac{\omega^2}{\rho c^2} w p^{sc}\} dv - \iint_{\Gamma_{ext}} \frac{1}{\rho} w \hat{n} \cdot \nabla p^{sc} ds + j\omega \iint_{\Gamma_Z} Y w p^{sc} ds = 0 \tag{8.148}$$

By decomposing Equation 8.148 (p. 281) into the environment medium (Ω_0) and non-environment medium (Ω_d), and assuming that the incident pressure wave satisfies the wave equation in the environment medium, we can state that:

$$\nabla^2 p_0^{inc} + k_0^2 p_0^{inc} = 0 \tag{8.149}$$

The "weak" form of the incident wave including acoustic-structure interaction can be written as:

$$\begin{aligned} &\iiint_{V_d} \{\nabla w \cdot (\frac{1}{\rho_d} \nabla p^{inc}) - \frac{\omega^2}{\rho_d c_d^2} w p^{inc}\} dv - \oiint_{S_d} \frac{1}{\rho_d} w \nabla p^{inc} \cdot \hat{n} ds - \omega^2 \iint_{S_{FSI}} w \hat{n} \cdot \bar{u}_S ds \\ &+ j\omega \iint_{S_Z} w v_{n,s} ds + \iint_{S_{FSI}} \frac{1}{\rho_F} w \nabla p^{inc} \cdot \hat{n} ds + j\omega \iint_{S_Z} Y w p^{inc} ds + \iint_{S_Z} \frac{1}{\rho_0} w \nabla p^{inc} \cdot \hat{n} ds = 0 \end{aligned} \tag{8.150}$$

Substituting Equation 8.148 (p. 281) and Equation 8.150 (p. 281) into Equation 8.19 (p. 257) yields the "weak" form for pure scattered pressure based formulation, expressed as:

$$\begin{aligned} &\iiint_{\Omega_F} \{\nabla w \cdot (\frac{1}{\rho_F} \nabla p^{sc}) - \frac{\omega^2}{\rho_F c_F^2} w p^{sc}\} dv \\ &- \iint_{\Gamma_{ext}} \frac{1}{\rho_F} w \nabla p^{sc} \cdot \hat{n} ds + j\omega \iint_{\Gamma_Z} Y w p^{sc} ds - \omega^2 \iint_{\Gamma_{FSI}} w \hat{n} \cdot \bar{u}_S ds \\ &= -\iiint_{\Omega_d} \{\nabla w \cdot (\frac{1}{\rho_d} \nabla p^{inc}) - \frac{\omega^2}{\rho_d c_d^2} w p^{inc}\} dv + \oiint_{\Gamma_d} \frac{1}{\rho_d} w \nabla p^{inc} \cdot \hat{n} ds \\ &- j\omega \iint_{\Gamma_Z} w v_{n,s} ds - \iint_{\Gamma_{FSI}} \frac{1}{\rho_F} w \nabla p^{inc} \cdot \hat{n} ds - j\omega \iint_{\Gamma_Z} w (Y p^{inc} + \frac{1}{\rho_0} \hat{n} \cdot \nabla p^{inc}) ds \end{aligned} \tag{8.151}$$

By combining the Neumann boundary condition Γ_N (expressed in Equation 8.145 (p. 281)) with Equation 8.151 (p. 281), the matrix form is written as:

$$\{-\omega^2[M_F]+j\omega[C_F]+[K_F]\}\{p_e\}-\omega^2\bar{\rho}_0[R]^T\{u_e\}=\{f_F\} \tag{8.152}$$

where:

$$[M_F]=\bar{\rho}_0\iiint_{\Omega_F}\frac{1}{\rho_F c_F^2}\{N\}\{N\}^T dv$$

$$[C_F]=\bar{\rho}_0\iint_{\Gamma_Z} Y\{N\}\{N\}^T ds$$

$$[K_F]=\bar{\rho}_0\iiint_{\Omega_F}\frac{1}{\rho_F}\{\nabla N\}\{\nabla N\}^T dv$$

$$[R]^T=\iint_{\Gamma_{FSI}}\{N\}\{n\}^T\{N'\}^T ds$$

$$\{f_F\}=-\iiint_{\Omega_d}\frac{1}{\rho_d}\{\nabla N\}\cdot\nabla p^{inc}-\frac{\omega^2}{\rho_d c_d^2}\{N\}p^{inc}\}dv+\oiint_{\Gamma_d}\frac{1}{\rho_d}\{N\}n\cdot\nabla p^{inc}ds-j\omega\iint_{\Gamma_Z}\{N\}v_{n,s}ds$$
$$-\iint_{\Gamma_{FSI}}\frac{1}{\rho_F}\{N\}\hat{n}\cdot\nabla p^{inc}ds-j\omega\iint_{\Gamma_Z}\{N\}(Yp^{inc}+\frac{1}{\rho_0}n\cdot\nabla p^{inc})ds-\iint_{\Gamma_N}\frac{1}{\rho_F}\{N\}\hat{n}\cdot\nabla p^{inc}ds$$

Since the scattered pressure is solved in the acoustic domain and the total pressure is coupled with the structural domain, the total pressure in the structural matrix in Equation 8.117 (p. 277) must be split to ensure a consistent pressure degree of freedom in the solution. This is expressed as:

$$[M_S]\ddot{\bar{u}}+[C_S]\dot{\bar{u}}+[K_S]\bar{u}-[R]p^{sc}-[R]p^{inc}+[f_S]=0 \tag{8.153}$$

The coupling matrix equation is given by:

$$\left(-\omega^2\begin{bmatrix}[M_S] & 0\\ \bar{\rho}_0[R]^T & [M_F]\end{bmatrix}+j\omega\begin{bmatrix}[C_S] & 0\\ 0 & [C_F]\end{bmatrix}+\begin{bmatrix}[K_S] & -[R]\\ 0 & [K_F]\end{bmatrix}\right)\begin{Bmatrix}\{u_e\}\\ \{p_e^{sc}\}\end{Bmatrix}=\begin{Bmatrix}f_S\\ f_F\end{Bmatrix}+\begin{Bmatrix}[R]p^{inc}\\ 0\end{Bmatrix} \tag{8.154}$$

Equation 8.49 (p. 263) is used for the scattered pressure in the PML, because only scattered pressure is absorbed and the incident is assumed to satisfy the wave equation in the environment medium.

8.6. Acoustic Output Quantities

Either the total nodal pressure or the scattered nodal pressure is solved and output at the nodes. Using the computed nodal pressure values, the pressure gradient is evaluated at the nodes and the element centroid, which is expressed as follows:

$$\nabla p=\{\nabla N\}^T\{p_e\} \tag{8.155}$$

Both the nodal velocity (output as PG on the **PRNSOL**, **PLNSOL**, and **NSOL** commands) and the element centroid velocity (output as SMISC on the element table) are calculated for modal (**ANTYPE**,MODAL) and full harmonic (**ANTYPE**,HARM) analyses as follows:

$$\bar{v}=(-\frac{4\mu}{3\rho_0^2c_0^2}+j\frac{1}{\omega\rho_0})\nabla p \tag{8.156}$$

The nodal and element centroid sound pressure levels (SPL) are output in the post-processor. SPL is given by:

$$L_{SPL} = 10\log\left(\frac{p_{rms}^2}{p_{ref}^2}\right) = 20\log\left(\frac{p_{rms}}{p_{ref}}\right) \quad (dB) \tag{8.157}$$

where:

p_{ref} = reference pressure (input as PREF on **R** command, defaults to 20 x 10^{-6}Pa)

$p_{rms} = \frac{p}{\sqrt{2}}$ = root mean square of peak pressure in modal and harmonic analysis

The a-weighted sound pressure level (dBA) is defined by:

$$dBA = SPL + W_A \tag{8.158}$$

where:

$$W_A = 10\log\left[\frac{1.562339f^4}{(f^2+107.65265^2)(f^2+737.86223^2)}\right] + 10\log\left[\frac{2.242881\times10^6 f^4}{(f^2+20.598997^2)^2(f^2+12194.22^2)^2}\right]$$

The far-field pressure parameters (output via the **PLFAR** and **PRFAR** command) outside of the model are expressed as:

$$p(\vec{r}) = \oiint_{S[V]}\left[p(\vec{r}_s)\frac{\partial G(\vec{r}\mid\vec{r}_s)}{\partial n} - G(\vec{r}\mid\vec{r}_s)\frac{\partial p(\vec{r}_s)}{\partial n}\right]ds_s \tag{8.159}$$

where:

S[V] = enclosed equivalent source surface (defined as MXWF on **SF** command)

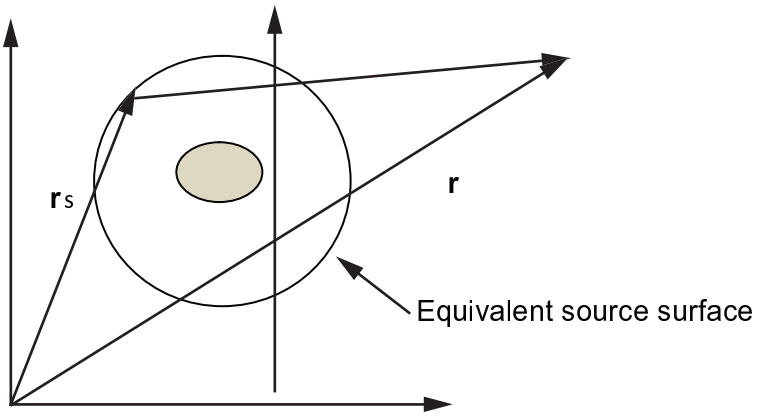
$G(\vec{r}\mid\vec{r}_s) = \frac{e^{-jk|\vec{r}-\vec{r}_s|}}{4\pi|\vec{r}-\vec{r}_s|}$ = Green's function in free space

$\vec{r}$ = far-field position

$\vec{r}_s$ = equivalent source position on the enclosed surface

k = wave number

Figure 8.2: Equivalent Source Principle



The sound power level radiated by the sound source is expressed as:

$$L_W = 10\log\left(\frac{W_{rad}}{W_{ref}}\right) \quad (dB) \tag{8.160}$$

where:

W_{ref} = reference sound power (defaults to 10-12 w)
W_{rad} = radiated sound power

The radiated sound power is expressed as:

$$W_{rad} = \iint U d\Omega = \frac{1}{2}\mathrm{Re}\iint p\vec{v}^{\,*}\cdot\hat{r}r^2\sin\theta d\theta d\varphi \tag{8.161}$$

The directivity of the sound source is expressed as:

$$G_D(\varphi,\theta) = \frac{\Omega U(\varphi,\theta)}{W_{rad}} \tag{8.162}$$

where:

$\Omega = \iint d\Omega = \iint \sin\theta d\theta d\varphi =$ solid angle

$U(\varphi,\theta) = \frac{1}{2}\mathrm{Re}(p\vec{v}^{\,*})\cdot\hat{r}r^2 =$ sound radiation intensity

$\hat{r} =$ radical unit vector in spherical coordinates

The target strength (TS) is used to describe the reflected acoustic signals. Target strength is given by:

$$TS = 10\log(R^2 p_{sc}^2) \quad (dB) \tag{8.163}$$

where:

R = radius at the target position
p_{sc} = scattered pressure at the target position

The time-averaged sound power through an area S in a harmonic analysis is defined by:

$$P = \frac{1}{2}\iint_S pv^* \cdot \hat{n} ds \tag{8.164}$$

where:

p = complex pressure
v* = complex conjugate velocity

In a network, the return loss and transmission loss are respectively defined as:

$$RL = 10\log_{10}\left(\frac{P_{in}}{P_r}\right) \quad (dB) \tag{8.165}$$

$$TL = 10\log_{10}\left(\frac{P_{in}}{P_t}\right) \quad (dB) \tag{8.166}$$

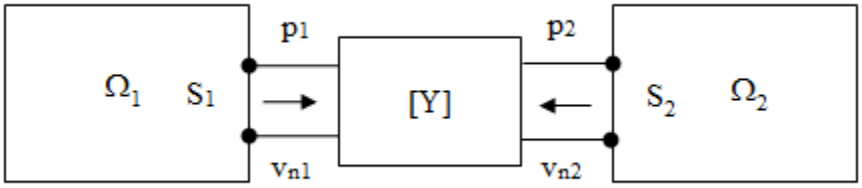
where:

P_{in} = incident sound power at inlet
P_r = reflected sound power at inlet
P_t = transmitted sound power at outlet

8.7. Transfer Admittance Matrix

A complicated structure may be trimmed and represented by a 2×2 transfer admittance matrix that is a part of the finite element model.

Figure 8.3: Two-Port Transfer Admittance Matrix



The transfer admittance matrix is given by:

$$\begin{Bmatrix} v_{n1} \\ v_{n2} \end{Bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix} + \begin{Bmatrix} \alpha_1 \\ \alpha_2 \end{Bmatrix} \tag{8.167}$$

where:

p_1 = pressure at port 1
v_{n1} = normal velocity at port 1
p_2 = pressure at port 2
v_{n2} = normal velocity at port 2
Y_{11}, Y_{22} = self-admittances
Y_{12}, Y_{21} = mutual admittances
α_1, α_2 = internal sources (usually equal to zero in an acoustic analysis)

The following related topics are available:
8.7.1. Transfer Admittance Matrix Connected to Acoustic Domains
8.7.2. Transfer Admittance Matrix Connected to the Structural and Acoustic Domain

8.7.1. Transfer Admittance Matrix Connected to Acoustic Domains

When two acoustic domains are separated by a transfer admittance matrix, the boundary surface integrations with the normal velocity on the S_1 and S_2 surfaces in the "weak" form (Equation 8.8 (p. 255)-Equation 8.10 (p. 255)) are given by:

$$j\omega \iint_{S_1} wv_{n,1} ds_1 = j\omega \iint_{S_1} NN^T ds_1 \{v_{n,1}\} = j\omega [C_1]\{v_{n,1}\} \tag{8.168}$$

$$j\omega \iint_{S_2} wv_{n,2} ds_2 = j\omega \iint_{S_2} NN^T ds_2 \{v_{n,2}\} = j\omega [C_2]\{v_{n,2}\} \tag{8.169}$$

where:

[C] = damping matrix

The outward normal vectors o the integration surface point toward the inside of the transfer admittance matrix. It is consistent with the defined direction of normal velocity in the transfer admittance matrix. Therefore, the velocity vector can be replaced by the pressure via the transfer admittance matrix, while the ports of the transfer admittance matrix are connected to the surface S_1 and S_2 such that:

$$j\omega \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \begin{Bmatrix} v_{n,1} \\ v_{n,2} \end{Bmatrix} = j\omega \begin{bmatrix} Y_{11}[C_1] & Y_{12}[C_1] \\ Y_{21}[C_2] & Y_{22}[C_2] \end{bmatrix} \begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix} + j\omega \begin{Bmatrix} [C_1]\{\alpha_1\} \\ [C_2]\{\alpha_2\} \end{Bmatrix} \tag{8.170}$$

The equivalent damping and stiffness matrices are given by:

$$j\omega[\hat{C}]\{p\} = j\omega \begin{bmatrix} Y_{11,r}[C_1] & Y_{12,r}[C_1] \\ Y_{21,r}[C_2] & Y_{22,r}[C_2] \end{bmatrix} \begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix} \tag{8.171}$$

$$[\hat{K}]\{p\} = -\omega \begin{bmatrix} Y_{11,i}[C_1] & Y_{12,i}[C_1] \\ Y_{21,i}[C_2] & Y_{22,i}[C_2] \end{bmatrix} \begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix} \tag{8.172}$$

where the indices r and I represent the real and imaginary part of a complex variable.

The internal source is cast by:

$$\{\hat{f}\} = -j\omega \begin{Bmatrix} [C_1]\{\alpha_1\} \\ [C_2]\{\alpha_2\} \end{Bmatrix} \tag{8.173}$$

8.7.2. Transfer Admittance Matrix Connected to the Structural and Acoustic Domain

Assume the transfer admittance matrix to represent porous material on a structural surface, i.e. a port of the transfer admittance matrix is connected to the FSI interface.

The "weak" form in the structural domain is given by:

$$\iiint_V \{\bar{\bar{\sigma}}(\bar{u}_S) : \varepsilon(\bar{v}_S) - \omega^2 \rho_S \bar{u}_S \cdot \bar{v}_S\} dv + \oiint_{\Gamma_S} \bar{\bar{\sigma}}(\bar{u}_S)\bar{n}_{F,1} \cdot \bar{v}_S ds_1 = 0 \quad \text{in } \Omega_S \tag{8.174}$$

The "weak" form in the acoustic domain is given by:

$$\iiint_{\Omega_F} \nabla w \cdot \left(\frac{1}{\rho_F}[\Lambda]^{-1} \cdot \nabla p\right) dv - \omega^2 \iiint_{\Omega_F} \frac{1}{\rho_F c^2} w p dv - \oiint_{\Gamma_F} w \frac{1}{\rho_F} \nabla p \cdot \hat{n}_{F,2} ds_2 = j\omega \iiint_{\Omega_F} w\left(\frac{Q}{\rho_F}\right) dv \quad \text{in } \Omega_F \tag{8.175}$$

where:

$\mathbf{n}_{F,1}$ is the outward normal unit vector of the porous material domain and points toward the solid

$\mathbf{n}_{F,2}$ is the outward normal unit vector of the acoustic domain and points toward the porous material

In the fluid domain

$$\frac{1}{\rho_F} \hat{n}_F \cdot \nabla p = -j\omega \hat{n}_F \cdot \vec{v}_F \quad \text{in } \Omega_F$$

The transfer admittance matrix (Equation 8.167 (p. 285)) is also written as:

$$\begin{Bmatrix} \hat{n}_{S,1} \cdot \vec{v}_s \\ \hat{n}_{F,2} \cdot \vec{v}_a \end{Bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{Bmatrix} p_s \\ p_a \end{Bmatrix} + \begin{Bmatrix} \alpha_1 \\ \alpha_2 \end{Bmatrix} \tag{8.176}$$

where:

$\mathbf{n}_{S,1}$ = the outward normal unit vector of the structural domain

p_s = the pressure on the surface S_1 of the structural domain

$\mathbf{v}_s$ = the particle velocity on the surface S_1 of the structural domain

p_a = the pressure on the surface S_2 of the acoustic domain

$\mathbf{v}_a$ = the particle velocity on the surface S_2 of the acoustic domain

The transformation of the transfer admittance matrix in Equation 8.176 (p. 287) is given by:

$$\begin{Bmatrix} p_s \\ \hat{n}_{F,2} \cdot \vec{v}_a \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{Bmatrix} \hat{n}_{S,1} \cdot \vec{v}_s \\ p_a \end{Bmatrix} + \begin{Bmatrix} \beta_1 \\ \beta_2 \end{Bmatrix} \tag{8.177}$$

where:

$$a_{11} = \frac{1}{Y_{11}} \qquad a_{12} = -\frac{Y_{12}}{Y_{11}} \qquad \beta_1 = -\frac{\alpha_1}{Y_{11}}$$

$$a_{21} = \frac{Y_{21}}{Y_{11}} \qquad a_{22} = Y_{22} - \frac{Y_{12}Y_{21}}{Y_{11}} \qquad \beta_2 = \alpha_2 - \alpha_1 \frac{Y_{21}}{Y_{11}}$$

With $\hat{n}_{S,1} = -\hat{n}_{F,1}$ and $\vec{v}_s = j\omega\vec{u}_s$, the transformed transfer admittance matrix is re-written as:

$$\begin{Bmatrix} p_s \\ \hat{n}_{F,2} \cdot \vec{v}_a \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{Bmatrix} -j\omega \hat{n}_{F,1} \cdot \vec{u}_s \\ p_a \end{Bmatrix} + \begin{Bmatrix} \beta_1 \\ \beta_2 \end{Bmatrix} \tag{8.178}$$

Considering the continuous boundary condition on the FSI interface:

$$\bar{\bar{\sigma}}(\vec{u}_S)\vec{n}_F + p_s\vec{n}_F = 0 \qquad \text{on } \Gamma_S \tag{8.179}$$

The boundary integration in the structure "weak" form is given by:

$$\iint_{\Gamma_S} \bar{\bar{\sigma}}(\vec{u}_S)\vec{n}_{F,1} \cdot \vec{v}_S ds_1 = -\iint_{\Gamma_S} \vec{n}_{F,1} \cdot \vec{v}_S [-a_{11} j\omega \vec{n}_{F,1} \cdot \vec{u}_S + a_{12} p_a + \beta_1] ds_1 \tag{8.180}$$

Using Equation 8.8 (p. 255) and Equation 8.9 (p. 255), the boundary integration in the acoustic "weak" form is given by:

$$-\iint_{\Gamma_F} w \frac{1}{\rho_F} \nabla p \cdot \hat{n}_{F,2} ds_2 = j\omega \iint_{\Gamma_F} w[-a_{21} j\omega \hat{n}_{F,1} \cdot \vec{u}_s + a_{22} p_a + \beta_2] ds_2 \tag{8.181}$$

Substituting the boundary integrations into Equation 8.174 (p. 286) and Equation 8.175 (p. 287), the coupled matrix equation is derived as:

$$-\omega^2 \begin{bmatrix} M_S & 0 \\ \bar{\rho}_0 R_{as} & M_F \end{bmatrix} \begin{bmatrix} \vec{u} \\ p \end{bmatrix} + j\omega \begin{bmatrix} C_S + C_{ss} & C_{sa} \\ C_{as} & C_F + C_{aa} \end{bmatrix} \begin{bmatrix} \vec{u} \\ p \end{bmatrix} + \begin{bmatrix} K_S + K_{ss} & R_{sa} \\ 0 & K_F + K_{aa} \end{bmatrix} \begin{bmatrix} \vec{u} \\ p \end{bmatrix} = \begin{bmatrix} f_S + f_{ss} \\ f_F + f_{aa} \end{bmatrix} \tag{8.182}$$

where:

$$[R_{sa}] = -a_{12,r} \iint_{\Gamma_S} \{N_n\}\{N_p\}^T ds_1 \quad ; \quad [C_{sa}] = -\frac{a_{12,i}}{\omega} \iint_{\Gamma_S} \{N_n\}\{N_p\}^T ds_1$$

$$[C_{ss}] = a_{11,r} \iint_{\Gamma_S} \{N_n\}\{N_n\}^T ds_1 \quad ; \quad [K_{ss}] = -\omega a_{11,i} \iint_{\Gamma_S} \{N_n\}\{N_n\}^T ds_1$$

$$\{f_{ss}\} = \beta_1 \iint_{\Gamma_S} \{N_n\} ds_1$$

$$[R_{as}] = -a_{21,r} \iint_{\Gamma_{F,2}} \{N_p\}\{N_n\}^T ds_2 \quad ; \quad [C_{as}] = \bar{\rho}_0 \omega a_{21,i} \iint_{\Gamma_{F,2}} \{N_p\}\{N_n\}^T ds_2$$

$$[C_{aa}] = \bar{\rho}_0 a_{22,r} \iint_{\Gamma_{F,2}} \{N_p\}\{N_p\}^T ds_2 \quad ; \quad [K_{aa}] = -\bar{\rho}_0 \omega a_{22,i} \iint_{\Gamma_{F,2}} \{N_p\}\{N_p\}^T ds_2$$

$$\{f_{aa}\} = -j\bar{\rho}_0 \omega \beta_2 \iint_{\Gamma_{F,2}} \{N_p\} ds_2$$

$$\{N_n\} = [n_{x,F1}N_1, \ldots, n_{x,F1}N_n, \quad n_{y,F1}N_1, \ldots, n_{y,F1}N_n, \quad n_{z,F1}N_1, \ldots, n_{z,F1}N_n]^T$$

$$\{N_p\} = [N_1, N_2, \ldots, N_n]^T = \text{vector of the element pressure shape function}$$

It is impossible to evaluate $\{N_n\}$ on the acoustic surface S_2, since the normal vector $\mathbf{n}_{F,1}$ is involved. An alternative is to evaluate $\{N_n\}$ on the solid surface S_1, for instance:

$$[R_{as}] = -a_{21,r} \iint_{\Gamma_{F,1}} \{N_p\}\{N_n\}^T ds_1 \quad ; \quad [C_{as}] = \bar{\rho}_0 \omega a_{21,i} \iint_{\Gamma_{F,1}} \{N_p\}\{N_n\}^T ds_1$$

Chapter 9: Diffusion

The following topics related to diffusion are available:

9.1. Diffusion Fundamentals

The diffusion process is governed by the first Fick's law:

$$\{J\} = -[D]\nabla C \tag{9.1}$$

It is also governed by the mass balance equation:

$$\nabla \cdot \{J\} + \frac{\partial C}{\partial t} = r \tag{9.2}$$

where:

{J} = diffusion flux vector

$$[D] = \begin{bmatrix} Dxx & 0 & 0 \\ 0 & Dyy & 0 \\ 0 & 0 & Dzz \end{bmatrix} \text{=diffusivity matrix;}$$

Dxx, Dyy, Dzz = diffusivity coefficients in the element X, Y, and Z directions, respectively (input as DXX, DYY, DZZ on **MP** command)

C = C(x,y,z,t) = concentration (CONC degree of freedom)

r = diffusing substance generation rate per unit volume (input as DGEN on **BF** or **BFE** commands)

$$\nabla = \text{ gradient operator } = \left\{ \frac{\partial}{\partial x} \quad \frac{\partial}{\partial y} \quad \frac{\partial}{\partial z} \right\}^T$$

$\nabla \cdot$ = divergence operator

t = time

Substituting Equation 9.1 (p. 289) into Equation 9.2 (p. 289) produces the second Fick's law:

$$\frac{\partial C}{\partial t} = [D]\nabla^2 C + r \tag{9.3}$$

where:

∇^2 = Laplacian operator.

The applicable boundary conditions and loads are:

1. Specified concentration (input as CONC on **D**, **DA**, **DK**, **DL** commands). Initial concentration can also be applied using the CONC label with the **IC** command.

2. Specified diffusion flux r_s acting on a surface (input as DFLUX on **SF** or **SFE** commands):

   $$\{r\}^T\{n\} = -r_s \tag{9.4}$$

   where:

   {n} = unit outward normal vector
   {r} = diffusion flow vector

3. Diffusing substance generation rate r (input as DGEN on **BF** or **BFE** commands).

4. Diffusion flow rate (input as RATE on **F** and **FK** commands).

9.2. Normalized Concentration Approach

Equation 9.3 (p. 289), along with the boundary conditions and loads, governs the diffusion process in a homogeneous domain. In an inhomogeneous domain, different materials have different saturated concentrations. This difference in the saturation levels is responsible for the discontinuity in the concentration across the material interface. To be able to use Equation 9.3 (p. 289) in a finite element analysis, a continuous variable, normalized concentration $\bar{C}$, is introduced [404] (p. 943).

$$\bar{C} = \frac{C}{C_{sat}} \tag{9.5}$$

where C_{sat} is the saturated concentration of the material (input as CSAT on **MP** command).

Substituting the concentration $C = C_{sat}\bar{C}$ from Equation 9.5 (p. 290) into Equation 9.3 (p. 289) produces a governing equation for the diffusion analysis in terms of normalized concentration $\bar{C}$:

$$\frac{\partial\left(C_{sat}\bar{C}\right)}{\partial t} = [D]\nabla^2\left(C_{sat}\bar{C}\right) + r \tag{9.6}$$

When C_{sat} is not specified, it defaults to 1.0 and Equation 9.6 (p. 290) becomes Equation 9.3 (p. 289).

C_{sat} cannot be temperature-dependent.

9.3. Derivation of Diffusion Matrices

The concentration is approximated over the finite element as follows:

$$C = \{N\}^T\{C_e\} \tag{9.7}$$

where:

{N} = element shape functions

$\{C_e\}$ = nodal concentration vector (input/output as CONC)

The application of the variational principle and finite element approximation Equation 9.7 (p. 290) to Equation 9.6 (p. 290) produces the matrix equation as:

$$\left[C^d\right]\left\{\dot{C}_e\right\}+\left[K^d\right]\{C_e\}=\{R_e^{flx}\}+\{R_e^g\}+\{R_e^{nd}\} \tag{9.8}$$

where:

$\left[C^d\right]=C_{sat}\int_{vol}\{N\}\{N\}^T d(vol)$ = element diffusion damping matrix

$\left[K^d\right]=C_{sat}\int_{vol}\left(\nabla\{N\}^T\right)^T[D]\left(\nabla\{N\}^T\right)d(vol)$ = element diffusion conductivity matrix

$\{R_e^{flx}\}=\int_S\{r_s\}\{N\}^T d(S)=$ element diffusion flux vector

$\{R_e^g\}=\int_{vol}\{r\}\{N\}^T d(vol)=$ element diffusing substance generation load vector

vol = element volume

S = element surface

$\left\{R_e^{nd}\right\}=$ nodal diffusion flow rate vector applied to the element (input / output as RATE)

9.4. Diffusion Analysis Results

After the matrix Equation 9.8 (p. 291) is solved for the nodal concentration vector $\{C_e\}$, the diffusion gradient {g}, and the diffusion flux {J} are evaluated at the integration points as follows:

$$\{g\}=C_{sat}\nabla\bar{C} \tag{9.9}$$

$$\{J\}=-[D]\{g\} \tag{9.10}$$

Nodal values of concentration gradient {g} (output as CG) and diffusion flux {J} (output as DF) are computed from the integration points values, as described in Nodal and Centroidal Data Evaluation .

With the normalized concentration approach, the element "true" concentration $C=C_{sat}\bar{C}$ is calculated at the element centroid and output as SMISC,1.

Chapter 10: Coupling

Coupled-field analyses are useful for solving problems where the coupled interaction of phenomena from various disciplines of physical science is significant. Several examples of this include: an electric field interacting with a magnetic field, a magnetic field producing structural forces, a temperature field influencing fluid flow, a temperature field giving rise to thermal strains and the usual influence of temperature dependent material properties. The latter two examples can be modeled with most non-coupled-field elements, as well as with coupled-field elements.

The following coupled-field topics are available:

10.1. Coupled Effects

The following topics concerning coupled effects are available:

10.1.1. Elements

The following elements have coupled-field capability:

Table 10.1: Elements Used for Coupled Effects

SOLID5	3-D Coupled-Field Solid (Derivation of Electromagnetic Matrices (p. 193), Coupled Effects (p. 293), SOLID5 - 3-D Coupled-Field Solid (p. 411))
PLANE13	2-D Coupled-Field Solid (Derivation of Electromagnetic Matrices (p. 193), Coupled Effects (p. 293), SOLID5 - 3-D Coupled-Field Solid (p. 411))
FLUID29	2-D Acoustic Fluid (Derivation of Acoustic Matrices (p. 256), FLUID29 - 2-D Acoustic Fluid (p. 429))
FLUID30	3-D 8-Node Acoustic Fluid (Derivation of Acoustic Matrices (p. 256), FLUID30 - 3-D Acoustic Fluid (p. 430))
PLANE53	2-D 8-Node Magnetic Solid (Derivation of Electromagnetic Matrices (p. 193), Electromagnetic Field Evaluations (p. 200), PLANE53 - 2-D 8-Node Magnetic Solid (p. 460))
LINK68	Coupled Thermal-Electric Line (LINK68 - Coupled Thermal-Electric Line (p. 476))
SOLID97	3-D Magnetic Solid (SOLID97 - 3-D Magnetic Solid (p. 491))

SOLID98	Tetrahedral Coupled-Field Solid (Derivation of Electromagnetic Matrices (p. 193), Coupled Effects (p. 293), SOLID98 - Tetrahedral Coupled-Field Solid (p. 492))
FLUID116	Coupled Thermal-Fluid Pipe (FLUID116 - Coupled Thermal-Fluid Pipe (p. 500))
CIRCU124	Electric Circuit Element (Stranded Coil Analyses (p. 210), CIRCU124 - Electric Circuit (p. 508))
TRANS126	Electromechanical Transducer (Capacitance Computation (p. 220), Review of Coupled Electromechanical Methods (p. 321), TRANS126 - Electromechanical Transducer (p. 513))
SHELL157	Coupled Thermal-Electric Shell (SHELL157 - Thermal-Electric Shell (p. 543))
PLANE223	2-D 8-Node Coupled-Field Solid (PLANE223 - 2-D 8-Node Coupled-Field Solid (p. 621))
FLUID220	3-D 20-Node Acoustic Fluid
FLUID221	3-D 10-Node Acoustic Fluid
SOLID226	3-D 20-Node Coupled-Field Solid (SOLID226 - 3-D 20-Node Coupled-Field Solid (p. 623))
SOLID227	3-D 10-Node Coupled-Field Solid (SOLID227 - 3-D 10-Node Coupled-Field Solid (p. 625))
PLANE233	2-D 8-Node Electromagnetic Solid (Derivation of Electromagnetic Matrices (p. 193), Electromagnetic Field Evaluations (p. 200), PLANE233 - 2-D 8-Node Electromagnetic Solid (p. 628))
SOLID236	3-D 20-Node Electromagnetic Solid (Derivation of Electromagnetic Matrices (p. 193), Electromagnetic Field Evaluations (p. 200), SOLID236 - 3-D 20-Node Electromagnetic Solid (p. 629))
SOLID237	3-D 10-Node Electromagnetic Solid (Derivation of Electromagnetic Matrices (p. 193), Electromagnetic Field Evaluations (p. 200), SOLID237 - 3-D 10-Node Electromagnetic Solid (p. 630))

There are certain advantages and disadvantages inherent with coupled-field formulations:

10.1.1.1. Advantages

1. Allows for solutions to problems otherwise not possible with usual finite elements.
2. Simplifies modeling of coupled-field problems by permitting one element type to be used in a single analysis pass.

10.1.1.2. Disadvantages

1. Increases problem size (unless a segregated solver is used).
2. Inefficient matrix reformulation (if a section of a matrix associated with one phenomena is reformed, the entire matrix will be reformed).
3. Larger storage requirements.

10.1.2. Coupling Methods

There are basically two methods of coupling distinguished by the finite element formulation techniques used to develop the matrix equations. These are illustrated here with two types of degrees of freedom ($\{X_1\}$, $\{X_2\}$):

1. Strong (also matrix, simultaneous, or full) coupling - where the matrix equation is of the form:

$$\begin{bmatrix} [K_{11}] & [K_{12}] \\ [K_{21}] & [K_{22}] \end{bmatrix} \begin{Bmatrix} \{X_1\} \\ \{X_2\} \end{Bmatrix} = \begin{Bmatrix} \{F_1\} \\ \{F_2\} \end{Bmatrix} \tag{10.1}$$

   and the coupled effect is accounted for by the presence of the off-diagonal submatrices $[K_{12}]$ and $[K_{21}]$. This method provides for a coupled response in the solution after one iteration.

2. Weak (also load vector or sequential) coupling - where the coupling in the matrix equation is shown in the most general form:

$$\begin{bmatrix} [K_{11}(\{X_1\},\{X_2\})] & [0] \\ [0] & [K_{22}(\{X_1\},\{X_2\})] \end{bmatrix} \begin{Bmatrix} \{X_1\} \\ \{X_2\} \end{Bmatrix} = \begin{Bmatrix} \{F_1(\{X_1\},\{X_2\})\} \\ \{F_2(\{X_1\},\{X_2\})\} \end{Bmatrix} \tag{10.2}$$

   and the coupled effect is accounted for in the dependency of $[K_{11}]$ and $\{F_1\}$ on $\{X_2\}$ as well as $[K_{22}]$ and $\{F_2\}$ on $\{X_1\}$. At least two iterations are required to achieve a coupled response.

The following is a list of the types of coupled-field analyses including methods of coupling present in each:

Table 10.2: Coupling Methods

Analysis Category	Coupling Method Used	Example Applications
Thermal-Structural Analysis (p. 296)	S, W	High temperature turbine
Magneto-Structural Analysis (Vector Potential) (p. 297)	W	Solenoid, high energy magnets (MRI)
Magneto-Structural Analysis (Scalar Potential) (p. 297)		
Electromagnetic Analysis (p. 298)	S	Current fed massive conductors
Electro-Thermo-Structural Analysis (p. 299)	W	Electro-thermal MEMS actuators
Electro-Magneto-Thermo-Structural Analysis (p. 299)	W	Direct current electromechanical devices in general
Electro-Magneto-Thermal Analysis (p. 300)		
Piezoelectric Analysis (p. 300)	S	Transducers, resonators
Electroelastic Analysis (p. 301)	W	Dielectric elastomers
Thermo-Piezoelectric Analysis (p. 301)	S, W	Sensors and actuators for smart structures
Piezoresistive Analysis (p. 302)	W	Pressure and force sensors
Thermo-Pressure Analysis (p. 303)	S, W	Piping networks
Acoustic-Structural Analysis (p. 303)	S, W	Acoustics
Thermo-Electric Analysis (p. 303)	S, W	High temperature electronics, Peltier coolers, thermoelectric generators
Magnetic-Thermal Analysis (p. 304)	W	Direct current transients: power interrupts, surge protection

Analysis Category	Coupling Method Used	Example Applications
Circuit-Magnetic Analysis (p. 304)	S	Circuit-fed solenoids, transformers, and motors
Structural-Diffusion Analysis (p. 304)	S, W	Hygroscopic swelling of polymers in electronics packaging
Thermal-Diffusion Analysis (p. 305)	W	Temperature-dependent moisture migration
Structural-Thermal-Diffusion Analysis (p. 305)	S, W	Sodium expansion in aluminum reduction cells

where:

S = strong coupling
W = weak coupling

The solution sequence follows the standard finite element methodology. Convergence is achieved when changes in all unknowns (i.e. DOF) and knowns, regardless of units, are less than the values specified (on the **CNVTOL** command). Some of the coupling described above is always or usually one-way. For example, in Category A, the temperatures affect the displacements of the structure by way of the thermal strains, but the displacements usually do not affect the temperatures.

The following descriptions of coupled phenomena will include:

1. Applicable element types
2. Basic matrix equation indicating coupling terms in bold print. In addition to the terms indicated in bold print, any equation with temperature as a degree of freedom can have temperature-dependency in all terms.
3. Applicable analysis types, including the matrix and/or vector terms possible in each analysis type.

The nomenclature used on the following pages is given in Table 10.3: Nomenclature of Coefficient Matrices (p. 306) at the end of the section. In some cases, element KEYOPTS are used to select the DOF of the element. DOF will not be fully active unless the appropriate material properties are specified. Some of the elements listed may not be applicable for a particular use as it may be only 1-D, whereas a 3-D element is needed (e.g. FLUID116).

10.1.2.1. Thermal-Structural Analysis

(see Derivation of Structural Matrices (p. 12), Derivation of Heat Flow Matrices (p. 235), and Thermoelasticity (p. 309))

1. Element type: SOLID5, PLANE13, SOLID98, PLANE223, SOLID226, SOLID227
2. Matrix equation:

(a). Strong coupling

$$\begin{bmatrix} [M] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\dot{T}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [C^{tu}] & [C^{t}] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{T}\} \end{Bmatrix} + \begin{bmatrix} [K] & [K^{ut}] \\ [0] & [K^{t}] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{T\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{Q\} \end{Bmatrix} \quad (10.3)$$

(b). Weak coupling

$$\begin{bmatrix}[M] & [0]\\ [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{T}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0]\\ [0] & [C^t]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{T}\}\end{Bmatrix}+\begin{bmatrix}[K] & [0]\\ [0] & [K^t]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{T\}\end{Bmatrix}=\begin{Bmatrix}\{F\}+\{F^{th}\}\\ \{Q\}+\{Q^{ted}\}\end{Bmatrix} \tag{10.4}$$

where:

$[K^t] = [K^{tb}] + [K^{tc}]$
$\{F\} = \{F^{nd}\} + \{F^{pr}\} + \{F^{ac}\}$
$\{Q\} = \{Q^{nd}\} + \{Q^{g}\} + \{Q^{c}\}$

3. Analysis types:

(a). Strong coupling: static, transient, or harmonic

(b). Weak coupling: static or transient

Note

Strong coupling is supported only by PLANE223, SOLID226, and SOLID227.

$\{Q^{ted}\}$ is applicable to only PLANE223, SOLID226, and SOLID227.

10.1.2.2. Magneto-Structural Analysis (Vector Potential)

(see Derivation of Electromagnetic Matrices (p. 193) and Piezoelectrics (p. 313))

1. Element type: PLANE13

2. Matrix equation:

$$\begin{bmatrix}[M] & [0]\\ [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{A}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0]\\ [0] & [C^m]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{A}\}\end{Bmatrix}+\begin{bmatrix}[K] & [0]\\ [0] & [K^m]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{A\}\end{Bmatrix}=\begin{Bmatrix}\{F\}\\ \{\psi_i\}\end{Bmatrix} \tag{10.5}$$

where:

$\{F\} = \{F^{nd}\} + \{F^{pr}\} + \{F^{ac}\} + \{F^{th}\} + \{F^{jb}\} + \{F^{mx}\}$
$\{\Psi_i\} = \{\psi_i^{nd}\} + \{\psi^{s}\} + \{\psi^{pm}\}$

3. Analysis types: Static or Transient

10.1.2.3. Magneto-Structural Analysis (Scalar Potential)

1. Element type: SOLID5, SOLID98

2. Matrix equation:

$$\begin{bmatrix}[K] & [0]\\ [0] & [K^m]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{\phi\}\end{Bmatrix}=\begin{Bmatrix}\{F\}\\ \{\psi_f\}\end{Bmatrix} \tag{10.6}$$

where:

$$\{F\} = \{F^{nd}\} + \{F^{pr}\} + \{F^{ac}\} + \{F^{th}\} + \{F^{mx}\}$$

$$\{\Psi_f\} = \{\psi_f^{nd}\} + \{\psi^{b}\} + \{\psi^{pm}\}$$

3. Analysis types: Static

10.1.2.4. Electromagnetic Analysis

(see Derivation of Electromagnetic Matrices (p. 193))

1. Element type: PLANE13, PLANE53, SOLID97, PLANE233, SOLID236, SOLID237

2. Matrix equation:

   (a) Time-integrated electric potential formulation

$$\begin{bmatrix} [C^{AA}] & [C^{Av}] \\ [C^{Av}]^T & [C^{vv}] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{\dot{v}\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{v\} \end{Bmatrix} = \begin{Bmatrix} \{\psi_i\} \\ \{I\} \end{Bmatrix} \tag{10.7}$$

   where:

$$\{\Psi_i\} = \{\psi_i^{nd}\} + \{\psi^{s}\} + \{\psi^{pm}\}$$

$$\{I\} = \{I^{nd}\}$$

   The above formulation is used with PLANE13, PLANE53, and SOLID97. It is also available with KEYOPT(2) = 2 of PLANE233, SOLID236, and SOLID237.

   (b) Electric potential formulation

$$\begin{bmatrix} [M^{AA}] & [0] \\ [C^{AV}]^T & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{A}\} \\ \{\ddot{V}\} \end{Bmatrix} + \begin{bmatrix} [C^{AA}] & [C^{AV}] \\ [K^{AV}]^T & [C^{VV}] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [K^{AV}] \\ [0] & [K^{VV}] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{\Psi_i\} \\ \{I\} \end{Bmatrix} \tag{10.8}$$

   The above formulation is the default with PLANE233, SOLID236, and SOLID237.

3. Analysis types: Static, harmonic or transient

10.1.2.5. Stranded Coil Analysis

(see Stranded Coil Analysis)

1. Element type: PLANE53, SOLID97, PLANE233, SOLID236, SOLID237

2. Matrix equation:

   (a) A-VOLT-EMF formulation

$$\begin{bmatrix} [0] & [0] & [0] \\ s[K^{AV}]^T & [0] & [0] \\ -[K^{AV}]^T & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{A}_e\} \\ \{\dot{V}\} \\ \{\dot{e}\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [K^{AV}] & -[K^{AV}] \\ [0] & s[K^{VV}] & [0] \\ [0] & [0] & -[K^{VV}] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{V\} \\ \{e\} \end{Bmatrix} = \begin{Bmatrix} \{\Psi^{pm}\} \\ \{I\} \\ \{0\} \end{Bmatrix} \tag{10.9}$$

where:

$\{I\} = \{I^{nd}\}$

The above formulation is used with PLANE233, SOLID236, and SOLID237.

(b) A-CURR formulation

$$\begin{bmatrix} [0] & [0] \\ [C^{iA}] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{0\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [K^{Ai}] \\ [0] & [K^{ii}] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{i\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{V_0\} \end{Bmatrix} \tag{10.10}$$

The above formulation is used with KEYOPT(1) = 2 (voltage-fed stranded coil) of PLANE53 and SOLID97. For the circuit-coupled stranded coil with PLANE53 and SOLID97, see section 11.1.2.17 Circuit-Magnetic Analysis.

3. Analysis types: Static, harmonic or transient

10.1.2.6. Electro-Thermo-Structural Analysis

(see Derivation of Structural Matrices (p. 12), Derivation of Heat Flow Matrices (p. 235), Thermoelasticity (p. 309), and Thermoelectrics (p. 319))

1. Element type: PLANE223, SOLID226, SOLID227

2. Matrix equation

$$\begin{bmatrix} [M] & [0] & [0] \\ [0] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{T}\} \\ \{\ddot{V}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] & [0] \\ [C^{tu}] & [C^{t}] & [0] \\ [0] & [0] & [C^{v}] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{T}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} [K] & [K^{ut}] & [0] \\ [0] & [K^{t}] & [0] \\ [0] & [K^{vt}] & [K^{v}] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{T\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{Q\} \\ \{I\} \end{Bmatrix} \tag{10.11}$$

where:

$[K^t] = [K^{tb}] + [K^{tc}]$
$\{F\} = \{F^{nd}\} + \{F^{pr}\} + \{F^{ac}\}$
$\{Q\} = \{Q^{nd}\} + \{Q^{g}\} + \{Q^{c}\} + \{Q^{j}\} + \{Q^{p}\}$
$\{I\} = \{I^{nd}\}$

3. Analysis types: static and transient

10.1.2.7. Electro-Magneto-Thermo-Structural Analysis

(see Derivation of Electromagnetic Matrices (p. 193) and Piezoelectrics (p. 313))

1. Element types: SOLID5, SOLID98

2. Matrix equation:

$$\begin{bmatrix} [M] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{T}\} \\ \{\ddot{V}\} \\ \{\ddot{\phi}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] & [0] & [0] \\ [0] & [C^t] & [0] & [0] \\ [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{T}\} \\ \{\dot{V}\} \\ \{\dot{\phi}\} \end{Bmatrix}$$
$$+ \begin{bmatrix} [K] & [0] & [0] & [0] \\ [0] & [K^t] & [0] & [0] \\ [0] & [0] & [K^v] & [0] \\ [0] & [0] & [0] & [K^m] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{T\} \\ \{V\} \\ \{\phi\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{Q\} \\ \{I\} \\ \{\psi_f\} \end{Bmatrix} \tag{10.12}$$

where:

$$[K^t] = [K^{tb}] + [K^{tc}]$$
$$\{F\} = \{F^{nd}\} + \{F^{th}\} + \{F^{ac}\} + \{F^{jb}\} + \{F^{pr}\} + \{F^{mx}\}$$
$$\{Q\} = \{Q^{nd}\} + \{Q^{g}\} + \{Q^{j}\} + \{Q^{c}\}$$
$$\{I\} = \{I^{nd}\}$$
$$\{\Psi_f\} = \{\psi_f^{nd}\} + \{\psi^{g}\} + \{\psi^{pm}\}$$

3. Analysis types: Static or Transient

10.1.2.8. Electro-Magneto-Thermal Analysis

(see Derivation of Electromagnetic Matrices (p. 193))

1. Element types: SOLID5, SOLID98

2. Matrix equation:

$$\begin{bmatrix} [C^t] & [0] & [0] \\ [0] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{T}\} \\ \{\dot{V}\} \\ \{\dot{\phi}\} \end{Bmatrix} + \begin{bmatrix} [K^t] & [0] & [0] \\ [0] & [K^v] & [0] \\ [0] & [0] & [K^m] \end{bmatrix} \begin{Bmatrix} \{Q\} \\ \{I\} \\ \{\psi_f\} \end{Bmatrix} \tag{10.13}$$

where:

$$[K^t] = [K^{tb}] + [K^{tc}]$$
$$\{Q\} = \{Q^{nd}\} + \{Q^{g}\} + \{Q^{j}\} + \{Q^{c}\}$$
$$\{I\} = \{I^{nd}\}$$
$$\{\Psi_f\} = \{\psi_f^{nd}\} + \{\psi^{g}\} + \{\psi^{pm}\}$$

3. Analysis types: Static or Transient

10.1.2.9. Piezoelectric Analysis

(see Piezoelectrics (p. 313))

1. Element types: SOLID5, PLANE13, SOLID98, PLANE223, SOLID226, and SOLID227.

2. Matrix equation:

$$\begin{bmatrix}[M] & [0]\\ [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{V}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0]\\ [0] & -[C^{vh}]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{V}\}\end{Bmatrix}+\begin{bmatrix}[K] & [K^{z}]\\ [K^{z}]^{T} & -[K^{d}]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{V\}\end{Bmatrix}=\begin{Bmatrix}\{F\}\\ \{L\}+\{L^{th}\}\end{Bmatrix} \tag{10.14}$$

where:

$\{F\} = \{F^{nd}\} + \{F^{th}\} + \{F^{ac}\} + \{F^{pr}\}$
$\{L\} = \{L^{nd}\} + \{L^{c}\} + \{L^{sc}\}+\{L^{th}\}$

Note

$\{L^{c}\}$ and $\{L^{sc}\}$ are applicable to only PLANE223, SOLID226, and SOLID227.

3. Analysis types: Static, modal, harmonic, or transient

10.1.2.10. Electroelastic Analysis

(see Electroelasticity (p. 317))

1. Element types: PLANE223, SOLID226, and SOLID227.

2. Matrix equation:

$$\begin{bmatrix}[M] & [0]\\ [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{v}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0]\\ [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{v}\}\end{Bmatrix}+\begin{bmatrix}[K] & [0]\\ [0] & [K^{d}]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{v\}\end{Bmatrix}=\begin{Bmatrix}\{F\}\\ \{L\}\end{Bmatrix} \tag{10.15}$$

where:

$\{F\} = \{F^{nd}\} + \{F^{th}\} + \{F^{ac}\} + \{F^{pr}\} + \{F^{e}\}$
$\{L\} = \{L^{nd}\} + \{L^{c}\} + \{L^{sc}\}$

3. Analysis types: Static or transient

10.1.2.11. Thermo-Piezoelectric Analysis

(see Derivation of Structural Matrices (p. 12), Derivation of Heat Flow Matrices (p. 235), Thermoelasticity (p. 309), and Piezoelectrics (p. 313))

1. Element type: PLANE223, SOLID226, SOLID227

2. Matrix equation:

   a. Strong coupling:

$$\begin{bmatrix}[M] & [0] & [0]\\ [0] & [0] & [0]\\ [0] & [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{T}\}\\ \{\ddot{V}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0] & [0]\\ [C^{tu}] & [C^{t}] & [0]\\ [0] & [0] & -[C^{vh}]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{T}\}\\ \{\dot{V}\}\end{Bmatrix}$$
$$+\begin{bmatrix}[K] & [K^{ut}] & [K^{z}]\\ [0] & [K^{t}] & [0]\\ [K^{z}]^{T} & [K^{zt}] & -[K^{d}]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{T\}\\ \{V\}\end{Bmatrix}=\begin{Bmatrix}\{F\}\\ \{Q\}\\ \{L\}\end{Bmatrix} \tag{10.16}$$

b. Weak coupling:

$$\begin{bmatrix}[M] & [0] & [0]\\ [0] & [0] & [0]\\ [0] & [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{T}\}\\ \{\ddot{V}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0] & [0]\\ [0] & [C^{t}] & [0]\\ [0] & [0] & -[C^{vh}]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{T}\}\\ \{\dot{V}\}\end{Bmatrix}$$
$$+\begin{bmatrix}[K] & [0] & [K^{z}]\\ [0] & [K^{t}] & [0]\\ [K^{z}]^{T} & [0] & -[K^{d}]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{T\}\\ \{V\}\end{Bmatrix}=\begin{Bmatrix}\{F\}+\{F^{th}\}\\ \{Q\}+\{Q^{ted}\}\\ \{L\}+\{L^{th}\}\end{Bmatrix} \tag{10.17}$$

where:

$[K^{t}] = [K^{tb}] + [K^{tc}]$
$\{F\} = \{F^{nd}\} + \{F^{pr}\} + \{F^{ac}\}$
$\{Q\} = \{Q^{nd}\} + \{Q^{g}\} + \{Q^{c}\}$
$\{L\} = \{L^{nd}\} + \{L^{c}\} + \{L^{sc}\}$

3. Analysis types:

   a. Strong coupling: static, transient, harmonic, modal

   b. Weak coupling: static or transient

10.1.2.12. Piezoresistive Analysis

(see Piezoresistivity (p. 318))

1. Element type: PLANE223, SOLID226, SOLID227

2. Matrix equation:

$$\begin{bmatrix}[M] & [0]\\ [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{V}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0]\\ [0] & [C^{v}]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{V}\}\end{Bmatrix}+\begin{bmatrix}[K] & [0]\\ [0] & [K^{v}]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{V\}\end{Bmatrix}=\begin{Bmatrix}\{F\}\\ \{I\}\end{Bmatrix} \tag{10.18}$$

where:

$[K^{v}]$ = conductivity matrix (see Equation 10.61 (p. 319)) updated for piezoresistive effects
$\{F\} = \{F^{nd}\} + \{F^{th}\} + \{F^{pr}\} + \{F^{ac}\}$
$\{I\} = \{I^{nd}\}$

3. Analysis types: Static or transient

10.1.2.13. Thermo-Pressure Analysis

(see FLUID116 - Coupled Thermal-Fluid Pipe (p. 500))

1. Element type: FLUID116

2. Matrix equation:

$$\begin{bmatrix} [C^t] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{T}\} \\ \{\dot{P}\} \end{Bmatrix} + \begin{bmatrix} [K^t] & [0] \\ [0] & [K^p] \end{bmatrix} \begin{Bmatrix} \{T\} \\ \{P\} \end{Bmatrix} = \begin{Bmatrix} \{Q\} \\ \{W\} \end{Bmatrix} \tag{10.19}$$

   where:

$$[K^t] = [K^{tb}] + [K^{tc}] + [K^{tm}]$$
$$\{Q\} = \{Q^{nd}\} + \{Q^c\} + \{Q^g\}$$
$$\{W\} = \{W^{nd}\} + \{W^h\}$$

3. Analysis types: Static or Transient

10.1.2.14. Acoustic-Structural Analysis

(See Derivation of Acoustic Matrices (p. 256).)

1. Element type: FLUID29, FLUID30, FLUID220, and FLUID221 (with other structural elements)

2. Matrix equation:

$$\begin{bmatrix} [M] & [0] \\ [M^{fs}] & [M^p] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{P}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [0] & [C^p] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{P}\} \end{Bmatrix} + \begin{bmatrix} [K] & [K^{fs}] \\ [0] & [K^p] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{P\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{W\} \end{Bmatrix} \tag{10.20}$$

   Values for [M], [C], and [K] are provided by other elements.

3. Analysis types: Transient, harmonic and modal analyses can be performed.

10.1.2.15. Thermo-Electric Analysis

1. Element types: SOLID5, LINK68, SOLID98, SHELL157, PLANE223, SOLID226, and SOLID227

2. Matrix equation:

$$\begin{bmatrix} [C^t] & [0] \\ [0] & [C^v] \end{bmatrix} \begin{Bmatrix} \{\dot{T}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} [K^t] & [0] \\ [K^{vt}] & [K^v] \end{bmatrix} \begin{Bmatrix} \{T\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{Q\} \\ \{I\} \end{Bmatrix} \tag{10.21}$$

   where:

$$[K^t] = [K^{tb}] + [K^{tc}]$$
$$\{Q\} = \{Q^{nd}\} + \{Q^c\} + \{Q^g\} + \{Q^j\} + \{Q^p\}$$

$\{I\} = \{I^{nd}\}$

Note

$\{Q^p\}$, $[K^{vt}]$, and $[C^v]$ are used only for PLANE223, SOLID226, and SOLID227.

3. Analysis types: Static or Transient

10.1.2.16. Magnetic-Thermal Analysis

(see Derivation of Electromagnetic Matrices (p. 193))

1. Element type: PLANE13

2. Matrix equation:

$$\begin{bmatrix} [C^{AA}] & [0] \\ [0] & [C^t] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{\dot{T}\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [0] \\ [0] & [K^t] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{T\} \end{Bmatrix} = \begin{Bmatrix} \{\psi_i\} \\ \{Q\} \end{Bmatrix} \tag{10.22}$$

where:

$$[K^t] = [K^{tb}] + [K^{tc}]$$
$$\{\Psi_i\} = \{\psi_i^{nd}\} + \{\psi^s\} + \{\psi^{pm}\}$$
$$\{Q\} = \{Q^{nd}\} + \{Q^g\} + \{Q^j\} + \{Q^c\}$$

3. Analysis types: Static or Transient

10.1.2.17. Circuit-Magnetic Analysis

(See Stranded Coil Analyses (p. 210))

1. Element type: PLANE53, SOLID97, CIRCU124

2. Matrix equation:

$$\begin{bmatrix} [0] & [0] & [0] \\ [C^{iA}] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{0\} \\ \{0\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [K^{Ai}] & [0] \\ [0] & [K^{ii}] & [K^{ie}] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{i\} \\ \{e\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{0\} \\ \{0\} \end{Bmatrix} \tag{10.23}$$

3. Analysis types: Static, Transient, or Harmonic

10.1.2.18. Structural-Diffusion Analysis

(see Derivation of Structural Matrices (p. 12), Derivation of Diffusion Matrices (p. 290), and Structural-Diffusion Coupling (p. 323))

1. Element type: PLANE223, SOLID226, SOLID227

2. Matrix equation:

a. Strong coupling:

$$\begin{bmatrix}[M] & [0]\\ [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{C}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0]\\ [0] & [C^d]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{C}\}\end{Bmatrix}+\begin{bmatrix}[K] & [K^{ud}]\\ [0] & [K^d]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{C\}\end{Bmatrix}=\begin{Bmatrix}\{F\}\\ \{R\}\end{Bmatrix} \tag{10.24}$$

b. Weak coupling:

$$\begin{bmatrix}[M] & [0]\\ [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{C}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0]\\ [0] & [C^d]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{C}\}\end{Bmatrix}+\begin{bmatrix}[K] & [0]\\ [0] & [K^d]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{C\}\end{Bmatrix}=\begin{Bmatrix}\{F\}+\{F^{di}\}\\ \{R\}\end{Bmatrix} \tag{10.25}$$

where:

$\{F\} = \{F^{nd}\}+ \{F^{pr}\}+ \{F^{ac}\}$

3. Analysis types: static and transient

10.1.2.19. Thermal-Diffusion Analysis

(see Derivation of Heat Flow Matrices (p. 235), Derivation of Diffusion Matrices (p. 290))

1. Element type: PLANE223, SOLID226, SOLID227

2. Matrix equation:

$$\begin{bmatrix}[C^t] & [0]\\ [0] & [C^d]\end{bmatrix}\begin{Bmatrix}\{\dot{T}\}\\ \{\dot{C}\}\end{Bmatrix}+\begin{bmatrix}[K^t] & [0]\\ [0] & [K^d]\end{bmatrix}\begin{Bmatrix}\{T\}\\ \{C\}\end{Bmatrix}=\begin{Bmatrix}\{Q\}\\ \{R\}\end{Bmatrix} \tag{10.26}$$

where:

$[Kt] = [Ktb] + [Ktc]$

$\{Q\} = \{Q^{nd}\}+ \{Q^{c}\}+ \{Q^{g}\}+ \{Q^{p}\}$

3. Analysis types: static and transient

10.1.2.20. Structural-Thermal-Diffusion Analysis

(see Derivation of Structural Matrices (p. 12), Derivation of Heat Flow Matrices (p. 235), Derivation of Diffusion Matrices (p. 290), Thermoelasticity (p. 309), and Structural-Diffusion Coupling (p. 323))

1. Element type: PLANE223, SOLID226, SOLID227

2. Matrix equation

a. Strong coupling:

$$\begin{bmatrix}[M] & [0] & [0]\\ [0] & [0] & [0]\\ [0] & [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{T}\}\\ \{\ddot{C}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0] & [0]\\ [C^{tu}] & [C^t] & [0]\\ [0] & [0] & [C^d]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{T}\}\\ \{\dot{C}\}\end{Bmatrix}+\begin{bmatrix}[K] & [K^{ut}] & [K^{ud}]\\ [0] & [K^t] & [0]\\ [0] & [0] & [K^d]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{T\}\\ \{C\}\end{Bmatrix}=\begin{Bmatrix}\{F\}\\ \{Q\}\\ \{R\}\end{Bmatrix} \tag{10.27}$$

b. Weak coupling:

$$\begin{bmatrix} [M] & [0] & [0] \\ [0] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{T}\} \\ \{\ddot{C}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] & [0] \\ [0] & [C^t] & [0] \\ [0] & [0] & [C^d] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{T}\} \\ \{\dot{C}\} \end{Bmatrix} + \begin{bmatrix} [K] & [0] & [0] \\ [0] & [K^t] & [0] \\ [0] & [0] & [K^d] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{T\} \\ \{C\} \end{Bmatrix} = \begin{Bmatrix} \{F\} + \{F^{th}\} + \{F^{di}\} \\ \{Q\} + \{Q^{ted}\} \\ \{R\} \end{Bmatrix} \quad (10.28)$$

where:

$\{F\} = \{F^{nd}\} + \{F^{pr}\} + \{F^{ac}\}$

$\{Q\} = \{Q^{nd}\} + \{Q^{c}\} + \{Q^{g}\} + \{Q^{p}\}$

3. Analysis type: static and transient

Table 10.3: Nomenclature of Coefficient Matrices

Symbol	Meaning	Usage
[M]	structural mass matrix (discussed in Derivation of Structural Matrices (p. 12))	[1]
$[M^{fs}]$	fluid-structure coupling mass matrix (discussed in Derivation of Acoustic Matrices (p. 256))	[1]
$[M^{p}]$	acoustic mass matrix (discussed in Derivation of Acoustic Matrices (p. 256))	[1]
[C]	structural damping matrix (discussed in Derivation of Structural Matrices (p. 12))	[2]
$[C^t]$	thermal specific heat matrix (discussed in Derivation of Heat Flow Matrices (p. 235))	[2]
$[C^{tu}]$	thermoelastic damping matrix (discussed in Thermoelasticity (p. 309))	[2]
$[C^{AA}]$	magnetic damping matrix (discussed in Electromagnetic Field Evaluations (p. 200))	[2]
$[C^{p}]$	acoustic damping matrix (discussed in Derivation of Acoustic Matrices (p. 256))	[2]
$[C^{Av}]$	magnetic-electric damping matrix (discussed in Derivation of Electromagnetic Matrices (p. 193))	[2]
$[C^{vv}]$	electric damping matrix (discussed in Derivation of Electromagnetic Matrices (p. 193))	[2]
$[C^{iA}]$	inductive damping matrix (discussed in Stranded Coil Analyses (p. 210))	[2]
$[C^{v}]$	dielectric permittivity coefficient matrix (discussed in Quasistatic Electric Analysis (p. 199))	[2]
$[C^{vh}]$	dielectric damping matrix (discussed in Quasistatic Electric Analysis (p. 199))	[2]
[K]	structural stiffness matrix (discussed in Derivation of Structural Matrices (p. 12))	[3]
$[K^t]$	thermal conductivity matrix (may consist of 1, 2, or 3 of the following 3 matrices) (discussed in Derivation of Heat Flow Matrices (p. 235))	[3]
$[K^{tb}]$	thermal conductivity matrix of material (discussed in Derivation of Heat Flow Matrices (p. 235))	[3]
$[K^{tc}]$	thermal conductivity matrix of convection surface (discussed in Derivation of Heat Flow Matrices (p. 235))	[3]

Symbol	Meaning	Usage
$[K^{tm}]$	thermal conductivity matrix associated with mass transport (discussed in Derivation of Heat Flow Matrices (p. 235))	[3]
$[K^{ut}]$	thermoelastic stiffness matrix (discussed in Thermoelasticity (p. 309))	[3]
$[K^{m}]$	scalar magnetic potential coefficient matrix (discussed in Derivation of Electromagnetic Matrices (p. 193))	[3]
$[K^{AA}]$	vector magnetic potential coefficient matrix (discussed in Derivation of Electromagnetic Matrices (p. 193))	[3]
$[K^{Ai}]$	potential-current coupling stiffness matrix (discussed in Stranded Coil Analyses (p. 210))	[3]
$[K^{ii}]$	resistive stiffness matrix (discussed in Stranded Coil Analyses (p. 210))	[3]
$[K^{ie}]$	current-emf coupling stiffness (discussed in Stranded Coil Analyses (p. 210))	[3]
$[K^{v}]$	electrical conductivity coefficient matrix (discussed in Derivation of Electromagnetic Matrices (p. 193))	[3]
$[K^{z}]$	piezoelectric stiffness matrix (discussed in Piezoelectrics (p. 313))	[3]
$[K^{zt}]$	thermo-piezoelectric stiffness matrix (discussed in Piezoelectrics (p. 313))	[3]
$[K^{d}]$	dielectric coefficient matrix (discussed in Piezoelectrics (p. 313))	[3]
$[K^{vt}]$	Seebeck coefficient coupling matrix	[3]
$[K^{ud}]$	Diffision-elastic stiffness matrix (discussed in Structural-Diffusion Coupling (p. 323))	[3]

1. Coefficient matrices of second time derivatives of unknowns.
2. Coefficient matrices of first time derivative of unknowns
3. Coefficient matrices of unknowns

Vectors of Knowns

Symbol	Meaning	Associated Input / Output Label
$\{F^{nd}\}$	applied nodal force vector (discussed in Derivation of Structural Matrices (p. 12))	FX ... MZ
$\{F^{nr}\}$	Newton-Raphson restoring load vector (discussed in Newton-Raphson Procedure (p. 711)	FX ... MZ
$\{F^{th}\}$	thermal strain force vector (discussed in Derivation of Structural Matrices (p. 12))	FX ... MZ
$\{F^{pr}\}$	pressure load vector (discussed in Derivation of Structural Matrices (p. 12))	FX ... MZ
$\{F^{ac}\}$	force vector due to acceleration effects (i.e., gravity) (discussed in Derivation of Structural Matrices (p. 12))	FX ... MZ
$\{F^{jb}\}$	Lorentz force vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	FX ... FZ
$\{F^{mx}\}$	Maxwell force vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	FX ... FZ

Symbol	Meaning	Associated Input / Output Label
$\{F^e\}$	electrostatic body force load vector (discussed in Electroelasticity (p. 317))	FX ...FZ
$\{F^b\}$	body force load vector due to non-gravity effects (discussed in Derivation of Heat Flow Matrices (p. 235))	FX ... MZ
$\{F^{di}\}$	diffusion strain force vector (discussed in Structural-Diffusion Coupling (p. 323))	FX ... MZ
$\{Q^{nd}\}$	applied nodal heat flow rate vector (discussed in Derivation of Heat Flow Matrices (p. 235))	HEAT, HBOT, HE2, ... HTOP
$\{Q^f\}$	heat flux vector (discussed in Derivation of Heat Flow Matrices (p. 235))	HEAT, HBOT, HE2, ... HTOP
$\{Q^c\}$	convection surface vector (discussed in Derivation of Heat Flow Matrices (p. 235))	HEAT, HBOT, HE2, ... HTOP
$\{Q^g\}$	heat generation rate vector for causes other than Joule heating (discussed in Derivation of Heat Flow Matrices (p. 235))	HEAT, HBOT, HE2, ... HTOP
$\{Q^j\}$	heat generation rate vector for Joule heating (discussed in Electromagnetic Field Evaluations (p. 200))	HEAT
$\{Q^p\}$	Peltier heat flux vector (discussed in Thermoelectrics (p. 319))	HEAT
$\{Q^{ted}\}$	heat generation rate vector for thermoelastic damping	HEAT
$\{\psi_i^{nd}\}$	applied nodal source current vector (associated with {A}) (discussed in Derivation of Electromagnetic Matrices (p. 193))	CSGX, CSGY, CSGZ
$\{\psi_f^{nd}\}$	applied nodal flux vector (associated with $\{\varphi\}$) (discussed in Derivation of Electromagnetic Matrices (p. 193))	FLUX
$\{\Psi^g\}$	source (Biot-Savart) vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	FLUX
$\{\Psi^{pm}\}$	coercive force (permanent magnet) vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	FLUX
$\{\Psi^s\}$	source current vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	FLUX
$\{I^{nd}\}$	applied nodal electric current vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	AMPS
$\{V_0\}$	applied voltage drop vector (discussed in Stranded Coil Analysis)	VLTG
$\{L^{nd}\}$	applied nodal charge vector (discussed in Piezoelectrics (p. 313))	AMPS (CHRG for PLANE223, SOLID226, and SOLID227)
$\{L^c\}$	charge density load vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	CHRGD
$\{L^{sc}\}$	surface charge density load vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	CHRGS
$\{L^{th}\}$	thermo-piezoelectric load vector (discussed in Piezoelectrics (p. 313))	TEMP, EPTH

Symbol	Meaning	Associated Input / Output Label
$\{W^{nd}\}$	applied nodal fluid flow vector (discussed in FLUID116 - Coupled Thermal-Fluid Pipe (p. 500))	FLOW
$\{W^{h}\}$	static head vector (discussed in FLUID116 - Coupled Thermal-Fluid Pipe (p. 500))	FLOW
{R}	diffusion flow rate vector	RATE

Vectors of Unknowns

{u}	displacement vector (discussed in Derivation of Structural Matrices (p. 12))	UX ... ROTZ
{T}	thermal potential (temperature) vector (discussed in Derivation of Heat Flow Matrices (p. 235))	TEMP, TBOT, TE2, ... TTOP
{V}	electric potential vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	VOLT
$\{\Delta V\}$	voltage drop in a stranded coil analysis (discussed in Stranded Coil Analysis)	VOLT
$\{\nu\}$	time integrated electric potential vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	VOLT
$\{\phi\}$	magnetic scalar potential vector (discussed in Derivation of Electromagnetic Matrices (p. 193))	MAG
{A}	magnetic vector potential or edge-flux (discussed in Derivation of Electromagnetic Matrices (p. 193))	AX, AY, AZ or AZ
{i}	electric current vector (discussed in Stranded Coil Analyses (p. 210))	CURR
{e}	electromagnetic force drop vector (discussed in Stranded Coil Analyses (p. 210))	EMF
{P}	pressure vector (discussed in and Derivation of Acoustic Matrices (p. 256))	PRES
{C}	concentration vector (discussed in Derivation of Diffusion Matrices (p. 290))	CONC
$\dot{}$	time derivative	
$\ddot{}$	second time derivative	

10.2. Thermoelasticity

The capability to do a thermoelastic analysis exists in the following elements:

PLANE223 - 2-D 8-Node Coupled-Field Solid
SOLID226 - 3-D 20-Node Coupled-Field Solid
SOLID227 - 3-D 10-Node Coupled-Field Solid

These elements support both the thermal expansion and piezocaloric effects, and use the strong (matrix) coupling method.

In addition to the above, the following elements support the thermal expansion effect only in the form of a thermal strain load vector, i.e. use weak coupling method:

SOLID5 - 3-D 8-Node Coupled-Field Solid
PLANE13 - 2-D 4-Node Coupled-Field Solid
SOLID98 - 3-D 10-Node Coupled-Field Solid

Constitutive Equations of Thermoelasticity

The coupled thermoelastic constitutive equations (Nye([359] (p. 941))) are:

$$\{\varepsilon\} = [D]^{-1}\{\sigma\} + \{\alpha\}\Delta T \tag{10.29}$$

$$S = \{\alpha\}^T\{\sigma\} + \frac{\rho C_p}{T_0}\Delta T \tag{10.30}$$

where:

$\{\varepsilon\}$ = total strain vector = $[\varepsilon_x\ \varepsilon_y\ \varepsilon_z\ \varepsilon_{xy}\ \varepsilon_{yz}\ \varepsilon_{xz}]^T$
S = entropy density
$\{\sigma\}$ = stress vector = $[\sigma_x\ \sigma_y\ \sigma_z\ \sigma_{xy}\ \sigma_{yz}\ \sigma_{xz}]^T$
ΔT = temperature change = $T - T_{ref}$
T = current temperature
T_0 = absolute reference temperature = $T_{ref} + T_{off}$
T_{ref} = reference temperature (input on **TREF** command or as REFT on **MP** command)
T_{off} = offset temperature from absolute zero to zero (input on **TOFFST** command)
[D] = elastic stiffness matrix (inverse defined in Equation 2.4 (p. 6) or input using **TB**,ANEL command)
$\{\alpha\}$ = vector of coefficients of thermal expansion = $[\alpha_x\ \alpha_y\ \alpha_z\ 0\ 0\ 0]^T$ (input using, for example, ALPX, ALPY, ALPZ on **MP** command)
ρ = density (input as DENS on **MP** command)
C_p = specific heat at constant stress or pressure (input as C on **MP** command)

Using $\{\varepsilon\}$ and ΔT as independent variables, and replacing the entropy density S in Equation 10.30 (p. 310) by heat density Q using the second law of thermodynamics for a reversible change

$$Q = T_0 S \tag{10.31}$$

we obtain

$$\{\sigma\} = [D]\{\varepsilon\} - \{\beta\}\Delta T \tag{10.32}$$

$$Q = T_0\{\beta\}^T\{\varepsilon\} + \rho C_v \Delta T \tag{10.33}$$

where:

$\{\beta\}$ = vector of thermoelastic coefficients = [D] $\{\alpha\}$

C_v = specific heat at constant strain or volume = $C_p - \frac{T_0}{\rho}\{\alpha\}^T\{\beta\}$

Substituting Q from Equation 10.33 (p. 310) into the heat flow equation Equation 6.1 (p. 227) produces:

$$\frac{\partial Q}{\partial t} = T_0 \{\beta\}^T \frac{\partial \{\varepsilon\}}{\partial t} + \rho C_v \frac{\partial (\Delta T)}{\partial t} - [K]\nabla^2 T \tag{10.34}$$

where:

$$[K] = \begin{bmatrix} K_{xx} & 0 & 0 \\ 0 & K_{yy} & 0 \\ 0 & 0 & K_{zz} \end{bmatrix} = \text{thermal conductivity matrix}$$

K_{xx}, K_{yy}, K_{zz} = thermal conductivities (input as KXX, KYY, KZZ on **MP** command)

Derivation of Thermoelastic Matrices

Applying the variational principle to stress equation of motion and the heat flow conservation equation coupled by the thermoelastic constitutive equations, produces the following finite element matrix equation:

$$\begin{bmatrix} [M] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{T}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [C^{tu}] & [C^t] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{T}\} \end{Bmatrix} + \begin{bmatrix} [K] & [K^{ut}] \\ [0] & [K^t] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{T\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{Q\} \end{Bmatrix} \tag{10.35}$$

where:

[M] = element mass matrix (defined by Equation 2.58 (p. 14))
[C] = element structural damping matrix (discussed in Damping Matrices (p. 673))
[K] = element stiffness matrix (defined by Equation 2.58 (p. 14))
{u} = displacement vector
{F} = sum of the element nodal force (defined by Equation 2.56 (p. 14)) and element pressure (defined by Equation 2.58 (p. 14)) vectors
$[C^t]$ = element specific heat matrix (defined by Equation 6.22 (p. 235))
$[K^t]$ = element thermal conductivity matrix (defined by Equation 6.22 (p. 235))
{T} = temperature vector
{Q} = sum of the element heat generation load and element convection surface heat flow vectors (defined by Equation 6.22 (p. 235))

$$[K^{ut}] = \text{element thermoelastic stiffness matrix} = -\int_{vol} [B]^T \{\beta\}(\{N\}^T)\, d(vol)$$

[B] = strain-displacement matrix (see Equation 2.44 (p. 12))
{N} = element shape functions
$[C^{tu}]$ = element thermoelastic damping matrix = $-T_0[K^{ut}]^T$

Energy Calculation

In static and transient thermoelastic analyses, the element instantaneous total strain energy is calculated as:

$$U_t = \frac{1}{2} \int_{vol} \{\sigma\}^T \{\varepsilon\}\, d(vol) \tag{10.36}$$

where:

U_t = total strain energy (output as an NMISC element item UT).

Note that Equation 10.36 (p. 311) uses the total strain, whereas the standard strain energy (output as SENE) uses the elastic strain.

In a harmonic thermoelastic analysis, the time-averaged element total strain energy is given by:

$$U_t = \frac{1}{4}\int_{vol} \{\sigma\}^T \{\varepsilon\} * d(vol) \tag{10.37}$$

where:

$\{\varepsilon\}^*$ = complex conjugate of the total strain

The real part of Equation 10.37 (p. 312) represents the average stored strain energy, while its imaginary part - the average energy loss due to thermoelastic damping.

The thermoelastic damping can be quantified by the quality factor Q derived from the total strain energy Equation 10.37 (p. 312) using the real and imaginary solution sets:

$$Q^{-1} = \frac{\sum_{j=1}^{N_e} Im(U_t)}{\sum_{j=1}^{N_e} Re(U_t)} \tag{10.38}$$

where:

N_e = number of thermoelastic elements

10.3. Thermoplasticity

The capability to do a thermoplastic analysis exists in the following elements:

PLANE223 - 2-D 8-Node Coupled-Field Solid
SOLID226 - 3-D 20-Node Coupled-Field Solid
SOLID227 - 3-D 10-Node Coupled-Field Solid

These elements support the thermoplastic effect which manifests itself as an increase in temperature during plastic deformation due to the conversion of some of the plastic work into heat.

In a thermoplastic analysis, the stress equation of motion (Equation 2.51 (p. 13)) and heat flow conservation equation (Equation 6.1 (p. 227)) are coupled by the plastic heat density rate $\dot{Q}^p$ defined as:

$$\dot{Q}^p = \beta \dot{W}^p \tag{10.39}$$

where:

β = fraction of plastic work

$\dot{W}^p$
converted to heat; also referred to as Taylor-Quinney coefficient (input as QRATE on **MP** command)

$\dot{W}^p$ = plastic work rate = $\{\sigma\}^T \{\dot{\varepsilon}^p\}$

where:

$\{\sigma\}$ = stress vector = $\{\sigma_x \sigma_y \sigma_z \sigma_{xy} \sigma_{yz} \sigma_{xz}\}^T$

$\{\dot{\varepsilon}^p\}$ = plastic strain vector = $\{\varepsilon_x^p \varepsilon_y^p \varepsilon_z^p \varepsilon_{xy}^p \varepsilon_{yz}^p \varepsilon_{xz}^p\}^T$

The coupled-field finite element matrix equation for the thermoplastic analysis is:

$$\begin{bmatrix} [M] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{T}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [0] & [C^t] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{T}\} \end{Bmatrix} + \begin{bmatrix} [K] & [0] \\ [0] & [K^t] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{T\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{Q\} + \{Q^p\} \end{Bmatrix}$$

where:

[M] = element mass matrix (defined by Equation 2.58 (p. 14))
[C] = element structural damping matrix (discussed in Damping Matrices (p. 673))
[K] = element stiffness matrix (defined by Equation 2.58 (p. 14))
{u} = displacement vector
{F} = sum of the element nodal force (defined by Equation 2.56 (p. 14)) and element pressure (defined by Equation 2.58 (p. 14)) vectors
$[C^t]$ = element specific heat matrix (defined by Equation 6.22 (p. 235))
$[K^t]$ = element diffusion conductivity matrix (defined by Equation 6.22 (p. 235))
{T} = temperature vector
{Q} = sum of the element heat generation rate load and element convection surface heat flow vectors (defined by Equation 6.22 (p. 235))

$\{Q^p\}$ = element plastic heat generation rate load = $\int_{vol} \dot{Q}_n^p \{N\} d(vol)$

where:

$\dot{Q}_n^p$ = element plastic heat density rate at substep n (output as NMISC,5)
{N} = element shape functions

10.4. Piezoelectrics

The capability of modeling piezoelectric response exists in the following elements:

SOLID5 - 3-D 8-Node Coupled-Field Solid
PLANE13 - 2-D 4-Node Coupled-Field Solid
SOLID98 - 3-D 10-Node Coupled-Field Solid
PLANE223 - 2-D 8-Node Coupled-Field Solid
SOLID226 - 3-D 20-Node Coupled-Field Solid
SOLID227 - 3-D 10-Node Coupled-Field Solid

Constitutive Equations of Piezoelectricity

In linear piezoelectricity the equations of elasticity are coupled to the charge equation of electrostatics by means of piezoelectric constants (IEEE Standard on Piezoelectricity([89] (p. 925))):

$$\{T\} = [c^E]\{S\} - [e]\{E\} \tag{10.40}$$

$$\{D\} = [e]^T \{S\} + [\varepsilon^S]\{E\} \tag{10.41}$$

or equivalently

$$\begin{Bmatrix}\{T\}\\ \{D\}\end{Bmatrix}=\begin{bmatrix}[c^E] & [e]\\ [e]^T & -[\varepsilon^S]\end{bmatrix}\begin{Bmatrix}\{S\}\\ -\{E\}\end{Bmatrix} \tag{10.42}$$

where:

{T} = stress vector (referred to as {σ} elsewhere in this manual)
{D} = electric flux density vector
{S} = elastic strain vector (referred to as $\{\varepsilon^{el}\}$ elsewhere in this manual)
{E} = electric field intensity vector
$[c^E]$ = elasticity matrix (evaluated at constant electric field (referred to as [D] elsewhere in this manual))
[e] = piezoelectric stress matrix
$[\varepsilon^S]$ = dielectric matrix (evaluated at constant mechanical strain)

Equation 10.40 (p. 313) and Equation 10.41 (p. 313) are the usual constitutive equations for structural and electrical fields, respectively, except for the coupling terms involving the piezoelectric matrix [e].

The elasticity matrix [c] is the usual [D] matrix described in Structural Fundamentals (p. 5) (input using the **MP** commands). It can also be input directly in uninverted form [c] or in inverted form $[c]^{-1}$ as a general anisotropic symmetric matrix (input using **TB**,ANEL):

$$[c]=\begin{bmatrix}c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16}\\ & c_{22} & c_{23} & c_{24} & c_{25} & c_{26}\\ & & c_{33} & c_{34} & c_{35} & c_{36}\\ & \text{Symmetric} & & c_{44} & c_{45} & c_{46}\\ & & & & c_{55} & c_{56}\\ & & & & & c_{66}\end{bmatrix} \tag{10.43}$$

The piezoelectric stress matrix [e] (input using **TB**,PIEZ with *TBOPT* = 0) relates the electric field vector {E} in the order X, Y, Z to the stress vector {T} in the order X, Y, Z, XY, YZ, XZ and is of the form:

$$[e]=\begin{bmatrix}e_{11} & e_{12} & e_{13}\\ e_{21} & e_{22} & e_{23}\\ e_{31} & e_{32} & e_{33}\\ e_{41} & e_{42} & e_{43}\\ e_{51} & e_{52} & e_{53}\\ e_{61} & e_{62} & e_{63}\end{bmatrix} \tag{10.44}$$

The piezoelectric matrix can also be input as a piezoelectric strain matrix [d] (input using **TB**,PIEZ with *TBOPT* = 1). ANSYS will automatically convert the piezoelectric strain matrix [d] to a piezoelectric stress matrix [e] using the elasticity matrix [c] at the first defined temperature:

$$[e]=[c][d] \tag{10.45}$$

The orthotropic dielectric matrix $[\varepsilon^S]$ uses the electrical permittivities (input as PERX, PERY and PERZ on the **MP** commands) and is of the form:

$$[\varepsilon^S] = \begin{bmatrix} \varepsilon_{11} & 0 & 0 \\ 0 & \varepsilon_{22} & 0 \\ 0 & 0 & \varepsilon_{33} \end{bmatrix} \tag{10.46}$$

The anisotropic dielectric matrix at constant strain $[\varepsilon^S]$ (input using **TB**,DPER,,,,0 command) is used by PLANE223, SOLID226, and SOLID227 and is of the form:

$$[\varepsilon^S] = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ & \varepsilon_{22} & \varepsilon_{23} \\ \text{Symm} & & \varepsilon_{33} \end{bmatrix} \tag{10.47}$$

The dielectric matrix can also be input as a dielectric permittivity matrix at constant stress $[\varepsilon^T]$ (input using **TB**,DPER,,,,1). The program will automatically convert the dielectric matrix at constant stress to a dielectric matrix at constant strain:

$$[\varepsilon^S] = [\varepsilon^T] - [e]^T [d] \tag{10.48}$$

where:

$[\varepsilon^S]$ = dielectric permittivity matrix at constant strain
$[\varepsilon^T]$ = dielectric permittivity matrix at constant stress
[e] = piezoelectric stress matrix
[d] = piezoelectric strain matrix

Derivation of Piezoelectric Matrices

After the application of the variational principle and finite element discretization (Allik([81] (p. 925))), the coupled finite element matrix equation derived for a one element model is:

$$\begin{bmatrix} [M] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{V}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [0] & -[C^{vh}] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} [K] & [K^z] \\ [K^z]^T & -[K^d] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{L\} + \{L^{th}\} \end{Bmatrix} \tag{10.49}$$

where:

[K] = element stiffness matrix (defined by Equation 2.58 (p. 14))
[M] = element mass matrix (defined by Equation 2.58 (p. 14))
[C] = element structural damping matrix (discussed in Damping Matrices (p. 673))
{F} = vector of nodal and surface forces (defined by Equation 2.56 (p. 14) and Equation 2.58 (p. 14))
$[K^d]$ = element dielectric permittivity coefficient matrix ($[K^{vs}]$ in Equation 5.121 (p. 200) or $[K^{vh}]$ in Equation 5.120 (p. 199))
{L} = vector of nodal, surface, and body charges (defined by Equation 5.121 (p. 200))

$$[K^z] = \int_{vol} [B]^T [e][B] d(vol) = \text{piezoelectric coupling matrix}$$

[B] = strain-displacement matrix (see Equation 2.44 (p. 12))
$[C^{vh}]$ = element dielectric damping matrix (defined by Equation 5.120 (p. 199))

$$\{L^{th}\} = \int_{vol} (\nabla \{N\}^T)^T [e]\{\varepsilon^{th}\} d(vol) = \text{element thermo-piezoelectric load vector}$$

$\{\varepsilon^{th}\}$ = thermal strain vector (as defined by equation Equation 2.3 (p. 6))

{N} = element shape functions

Note

In a strongly coupled thermo-piezoelectric analysis (see Equation 10.16 (p. 302)), the electric potential and temperature degrees of freedom are coupled by:

$$[K^{zt}] = -\int_{vol} (\nabla\{N\}^T)^T [e]\{\alpha\}(\{N\}^T) d(vol)$$

where:

{α} = vector of coefficient of thermal expansion.

Energy Calculation

In static and transient piezoelectric analyses, the PLANE223, SOLID226, and SOLID227 element instantaneous elastic energy is calculated as:

$$U_E = \frac{1}{2}\int_{vol} \{T\}^T \{S\} d(vol) \quad (10.50)$$

where:

U_E = elastic strain energy (output as an NMISC element item UE).

and the electrostatic energy is calculated as:

$$U_D = \frac{1}{2}\int_{vol} \{E\}^T \{D\} d(vol) \quad (10.51)$$

where:

U_D = dielectric energy (output as an NMISC element item UD)

In a harmonic piezoelectric analysis, the time-averaged element energies are calculated as:

$$U_E = \frac{1}{4}\int_{vol} \{T\}^T \{S\}^* d(vol) \quad (10.52)$$

$$U_D = \frac{1}{4}\int_{vol} \{E\}^T \{D\}^* d(vol) \quad (10.53)$$

where:

{S}* = complex conjugate of the elastic strain
{D}* = complex conjugate of the electric flux density

The real parts of equations (Equation 10.52 (p. 316)) and (Equation 10.53 (p. 316)) represent the average stored elastic and dielectric energies, respectively. The imaginary parts represent the average elastic and electric losses. Therefore, the quality factor Q can be calculated from the total stored energy as:

$$Q^{-1} = \frac{\sum_{j=1}^{N_e} \mathrm{Im}(U_E + U_d)}{\sum_{j=1}^{N_e} \mathrm{Re}(U_E + U_d)} \tag{10.54}$$

where:

N_e = number of piezoelectric elements

The total stored energy $U_E + U_D$ is output as SENE. Therefore, the Q factor can be derived from the real and imaginary records of SENE summed over the piezoelectric elements.

10.5. Electroelasticity

The capability of modeling electrostatic force coupling in elastic dielectrics exists in the following elements:

PLANE223 - 2-D 8-Node Coupled-Field Solid
SOLID226 - 3-D 20-Node Coupled-Field Solid
SOLID227 - 3-D 10-Node Coupled-Field Solid

Elastic dielectrics exhibit a deformation when subject to an electrostatic field. The electrostatic body force that causes the deformation can be derived from the Maxwell stress tensor $[\sigma^M]$ (Landau and Lifshitz([358] (p. 941))).

$$[\sigma^M] = \begin{bmatrix} \sigma_x^M & \sigma_{xy}^M & \sigma_{xz}^M \\ & \sigma_y^M & \sigma_{yz}^M \\ \text{symm} & & \sigma_z^M \end{bmatrix} = \frac{1}{2}(\{E\}\{D\}^T + \{D\}\{E\}^T - \{D\}^T\{E\}[I]) \tag{10.55}$$

where:

{E} = electric field intensity vector
{D} = electric flux density vector

$$[I] = \text{identity matrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Applying the variational principle to the stress equation of motion with the electrostatic body force loading and to the charge equation of electrostatics, produces the following finite element equation for electroelasticity:

$$\begin{bmatrix} [M] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{V}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} [K] & [0] \\ [0] & [K^d] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{F\} + \{F^e\} \\ \{L\} \end{Bmatrix} \tag{10.56}$$

where:

[K] = element structural stiffness matrix (see $[K_e]$ in Equation 2.58 (p. 14))
[M] = element mass matrix (see $[M_e]$ in Equation 2.58 (p. 14))

[C] = element structural damping matrix (discussed in Damping Matrices (p. 673))
{F} = vector of nodal and surface forces (defined by Equation 2.56 (p. 14) and Equation 2.58 (p. 14))

$\{F^e\}$ = vector of nodal electrostatic forces = $-\int_{vol} [B]^T \{\sigma^M\} d(vol)$

[B] = strain-displacement matrix (see Equation 2.44 (p. 12))

$\{\sigma^M\}$ = Maxwell stress vector = $\{\sigma_x^M \ \sigma_y^M \ \sigma_z^M \ \sigma_{xy}^M \ \sigma_{yz}^M \ \sigma_{xz}^M\}^T$

$[K^d]$ = element dielectric permittivity coefficient matrix (see $[K^{vs}]$ in Equation 5.121 (p. 200))
{L} = vector of nodal, surface, and body charges (see $\{L_e\}$ in Equation 5.121 (p. 200))

10.6. Piezoresistivity

The capability to model piezoresistive effect exists in the following elements:

PLANE223 - 2-D 8-Node Coupled-Field Solid
SOLID226 - 3-D 20-Node Coupled-Field Solid
SOLID227 - 3-D 10-Node Coupled-Field Solid

In piezoresistive materials, stress or strain cause a change of electric resistivity:

$$[\rho] = [\rho^o]([I] + [r]) \tag{10.57}$$

where:

$[\rho]$ = electric resistivity matrix of a loaded material = $\begin{bmatrix} \rho_{xx} & \rho_{xy} & \rho_{xz} \\ & \rho_{yy} & \rho_{yz} \\ \text{symm} & & \rho_{zz} \end{bmatrix}$

$[\rho^o]$ = electric resistivity matrix of an unloaded material = $\begin{bmatrix} \rho_{xx}^o & 0 & 0 \\ 0 & \rho_{yy}^o & 0 \\ 0 & 0 & \rho_{zz}^o \end{bmatrix}$

$\rho_{xx}^o, \rho_{yy}^o, \rho_{zz}^o$ = electrical resistivities (input as RSVX, RSVY, RSVZ on **MP** command)

$[I]$ = identity matrix = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$[r]$ = relative change in resistivity = $\begin{bmatrix} r_x & r_{xy} & r_{xz} \\ & r_y & r_{yz} \\ \text{symm} & & r_z \end{bmatrix}$ calculated as:

$$\{r\} = [\pi]\{\sigma\} \tag{10.58}$$

where:

{r} = vector of matrix [r] components = $[r_x \ r_y \ r_z \ r_{xy} \ r_{yz} \ r_{xz}]^T$

$$[\pi] = \text{piezoresistive stress matrix} = \begin{bmatrix} \pi_{11} & \pi_{12} & \pi_{13} & \pi_{14} & \pi_{15} & \pi_{16} \\ \pi_{21} & \pi_{22} & \pi_{23} & \pi_{24} & \pi_{25} & \pi_{26} \\ \pi_{31} & \pi_{32} & \pi_{33} & \pi_{34} & \pi_{35} & \pi_{36} \\ \pi_{41} & \pi_{42} & \pi_{43} & \pi_{44} & \pi_{45} & \pi_{46} \\ \pi_{51} & \pi_{52} & \pi_{53} & \pi_{54} & \pi_{55} & \pi_{56} \\ \pi_{61} & \pi_{62} & \pi_{63} & \pi_{64} & \pi_{65} & \pi_{66} \end{bmatrix}$$

(input on **TB**,PZRS command with *TBOPT* = 0)

$$\{\sigma\} = \text{stress vector} = [\sigma_x \ \sigma_y \ \sigma_z \ \sigma_{xy} \ \sigma_{yz} \ \sigma_{xz}]^T$$

Similarly, for strains:

$$\{r\} = [m]\{\varepsilon^{el}\} \tag{10.59}$$

where:

[m] = piezoresistive strain matrix (input on **TB**,PZRS command with *TB0PT* = 1)

$\{\varepsilon^{el}\}$ = elastic strain vector

The coupled-field finite element matrix equation for the piezoresistive analysis is given by:

$$\begin{bmatrix} [M] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{V}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [0] & [C^v] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} [K] & [0] \\ [0] & [K^v] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{I\} \end{Bmatrix} \tag{10.60}$$

The terms used in the above equation are explained in Piezoresistive Analysis (p. 302) where the conductivity matrix $[K^v]$ is derived as:

$$[K^v] = \int_{vol} (\nabla\{N\}^T)^T [\rho]^{-1} (\nabla\{N\}^T) d(vol) \tag{10.61}$$

10.7. Thermoelectrics

The capability to model thermoelectric effects exists in the following elements:

PLANE223 - 2-D 8-Node Coupled-Field Solid
SOLID226 - 3-D 20-Node Coupled-Field Solid
SOLID227 - 3-D 10-Node Coupled-Field Solid

These elements support the Joule heating effect (irreversible), and the Seebeck, Peltier, and Thomson effects (reversible).

In addition to the above, the following elements support a basic thermoelectric analysis that takes into consideration Joule heating effect only:

SOLID5 - 3-D 8-Node Coupled-Field Solid
LINK68 - 3-D 2-Node Coupled Thermal-Electric Line
SOLID98 - 3-D 10-Node Coupled-Field Solid
SHELL157 - 3-D 4-Node Thermal-Electric Shell

Constitutive Equations of Thermoelectricity

The coupled thermoelectric constitutive equations (Landau and Lifshitz([358] (p. 941))) are:

$$\{q\} = [\Pi]\{J\} - [K]\{\nabla T\} \tag{10.62}$$

$$\{J\} = [\sigma](\{E\} - [\alpha]\{\nabla T\}) \tag{10.63}$$

Substituting $[\Pi]$ with $T[\alpha]$ to further demonstrate the coupling between the above two equations,

$$\{q\} = T[\alpha]\{J\} - [K]\{\nabla T\} \tag{10.64}$$

$$\{J\} = [\sigma](\{E\} - [\alpha]\{\nabla T\}) \tag{10.65}$$

where:

$[\Pi]$ = Peltier coefficient matrix = $T[\alpha]$
T = absolute temperature

$$[\alpha] = \begin{bmatrix} \alpha_{xx} & 0 & 0 \\ 0 & \alpha_{yy} & 0 \\ 0 & 0 & \alpha_{zz} \end{bmatrix} = \text{Seebeck coefficient matrix}$$

{q} = heat flux vector (output as TF)
{J} = electric current density (output as JC for elements that support conduction current calculation)

$$[K] = \begin{bmatrix} K_{xx} & 0 & 0 \\ 0 & K_{yy} & 0 \\ 0 & 0 & K_{zz} \end{bmatrix} = \text{thermal conductivity matrix evaluated at zero electric current } (\{J\} = \{0\})$$

$\{\nabla T\}$ = thermal gradient (output as TG)

$$[\sigma] = \begin{bmatrix} \frac{1}{\rho_{xx}} & 0 & 0 \\ 0 & \frac{1}{\rho_{yy}} & 0 \\ 0 & 0 & \frac{1}{\rho_{zz}} \end{bmatrix} = \text{electrical conductivity matrix evaluated at zero temperature gradient } (\{\nabla T\} = \{0\})$$

{E} = electric field (output as EF)
α_{xx}, α_{yy}, α_{zz} = Seebeck coefficients (input as SBKX, SBKY, SBKZ on **MP** command)
K_{xx}, K_{yy}, K_{zz} = thermal conductivities (input as KXX, KYY, KZZ on **MP** command)
ρ_{xx}, ρ_{yy}, ρ_{zz} = resistivity coefficients (input as RSVX, RSVY, RSVZ on **MP** command)

Note that the Thomson effect is associated with the temperature dependencies of the Seebeck coefficients (**MPDATA**,SBKX also SBKY, SBKZ).

Derivation of Thermoelectric Matrices

After the application of the variational principle to the equations of heat flow (Equation 6.1 (p. 227)) and of continuity of electric charge (Equation 5.5 (p. 178)) coupled by Equation 10.62 (p. 320) and Equation 10.63 (p. 320), the finite element equation of thermoelectricity becomes (Antonova and Looman([90] (p. 925))):

$$\begin{bmatrix} [C^t] & [0] \\ [0] & [C^v] \end{bmatrix} \begin{Bmatrix} \{\dot{T}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} [K^t] & [0] \\ [K^{vt}] & [K^v] \end{bmatrix} \begin{Bmatrix} \{T\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{Q\} + \{Q^p\} \\ \{I\} \end{Bmatrix} \tag{10.66}$$

where:

$[K^t]$ = element thermal conductivity matrix (defined by Equation 6.22 (p. 235))

$[C^t]$ = element specific heat matrix (defined by Equation 6.22 (p. 235))

$\{Q\}$ = sum of the element heat generation load and element convection surface heat flow vectors (defined by Equation 6.22 (p. 235))

$[K^v]$ = element electrical conductivity coefficient matrix (defined by Equation 5.119 (p. 199))

$[C^v]$ = element dielectric permittivity coefficient matrix (defined by Equation 5.119 (p. 199))

$[K^{vt}]$ = element Seebeck coefficient coupling matrix

$$= \int_{vol} (\nabla\{N\}^T)^T [\sigma][\alpha](\nabla\{N\}^T)\, d(vol)$$

$\{Q^p\}$ = element Peltier heat load vector

$$= \int_{vol} (\nabla\{N\}^T)^T [\Pi]\{J\}\, d(vol)$$

$\{N\}$ = element shape functions

$\{I\}$ = vector of nodal current load

10.8. Review of Coupled Electromechanical Methods

The sequential coupling between electrical and mechanical finite element physics domains for coupled Electromechanical analysis can be performed by the ANSYS Multi-field solver. The ANSYS Multi-field solver allows the most general treatment of individual physics domains. However, it cannot be applied to small signal modal and harmonic analyses because a total system eigen frequency analysis requires matrix coupling. Moreover, sequential coupling generally converges slower.

Strong Electromechanical coupling can be performed by the transducer element TRANS126, Gyimesi and Ostergaard([248] (p. 934)), Gyimesi and Ostergaard([330] (p. 939)), TRANS126 - Electromechanical Transducer (p. 513).

TRANS126 completely models the fully coupled system, converting electrostatic energy into mechanical energy and vise versa as well as storing electrostatic energy. Coupling between electrostatic forces and mechanical forces is obtained from virtual work principles (Gyimesi and Ostergaard([248] (p. 934))).

TRANS126 takes on the form of a 2-node line element with electrical voltage and mechanical displacement DOFs as across variables and electric current and mechanical force as through variables. Input for the element consists of a capacitance-stroke relationship that can be derived from electrostatic field solutions and using the **CMATRIX** command macro (Gyimesi et al.([288] (p. 937)), Gyimesi and Ostergaard([289] (p. 937))).

The element can characterize up to three independent translation degrees of freedom at any point to simulate 3-D coupling. Thus, the electrostatic mesh is removed from the problem domain and replaced by a set of TRANS126 elements hooked to the mechanical and electrical model providing a reduced order modeling of a coupled electromechanical system (Gyimesi and Ostergaard ([286] (p. 937)), Gyimesi et al.([287] (p. 937))).

TRANS126 allows treatment of all kinds of analysis types, including prestressed modal and harmonic analyses. However, TRANS126 is limited geometrically to problems when the capacitance can be accurately described as a function of a single degree of freedom, usually the stroke of a comb drive. In a bending electrode problem, like an optical switch, obviously, a single TRANS126 element can not be applied. When the gap is small and fringing is not significant, the capacitance between deforming electrodes can be practically modeled reasonably well by several capacitors connected parallel. The **EMTGEN** (electromechanical transducer generator) command macro can be applied to this case.

Convergence issues may be experienced with TRANS126 when applied to the difficult hysteric pull-in and release analysis (Avdeev et al.([331] (p. 939))) because of the negative total system stiffness matrix. The issue is resolved when the augmented stiffness method is applied.

10.9. Porous Media Flow

The coupled pore-pressure thermal elements used in analyses involving porous media are listed in Coupled Pore-Pressure Element Support.

The program models porous media containing fluid by treating the porous media as a multiphase material and applying an extended version of Biot's consolidation theory. The flow is considered to be a single-phase fluid. The porous media is assumed to be fully saturated.

Following are the governing equations for Biot consolidation problems:

$$\begin{cases} \nabla \bullet \sigma + f = 0 \quad \text{or} \quad \nabla \bullet (\sigma' - \alpha p I) + f = 0 \\ \alpha \dot{\varepsilon}_V + \dfrac{1}{K_m} \dot{p} + \nabla \bullet q = s \end{cases} \tag{10.67}$$

where

σ = Total Cauchy stress

$\bullet$ = Rate change operator (if used above a term)

$\nabla \bullet$ = Divergence operator of a vector or second order tensor

σ'' = Biot effective stress tensor

α = Biot coefficient

p = Pore pressure (positive sign for hydrostatic pressure)

I = Second-order identity tensor

f = Body force of the porous media

ε_V = Volumetric strain of the solid skeleton

K_m = Biot modulus

q = Flow flux vector

s = Flow source

The relationship between the Biot effective stress and the elastic strain of solid skeletons is given by:

$$\sigma'' = D : \varepsilon^e$$

where ε^e is the second-order elastic strain tensor and *D* is the fourth order elasticity tensor.

The relationship between the fluid flow flux and the pore pressure is described by Darcy's Law:

$$q = -k\nabla p$$

where *k* is the second-order permeability tensor and ∇ is the gradient operator.

Governing Equation Considerations

Equation 10.67 (p. 322) also addresses the use of hyperelastic materials, allowing for an initial, efficient analysis of porous materials with hyperelasticity models. In this case, however, the program assumes that all Biot and permeability parameters remain constant during deformation.

Material and Model Parameter Input

The following commands are available for inputting material and model parameters:

- **TB**,PM - - Input porous media material parameters.
- **SF**,,FFLX or **SFE**,,,FFLX -- Input flow flux boundary conditions.
- **BFE**,,FSOU -- Input flow source.

  Positive values: Fluids flow out of the source points or elements. Negative values: Fluids flow into the source points or elements.

Effective Stress Output

The following commands are available to output effective stresses: **PLNSOL**,ESIG, **PLESOL**,ESIG, **PRNSOL**,ESIG, and **PRESOL**,ESIG.

Total Stress Output

The following commands are available to output total stresses: **PLNSOL**,S, **PLESOL**,S, **PRNSOL**,S, and **PRESOL**,S.

Additional Information

For related information, see the following documentation:

Pore-Fluid-Diffusion-Structural Analysis in the *Coupled-Field Analysis Guide*
Coupled Pore-Fluid Diffusion and Structural Model of Porous Media in the *Material Reference*
TB,PM command

10.10. Structural-Diffusion Coupling

The capability to do a diffusion strain analysis exists in the following elements:

PLANE223 - 2-D 8-Node Coupled-Field Solid
SOLID226 - 3-D 20-Node Coupled-Field Solid
SOLID226 - 3-D 20-Node Coupled-Field Solid

Constitutive Equations

In a coupled structural-diffusion analysis, the total strain is composed of elastic $\{\varepsilon^{el}\}$ and diffusion $\{\varepsilon^{d}\}$ parts, respectively:

$$\{\mu\} = \{\mu^{el}\} + \{\mu^{di}\} = [D]^{-1}\{\tilde{A}\} + C_{sat}\{^2\}\,\Box C \tag{10.68}$$

{ε} = total strain vector = $[\varepsilon_x\ \varepsilon_y\ \varepsilon_z\ \varepsilon_{xy}\ \varepsilon_{yz}\ \varepsilon_{xz}]^T = \{\varepsilon^{el}\} + \{\varepsilon^{d}\}$

$\{\varepsilon^{el}\}$ = elastic strain vector (output as EPEL)

$\{\varepsilon^{di}\}$= diffusion strain vector (output as EPDI)

{σ} = stress vector = $[\sigma_x\ \sigma_y\ \sigma_z\ \sigma_{xy}\ \sigma_{yz}\ \sigma_{xz}]^T$ (output as SIG)

ΔC = concentration change = $C - C_{ref}$

C = concentration (input/output as CONC); normalized concentration if C_{sat}>1

C_{ref} = reference concentration (input as CREF on **MP** command)

[D] = elastic stiffness matrix (inverse defined in Equation 2.4 (p. 6) or input using **TB**,ANEL command)

{β}= vector of coefficients of diffusion expansion = $[\beta_x\ \beta_y\ \beta_z\ 0\ 0\ 0]^T$ (input using BETX, BETY, BETZ on **MP** command)

C_{sat} = saturated concentration (input as CSAT on **MP** command)

Derivation of Structural-Diffusion Matrices

Applying the variational principle to the structural equation (discussed in Derivation of Structural Matrices (p. 12)) and diffusion equation (Equation 9.6 (p. 290)) coupled by the constitutive equation (Equation 10.68 (p. 324)), produces the following finite element matrix equation for the structural-diffusion analysis:

$$\begin{bmatrix}[M] & [0]\\ [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{C}\}\end{Bmatrix} + \begin{bmatrix}[C] & [0]\\ [0] & [C^d]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{C}\}\end{Bmatrix} + \begin{bmatrix}[K] & [K^{ud}]\\ [0] & [K^d]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{C\}\end{Bmatrix} = \begin{Bmatrix}\{F\}\\ \{R\}\end{Bmatrix} \tag{10.69}$$

where:

[M] = element mass matrix (defined by Equation 2.58 (p. 14))

[C] = element structural damping matrix (discussed in Damping Matrices (p. 673))

[K] = element stiffness matrix (defined by Equation 2.58 (p. 14))

{u} = displacement vector

{F} = sum of the element nodal force (defined by Equation 2.56 (p. 14)) and element pressure (defined by Equation 2.58 (p. 14)) vectors

$[C^d]$ = element diffusion damping matrix (defined by Equation 9.8 (p. 291))

$[K^d]$ = element diffusion conductivity matrix (defined by Equation 9.8 (p. 291))

{C} = concentration vector; normalized if $C_{sat} > 1$

{R} = nodal diffusion flow rate vector (defined by Equation 9.8 (p. 291))

$\left[K^{ud}\right]$ = element diffusion-elastic stiffness matrix = $-\int_{vol}[B]^T[D]C_{sat}\{^2\}\left(\{N\}^T\right)d(vol)$

[B] = strain-displacement matrix (see Equation 2.44 (p. 12))

{N} = element shape functions

When the load vector coupling (KEYOPT(2)=1 with PLANE223, SOLID226, SOLID227) is used, the finite element matrix equation becomes symmetric:

$$\begin{bmatrix}[M] & [0]\\ [0] & [0]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}\}\\ \{\ddot{C}\}\end{Bmatrix}+\begin{bmatrix}[C] & [0]\\ [0] & [C^d]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{C}\}\end{Bmatrix}+\begin{bmatrix}[K] & [0]\\ [0] & [K^d]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{C\}\end{Bmatrix}=\begin{Bmatrix}\{F\}+\{F^{di}\}\\ \{R\}\end{Bmatrix} \tag{10.70}$$

where:

$\{F^{di}\}$ = element diffusion strain load vector = $-\int_{vol}[B]^T[D]\{\mu^{di}\}d(vol)$

Chapter 11: Shape Functions

This chapter provides the shape functions for elements. The shape functions are referred to by the individual element descriptions in Element Library (p. 411). All subheadings for this chapter are included in the table of contents to aid in finding a specific type of shape function.

The following shape function topics are available:

11.1. Understanding Shape Function Labels

The given functions are related to the nodal quantities by:

Table 11.1: Shape Function Labels

Variable	Input/Output Label	Meaning
u	UX	Translation in the x (or s) direction
v	UY	Translation in the y (or t) direction
w	UZ	Translation in the x (or r) direction
θ_x	ROTX	Rotation about the x direction
θ_y	ROTY	Rotation about the y direction
θ_z	ROTZ	Rotation about the z direction
A_x	AX	X-component of vector magnetic potential
A_y	AY	Y-component of vector magnetic potential
A_z	AZ	Z-component of vector magnetic potential
C	CONC	Concentration
P	PRES	Pressure
T	TEMP, TBOT, TE2, ... TTOP	Temperature

Variable	Input/Output Label	Meaning
V	VOLT	Electric potential or source current
ϕ	MAG	Scalar magnetic potential

The vector correspondences are not exact, since, for example, u, v, and w are in the element coordinate system, whereas UX, UY, UZ represent motions in the nodal coordinate system. Generally, the element coordinate system is the same as the global Cartesian system, except for:

1. Line elements (2-D Lines (p. 328) to Axisymmetric Harmonic Shells and General Axisymmetric Surfaces (p. 332)), where u motions are axial motions, and v and w are transverse motions.

2. Shell elements (3-D Shells (p. 335)), where u and v are in-plane motions and w is the out-of-plane motion.

Subscripted variables such as u_J refer to the u motion at node J. When these same variables have numbers for subscripts (e.g. u_1), nodeless variables for extra shape functions are being referred to. Coordinates s, t, and r are normalized, going from -1.0 on one side of the element to +1.0 on the other, and are not necessarily orthogonal to one another. L_1, L_2, L_3, and L_4 are also normalized coordinates, going from 0.0 at a vertex to 1.0 at the opposite side or face.

Elements with midside nodes allow those midside nodes to be dropped in most cases. A dropped midside node implies that the edge is and remains straight, and that any other effects vary linearly along that edge.

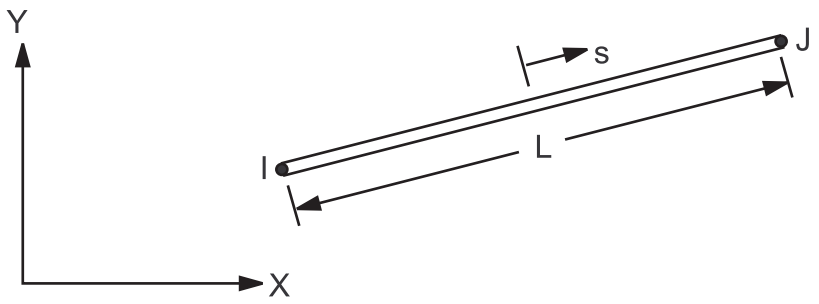
Gaps are left in the equation numbering to allow for additions. Labels given in subsection titles within parentheses are used to relate the given shape functions to their popular names, where applicable.

Some elements in Element Library (p. 411) (notably the 8-node solids) imply that reduced element geometries (e.g., wedge) are not available. However, the tables in Element Library (p. 411) refer only to the available shape functions. In other words, the shape functions used for the 8-node brick is the same as the 6-node wedge.

11.2. 2-D Lines

This section contains shape functions for line elements without and with rotational degrees of freedom (RDOF).

Figure 11.1: 2-D Line Element



11.2.1. 2-D Lines without RDOF

These shape functions are for 2-D line elements without RDOF.

$$u = \frac{1}{2}(u_I(1-s) + u_J(1+s)) \tag{11.1}$$

$$v = \frac{1}{2}(v_I(1-s) + v_J(1+s)) \tag{11.2}$$

$$T = \frac{1}{2}(T_I(1-s) + T_J(1+s)) \tag{11.3}$$

11.2.2. 2-D Lines with RDOF

These shape functions are for 2-D line elements with RDOF.

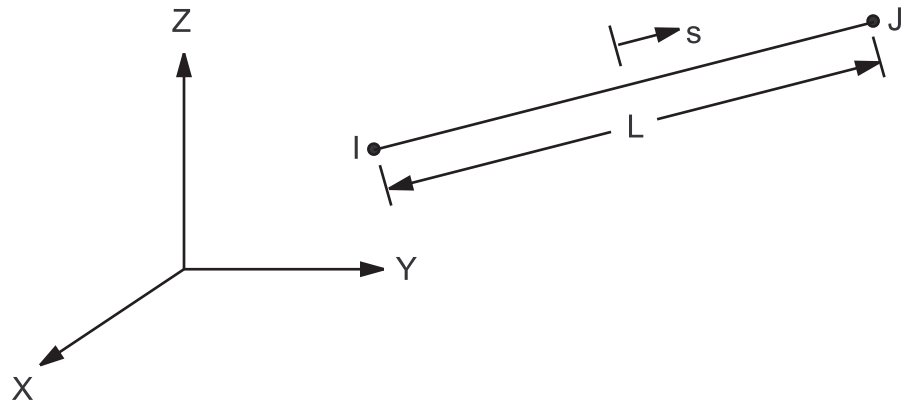
$$u = \frac{1}{2}(u_I(1-s) + u_J(1+s)) \tag{11.4}$$

$$v = \frac{1}{2}\left(v_I\left(1-\frac{s}{2}(3-s^2)\right) + v_J\left(1+\frac{s}{2}(3-s^2)\right)\right) + \frac{L}{8}(\theta_{Z,I}(1-s^2)(1-s) + \theta_{Z,J}(1-s^2)(1+s)) \tag{11.5}$$

11.3. 3-D Lines

This section contains shape functions for various line elements.

Figure 11.2: 3–D Line Element



11.3.1. 3-D 2-Node Lines (Not Combining Translations and Rotations)

These shape functions are for 3-D 2-node line elements without RDOF, such as LINK33, LINK68, or BEAM188.

$$u = \frac{1}{2}(u_I(1-s) + u_J(1+s)) \tag{11.6}$$

$$v = \frac{1}{2}(v_I(1-s) + v_J(1+s)) \tag{11.7}$$

$$w = \frac{1}{2}(w_I(1-s) + w_J(1+s)) \tag{11.8}$$

$$\theta_x = \frac{1}{2}(\theta_{xI}(1-s) + \theta_{xJ}(1+s)) \tag{11.9}$$

$$\theta_y = \frac{1}{2}(\theta_{yI}(1-s) + \theta_{yJ}(1+s)) \tag{11.10}$$

$$\theta_z = \frac{1}{2}(\theta_{zI}(1-s) + \theta_{zJ}(1+s)) \tag{11.11}$$

$$P = \frac{1}{2}(P_I(1-s) + P_J(1+s)) \tag{11.12}$$

$$T = \frac{1}{2}(T_I(1-s) + T_J(1+s)) \tag{11.13}$$

$$V = \frac{1}{2}(V_I(1-s) + V_J(1+s)) \tag{11.14}$$

11.3.2. 3-D 2-Node Lines (Combining Translations and Rotations)

These shape functions are for 3-D 2-node line elements with RDOF.

$$u = \frac{1}{2}(u_I(1-s) + u_J(1+s)) \tag{11.15}$$

$$v = \frac{1}{2}\left(v_I\left(1-\frac{s}{2}(3-s^2)\right) + v_J\left(1+\frac{s}{2}(3-s^2)\right)\right) + \frac{L}{8}(\theta_{z,I}(1-s^2)(1-s) - \theta_{z,J}(1-s^2)(1+s)) \tag{11.16}$$

$$w = \frac{1}{2}\left(w_I\left(1-\frac{s}{2}(3-s^2)\right) + w_J\left(1+\frac{s}{2}(3-s^2)\right)\right) - \frac{L}{8}(\theta_{y,I}(1-s^2)(1-s) - \theta_{y,J}(1-s^2)(1+s)) \tag{11.17}$$

$$\theta_x = \frac{1}{2}(\theta_{x,I}(1-s) + \theta_{x,J}(1+s)) \tag{11.18}$$

11.3.3. 3-D 3-Node Lines

These shape functions are for 3-D 3-node line elements such as BEAM188 and BEAM189.

$$u = \frac{1}{2}(u_I(-s+s^2) + u_J(s+s^2)) + u_K(1-s^2) \tag{11.19}$$

$$v = \frac{1}{2}(v_I(-s+s^2) + v_J(s+s^2)) + v_K(1-s^2) \tag{11.20}$$

$$w = \frac{1}{2}(w_I(-s+s^2) + w_J(s+s^2)) + w_K(1-s^2) \tag{11.21}$$

$$\theta_x = \frac{1}{2}(\theta_{xI}(-s+s^2)+\theta_{xJ}(s+s^2))+\theta_{xK}(1-s^2) \tag{11.22}$$

$$\theta_y = \frac{1}{2}(\theta_{yI}(-s+s^2)+\theta_{yJ}(s+s^2))+\theta_{yK}(1-s^2) \tag{11.23}$$

$$\theta_z = \frac{1}{2}(\theta_{zI}(-s+s^2)+\theta_{zJ}(s+s^2))+\theta_{zK}(1-s^2) \tag{11.24}$$

$$T = \frac{1}{2}(T_I(-s+s^2)+T_J(s+s^2))+T_K(1-s^2) \tag{11.25}$$

11.3.4. 3-D 4-Node Lines

These shape functions are for 3-D 4-node line elements such as BEAM188.

$$u = \frac{1}{16}(u_I(-9s^3+9s^2+s-1)+u_J(9s^3+9s^2-s-1) \\ +u_K(27s^3-9s^2-27s+9)+u_L(-27s^3-9s^2-27s+9)) \tag{11.26}$$

$$v = \frac{1}{16}(v_I(-9s^3+9s^2+s-1)+v_J(9s^3+9s^2-s-1) \\ +v_K(27s^3-9s^2-27s+9)+v_L(-27s^3-9s^2-27s+9)) \tag{11.27}$$

$$w = \frac{1}{16}(w_I(-9s^3+9s^2+s-1)+w_J(9s^3+9s^2-s-1) \\ +w_K(27s^3-9s^2-27s+9)+w_L(-27s^3-9s^2-27s+9)) \tag{11.28}$$

$$\theta_x = \frac{1}{16}(\theta_{x,I}(-9s^3+9s^2+s-1)+\theta_{x,J}(9s^3+9s^2-s-1) \\ +\theta_{x,K}(27s^3-9s^2-27s+9)+\theta_{x,L}(-27s^3-9s^2-27s+9)) \tag{11.29}$$

$$\theta_y = \frac{1}{16}(\theta_{y,I}(-9s^3+9s^2+s-1)+\theta_{y,J}(9s^3+9s^2-s-1) \\ +\theta_{y,K}(27s^3-9s^2-27s+9)+\theta_{y,L}(-27s^3-9s^2-27s+9)) \tag{11.30}$$

$$\theta_z = \frac{1}{16}(\theta_{z,I}(-9s^3+9s^2+s-1)+\theta_{z,J}(9s^3+9s^2-s-1) \\ +\theta_{z,K}(27s^3-9s^2-27s+9)+\theta_{z,L}(-27s^3-9s^2-27s+9)) \tag{11.31}$$

11.4. Axisymmetric Shells

This section contains shape functions for 2-node axisymmetric shell elements under axisymmetric load. These elements may have extra shape functions (ESF).

11.4.1. Axisymmetric Shell without ESF

These shape functions are for 2-node axisymmetric shell elements without extra shape functions, such as SHELL61.

$$u = \frac{1}{2}(u_I(1-s)+u_J(1+s)) \tag{11.32}$$

$$v = \frac{1}{2}(v_I(1-s) + v_J(1+s)) \tag{11.33}$$

$$w = \frac{1}{2}\left(w_I\left(1-\frac{s}{2}(3-s^2)\right) + w_J\left(1+\frac{s}{2}(3-s^2)\right)\right) + \frac{L}{8}(\theta_I(1-s^2)(1-s) - \theta_J(1-s^2)(1+s)) \tag{11.34}$$

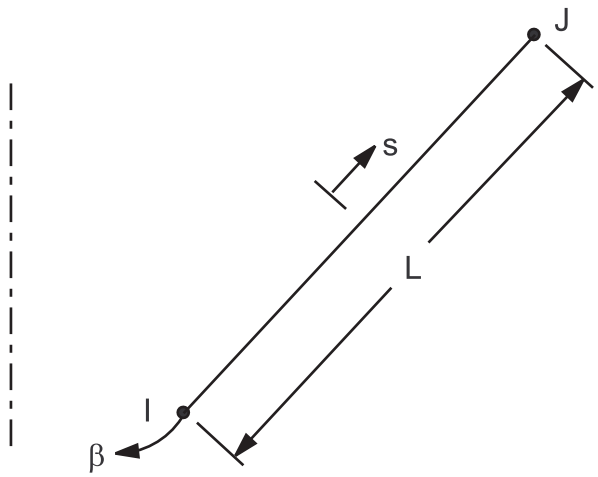
11.5. Axisymmetric Harmonic Shells and General Axisymmetric Surfaces

This section describes shape functions for axisymmetric shell elements under nonaxisymmetric load and for general axisymmetric surfaces.

11.5.1. Axisymmetric Harmonic Shells

This section describes shape functions for 2-node axisymmetric shell elements under nonaxisymmetric (harmonic) load. These elements may or may not have extra shape functions (ESF).

Figure 11.3: Axisymmetric Harmonic Shell Element



The shape functions for axisymmetric harmonic shells use the quantities $\sin \ell\beta$ and $\cos \ell\beta$, where ℓ = input quantity MODE on the **MODE** command. The $\sin \ell\beta$ and $\cos \ell\beta$ are interchanged if I_s = -1, where I_s = input quantity ISYM on the **MODE** command. If ℓ = 0, both $\sin \ell\beta$ and $\cos \ell\beta$ are set equal to 1.0.

11.5.1.1. Axisymmetric Harmonic Shells without ESF

These shape functions are for 2-node axisymmetric harmonic shell elements without extra shape functions, such as SHELL61 with KEYOPT(3) = 1.

$$u = \frac{1}{2}(u_I(1-s) + u_J(1+s))\cos\ell\beta \tag{11.35}$$

$$v = \frac{1}{2}(v_I(1-s)+v_J(1+s))\sin \ell\beta \tag{11.36}$$

$$w = \left(\frac{1}{2}\left(w_I\left(1-\frac{s}{2}(3-s^2)\right)+w_J\left(1+\frac{s}{2}(3-s^2)\right)\right) + \frac{L}{8}(\theta_I(1-s^2)(1-s)-\theta_J(1-s^2)(1+s))\right)\cos \ell\beta \tag{11.37}$$

11.5.1.2. Axisymmetric Harmonic Shells with ESF

These shape functions are for 2-node axisymmetric harmonic shell elements with extra shape functions, such as SHELL61 with KEYOPT(3) = 0.

$$u = \left(\frac{1}{2}\left(u_I\left(1-\frac{s}{2}(3-s^2)\right)+u_J\left(1+\frac{s}{2}(3-s^2)\right)\right) + \frac{L}{8}(u_1(1-s^2)(1-s)-u_2(1-s^2)(1+s))\right)\cos \ell\beta \tag{11.38}$$

$$v = \left(\frac{1}{2}\left(v_I\left(1-\frac{s}{2}(3-s^2)\right)+v_J\left(1+\frac{s}{2}(3-s^2)\right)\right) + \frac{L}{8}(v_1(1-s^2)(1-s)-v_2(1-s^2)(1+s))\right)\sin \ell\beta \tag{11.39}$$

$$w = \left(\frac{1}{2}\left(w_I\left(1-\frac{s}{2}(3-s^2)\right)+w_J\left(1+\frac{s}{2}(3-s^2)\right)\right) + \frac{L}{8}(\theta_I(1-s^2)(1-s)-\theta_J(1-s^2)(1+s))\right)\cos \ell\beta \tag{11.40}$$

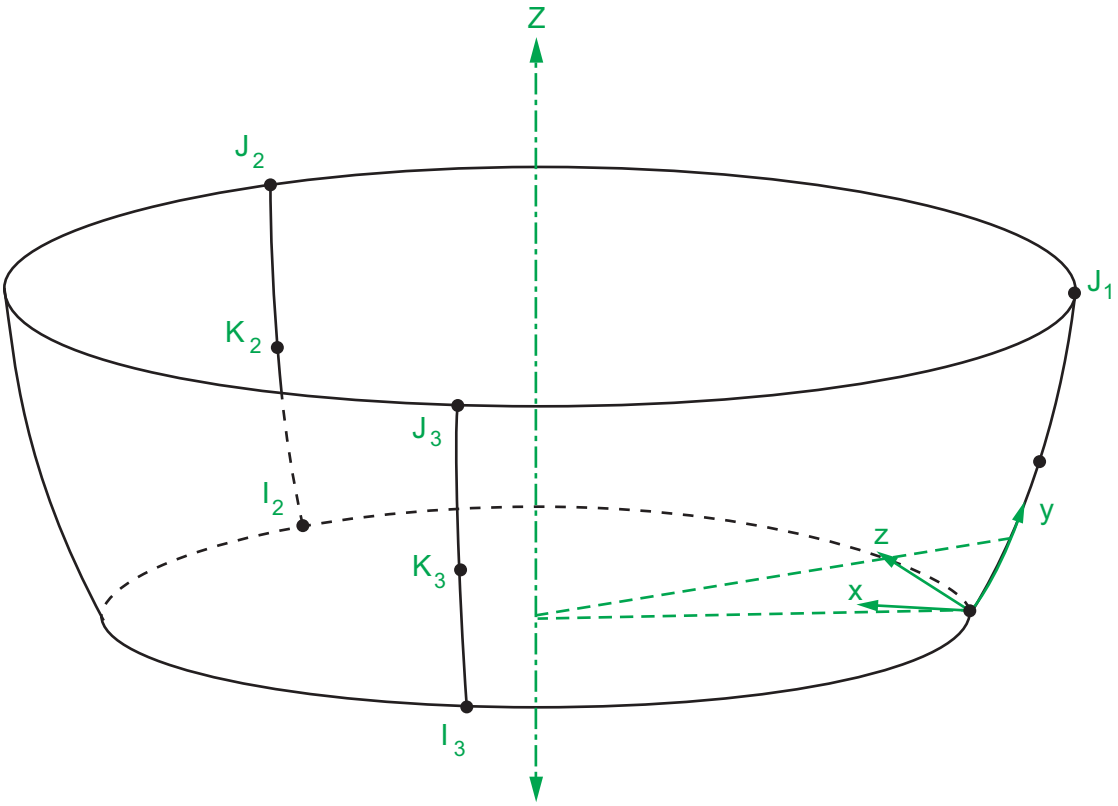
11.5.2. General Axisymmetric Surfaces

This section contains shape functions for 2- or 3- node-per-plane general axisymmetric surface elements such as SURF159. These elements are available in various configurations, including combinations of the following features:

- With or without midside nodes
- A varying number of node planes (N_{np}) in the circumferential direction (defined via KEYOPT(2)).

The elemental coordinates are cylindrical coordinates and displacements are defined and interpolated in that coordinate system, as shown in Figure 11.4: General Axisymmetric Surface Elements (when N_{np} = 3) (p. 334).

Figure 11.4: General Axisymmetric Surface Elements (when N_{hp} = 3)



When N_{np} is an odd number, the interpolation function used for displacement is:

$$u_i = h_i(s,t)(c_i + \sum_{m=1}^{\frac{NP-1}{2}} (a_i^m \cos m\theta + b_i^m \sin m\theta)) \tag{11.41}$$

where:

i = r, θ, z

h_i (s, t) = regular Lagrangian polynominal interpolation functions like Equation 11.6 (p. 329) or Equation 11.19 (p. 330).

c_i, a_i^m, b_i^m = coefficients for the Fourier terms.

When N_{np} is an even number, the interpolation function is:

$$u_i = h_i(s,t)(c_i + \sum_{m=1}^{\frac{NP-2}{2}} (a_i^m \cos m\theta + b_i^m \sin m\theta) + a_i^{\frac{NP}{2}} (\cos\frac{NP}{2}\theta + \sin\frac{NP}{2}\theta)) \tag{11.42}$$

All of the coefficients in Equation 11.41 (p. 334) and Equation 11.42 (p. 334) can be expressed by nodal displacements, using u_r = u, u_j = v, u_z = w without midside nodes, and N_{np} = 3:

$$u = \left(\frac{1}{2}(u_{I_1}(1-s)+u_{J_1}(1+s))\right)\left(\frac{1}{3}+\frac{2}{3}\cos\theta\right) + \left(\frac{1}{2}(u_{I_2}(1-s)+u_{J_2}(1+s))\right)\left(\frac{1}{3}-\frac{1}{3}\cos\theta+\frac{1}{\sqrt{3}}\sin\theta\right) + \left(\frac{1}{2}(u_{I_3}(1-s)+u_{J_3}(1+s))\right)\left(\frac{1}{3}-\frac{1}{3}\cos\theta-\frac{1}{\sqrt{3}}\sin\theta\right) \quad (11.43)$$

$$v = (\frac{1}{2}(v_{I_1}(1-s) \text{ . . . (analogous to u)} \quad (11.44)$$

$$w = (\frac{1}{2}(w_{I_1}(1-s) \text{ . . . (analogous to u)} \quad (11.45)$$

Similar to the element without midside nodes, the u, v, and w with midside nodes are expressed as:

$$u = \left(\frac{1}{2}(u_{I_1}(1-s)(-s)+u_{J_1}(1+s)(s)+(u_{M_1}(1-s^2)\right)\left(\frac{1}{3}+\frac{2}{3}\cos\theta\right) + \left(\frac{1}{2}(u_{I_2}(1-s)(-s)+u_{J_2}(1+s)(s)+(u_{M_2}(1-s^2)\right)\left(\frac{1}{3}-\frac{1}{3}\cos\theta+\frac{1}{\sqrt{3}}\sin\theta\right) + \left(\frac{1}{2}(u_{I_3}(1-s)(-s)+u_{J_3}(1+s)(s)+(u_{M_3}(1-s^2)\right)\left(\frac{1}{3}-\frac{1}{3}\cos\theta-\frac{1}{\sqrt{3}}\sin\theta\right) \quad (11.46)$$

$$v = (\frac{1}{2}(v_{I}(1-s) \text{ . . . (analogous to u)} \quad (11.47)$$

$$w = (\frac{1}{2}(w_{I}(1-s) \text{ . . . (analogous to w)} \quad (11.48)$$

11.6. 3-D Shells

This section contains shape functions for 3-D shell elements. These elements are available in a number of configurations, including certain combinations of the following features:

- triangular or quadrilateral.

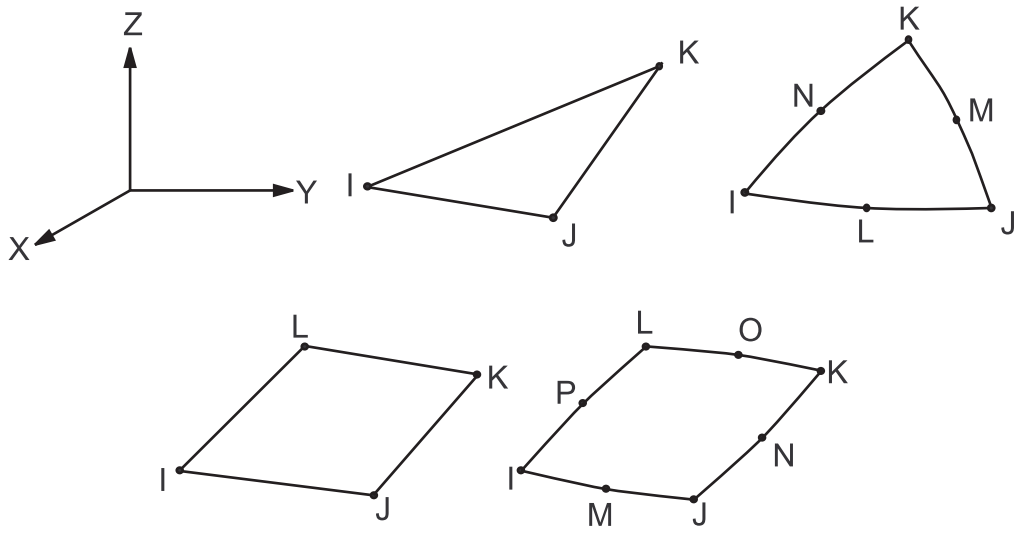
  - if quadrilateral, with or without extra shape functions (ESF).

- with or without rotational degrees of freedom (RDOF).

  - if with RDOF, with or without shear deflections (SD).

- with or without midside nodes.

Figure 11.5: 3-D Shell Elements



11.6.1. 3-D 3-Node Triangular Shells without RDOF (CST)

These shape functions are for 3-D 3-node triangular shell elements without RDOF, such as SHELL41, SHELL131, or SHELL132:

$$u = u_I L_1 + u_J L_2 + u_K L_3 \tag{11.49}$$

$$v = v_I L_1 + v_J L_2 + v_K L_3 \tag{11.50}$$

$$w = w_I L_1 + w_J L_2 + w_K L_3 \tag{11.51}$$

$$A_x = A_{xI} L_1 + A_{xJ} L_2 + A_{xK} L_3 \tag{11.52}$$

$$A_y = A_{yI} L_1 + A_{yJ} L_2 + A_{yK} L_3 \tag{11.53}$$

$$A_z = A_{zI} L_1 + A_{zJ} L_2 + A_{zK} L_3 \tag{11.54}$$

$$T = T_I L_1 + T_J L_2 + T_K L_3 \tag{11.55}$$

$$\phi = \phi_I L_1 + \phi_J L_2 + \phi_K L_3 \tag{11.56}$$

11.6.2. 3-D 6-Node Triangular Shells without RDOF (LST)

These shape functions are for 3-D 6-node triangular shell elements without RDOF, such as SHELL281 when used as a triangle:

$$u = u_I(2L_1 - 1)L_1 + u_J(2L_2 - 1)L_2 + u_K(2L_3 - 1)L_3 + u_L(4L_1L_2) + u_M(4L_2L_3) + u_N(4L_3L_1) \tag{11.57}$$

$$v = v_I(2L_1 - 1) \text{ . . . (analogous to u)} \tag{11.58}$$

$$w = w_I(2L_1 - 1) \text{ . . . (analogous to u)} \tag{11.59}$$

$$\theta_x = \theta_x(2L_1 - 1) \text{ . . . (analogous to u)} \tag{11.60}$$

$$\theta_y = \theta_y(2L_1 - 1) \text{ . . . (analogous to u)} \tag{11.61}$$

$$\theta_z = \theta_z(2L_1 - 1) \text{ . . . (analogous to u)} \tag{11.62}$$

$$T = T_I(2L_1 - 1) \text{ . . . (analogous to u)} \tag{11.63}$$

$$V = V_I(2L_1 - 1) \text{ . . . (analogous to u)} \tag{11.64}$$

$$C=C_I(2L_1\text{-}1) \text{ . . . (analogous to u)} \tag{11.65}$$

11.6.3. 3-D 3-Node Triangular Shells with RDOF but without SD

These shape functions are for the 3-D 3-node triangular shell elements with RDOF, but without shear deflection.

$$u = u_I L_1 + u_J L_2 + u_K L_3 \tag{11.66}$$

$$v = v_I L_1 + v_J L_2 + v_K L_3 \tag{11.67}$$

$$w = \text{not explicitly defined. A DKT element is used} \tag{11.68}$$

11.6.4. 3-D 4-Node Quadrilateral Shells without RDOF and without ESF (Q4)

These shape functions are for 3-D 4-node triangular shell elements without RDOF and without extra displacement shapes, such as SHELL41 with KEYOPT(2) = 1.

$$u = \frac{1}{4}(u_I(1-s)(1-t) + u_J(1+s)(1-t) + u_K(1+s)(1+t) + u_L(1-s)(1+t)) \tag{11.69}$$

$$v = \frac{1}{4}(v_I(1-s) \text{ . . . (analogous to u)} \tag{11.70}$$

$$w = \frac{1}{4}(w_I(1-s) \text{ . . . (analogous to u)} \tag{11.71}$$

$$\theta_x = \frac{1}{4}(\theta_x(1-s) \text{ . . . (analogous to u)} \tag{11.72}$$

$$\theta_y = \frac{1}{4}(\theta_y(1-s) \text{ . . . (analogous to u)} \tag{11.73}$$

$$\theta_z = \frac{1}{4}(\theta_z(1-s) \text{ . . . (analogous to u)} \tag{11.74}$$

$$A_x = \frac{1}{4}(A_{xI}(1-s) \text{ . . . (analogous to u)} \tag{11.75}$$

$$A_y = \frac{1}{4}(A_{yI}(1-s) \text{ . . . (analogous to u)} \tag{11.76}$$

$$A_z = \frac{1}{4}(A_{zI}(1-s) \text{ . . . (analogous to u)} \tag{11.77}$$

$$P = \frac{1}{4}(P_I(1-s) \text{ . . . (analogous to u)} \tag{11.78}$$

$$T = \frac{1}{4}(T_I(1-s) \text{ . . . (analogous to u)} \tag{11.79}$$

$$V = \frac{1}{4}(V_I(1-s) \ldots \text{(analogous to u)} \tag{11.80}$$

$$\phi = \frac{1}{4}(\phi_I(1-s) \ldots \text{(analogous to u)} \tag{11.81}$$

11.6.5. 3-D 4-Node Quadrilateral Shells without RDOF but with ESF (QM6)

These shape functions are for 3-D 4-node quadrilateral shell elements without RDOF but with extra shape functions, such as SHELL41 with KEYOPT(2) = 0:

$$\begin{aligned} u = \frac{1}{4}&(u_I(1-s)(1-t) + u_J(1+s)(1-t) \\ &+ u_K(1+s)(1+t) + u_L(1-s)(1+t)) \\ &+ u_1(1-s^2) + u_2(1-t^2) \end{aligned} \tag{11.82}$$

$$v = \frac{1}{4}(v_I(1-s) \ldots \text{(analogous to u)} \tag{11.83}$$

11.6.6. 3-D 8-Node Quadrilateral Shells without RDOF

These shape functions are for 3-D 8-node quadrilateral shell elements without RDOF, such as SHELL281:

$$\begin{aligned} u = \frac{1}{4}&(u_I(1-s)(1-t)(-s-t-1) + u_J(1+s)(1-t)(s-t-1) \\ &+ u_K(1+s)(1+t)(s+t-1) + u_L(1-s)(1+t)(-s+t-1)) \\ &+ \frac{1}{2}(u_M(1-s^2)(1-t) + u_N(1+s)(1-t^2) \\ &+ u_O(1-s^2)(1+t) + u_P(1-s)(1-t^2)) \end{aligned} \tag{11.84}$$

$$v = \frac{1}{4}(v_I(1-s) \ldots \text{(analogous to u)} \tag{11.85}$$

$$w = \frac{1}{4}(w_I(1-s) \ldots \text{(analogous to u)} \tag{11.86}$$

$$\theta_x = \frac{1}{4}(\theta_x(1-s) \ldots \text{(analogous to u)} \tag{11.87}$$

$$\theta_y = \frac{1}{4}(\theta_y(1-s) \ldots \text{(analogous to u)} \tag{11.88}$$

$$\theta_z = \frac{1}{4}(\theta_z(1-s) \ldots \text{(analogous to u)} \tag{11.89}$$

$$P = \frac{1}{4}(P_I(1-s) \ldots \text{(analogous to u)} \tag{11.90}$$

$$T = \frac{1}{4}(T_I(1-s) \ldots \text{(analogous to u)} \tag{11.91}$$

$$V = \frac{1}{4}(V_I(1-s) \ldots \text{(analogous to u)} \tag{11.92}$$

$$C=\frac{1}{4}(C_I(1\text{-}s) \text{ . . . (analogous to u)} \tag{11.93}$$

11.6.7. 3-D 4-Node Quadrilateral Shells with RDOF but without SD and without ESF

These shape functions are for 3-D 4-node quadrilateral shell elements with RDOF but without shear deflection and without extra shape functions:

$$u=\frac{1}{4}(u_I(1-s)(1-t)+u_J(1+s)(1-t) +u_K(1+s)(1+t)+u_L(1-s)(1+t)) \tag{11.94}$$

$$v=\frac{1}{4}(v_I(1-s) \text{ . . . (analogous to u)} \tag{11.95}$$

$$w = \text{not explicitly defined. Four overlaid triangles} \tag{11.96}$$

11.6.8. 3-D 4-Node Quadrilateral Shells with RDOF but without SD and with ESF

These shape functions are for 3-D 4-node quadrilateral shell elements with RDOF but without shear deflection and with extra shape functions:

$$u=\frac{1}{4}(u_I(1-s)(1-t)+u_J(1+s)(1-t) +u_K(1+s)(1+t)+u_L(1-s)(1+t)) +u_1(1-s^2)+u_2(1-t^2) \tag{11.97}$$

$$v=\frac{1}{4}(v_I(1-s) \text{ . . . (analogous to u)} \tag{11.98}$$

$$w = \text{not explicitly defined. Four overlaid triangles} \tag{11.99}$$

11.7. 2-D and Axisymmetric Solids

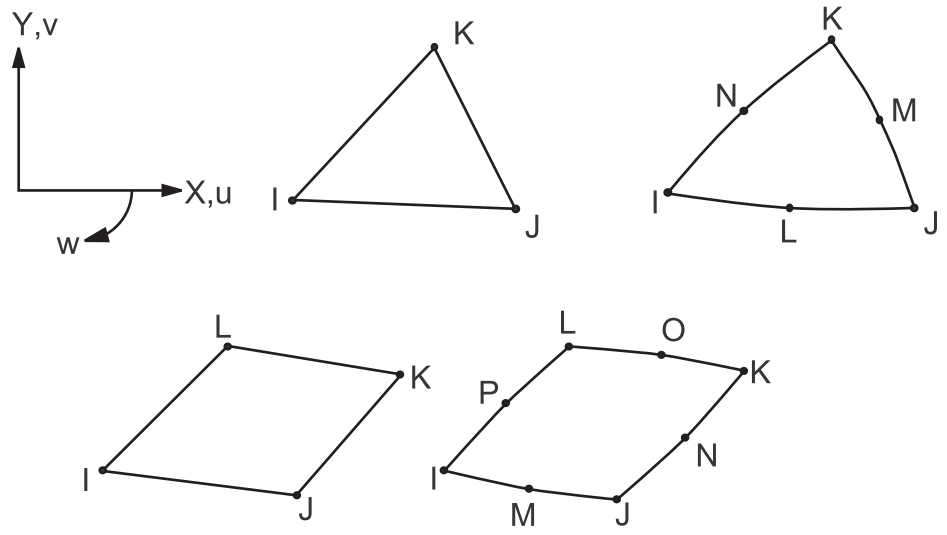
This section contains shape functions for 2-D and axisymmetric solid elements. These elements are available in a number of configurations, including certain combinations of the following features:

- triangular or quadrilateral.

  - if quadrilateral, with or without extra shape functions (ESF).

- with or without midside nodes.

Figure 11.6: 2-D and Axisymmetric Solid Element



11.7.1. 2-D and Axisymmetric 3-Node Triangular Solids (CST)

These shape functions are for 2-D 3-node and axisymmetric triangular solid elements, such as PLANE13 or PLANE182 with only 3 nodes input:

$$u = u_I L_1 + u_J L_2 + u_K L_3 \tag{11.100}$$

$$v = v_I L_1 + v_J L_2 + v_K L_3 \tag{11.101}$$

$$w = w_I L_1 + w_J L_2 + w_K L_3 \tag{11.102}$$

$$A_z = A_{zI} L_1 + A_{zJ} L_2 + A_{zK} L_3 \tag{11.103}$$

$$V_x = V_{xI} L_1 + A_{zJ} L_2 + A_{zK} L_3 \tag{11.104}$$

$$V_y = V_{yI} L_1 + A_{zJ} L_2 + A_{zK} L_3 \tag{11.105}$$

$$V_z = V_{zI} L_1 + A_{zJ} L_2 + A_{zK} L_3 \tag{11.106}$$

$$P = P_I L_1 + A_{zJ} L_2 + A_{zK} L_3 \tag{11.107}$$

$$T = T_I L_1 + T_J L_2 + T_K L_3 \tag{11.108}$$

$$V = V_I L_1 + V_J L_2 + V_K L_3 \tag{11.109}$$

$$E^K = E_I^K L_1 + V_J L_2 + V_K L_3 \tag{11.110}$$

$$E^D = E_I^D L_1 + V_J L_2 + V_K L_3 \tag{11.111}$$

11.7.2. 2-D and Axisymmetric 6-Node Triangular Solids (LST)

These shape functions are for 2-D 6-node and axisymmetric triangular solids, such as PLANE35:

$$\begin{aligned} u = {} & u_I(2L_1 - 1)L_1 + u_J(2L_2 - 1)L_2 + u_K(2L_3 - 1) \\ & + u_L(4L_1L_2) + u_M(4L_2L_3) + u_N(4L_3L_1) \end{aligned} \tag{11.112}$$

$$v = v_I(2L_1 - 1)L_1 + \text{ . . . (analogous to u)} \tag{11.113}$$

$$w = w_I(2L_1 - 1)L_1 + \ldots \text{ (analogous to u)} \tag{11.114}$$

$$A_z = A_{zI}(2L_1 - 1)L_1 \ldots \text{ (analogous to u)} \tag{11.115}$$

$$P = P_I(2L_1 - 1)L_1 + \ldots \text{ (analogous to u)} \tag{11.116}$$

$$T = T_I(2L_1 - 1)L_1 + \ldots \text{ (analogous to u)} \tag{11.117}$$

$$V = V_I(2L_1 - 1)L_1 + \ldots \text{ (analogous to u)} \tag{11.118}$$

$$C=C_i\left(2L_1\text{-}1\right)L_1\text{+ ... (analogous to u)} \tag{11.119}$$

11.7.3. 2-D and Axisymmetric 4-node Quadrilateral Solid without ESF (Q4)

These shape functions are for the 2-D 4-node and axisymmetric quadrilateral solid elements without extra shape functions, such as PLANE13 with KEYOPT(2) = 1 or LINK68.

$$u = \frac{1}{4}(u_I(1-s)(1-t) + u_J(1+s)(1-t) + u_K(1+s)(1+t) + u_L(1-s)(1+t)) \tag{11.120}$$

$$v = \frac{1}{4}(v_I(1-s) \ldots \text{ (analogous to u)} \tag{11.121}$$

$$w = \frac{1}{4}(w_I(1-s) \ldots \text{ (analogous to u)} \tag{11.122}$$

$$A_z = \frac{1}{4}(A_{zI}(1-s) \ldots \text{ (analogous to u)} \tag{11.123}$$

$$V_x = \frac{1}{4}V_{xI}(1-s) \ldots \text{ (analogous to u)} \tag{11.124}$$

$$V_y = \frac{1}{4}V_{yI}(1-s) \ldots \text{ (analogous to u)} \tag{11.125}$$

$$V_z = \frac{1}{4}V_{zI}(1-s) \ldots \text{ (analogous to u)} \tag{11.126}$$

$$P = \frac{1}{4}P_I(1-s) \ldots \text{ (analogous to u)} \tag{11.127}$$

$$T = \frac{1}{4}(T_I(1-s) \ldots \text{ (analogous to u)} \tag{11.128}$$

$$V = \frac{1}{4}(V_I(1-s) \ldots \text{ (analogous to u)} \tag{11.129}$$

$$E^K = \frac{1}{4}(E_I^K(1-s) \ldots \text{ (analogous to u)} \tag{11.130}$$

$$E^D = \frac{1}{4}(E_I^D(1-s) \ldots \text{ (analogous to u)} \tag{11.131}$$

11.7.4. 2-D and Axisymmetric 4-node Quadrilateral Solids with ESF (QM6)

These shape functions are for the 2-D 4-node and axisymmetric solid elements with extra shape functions, such as PLANE13 with KEYOPT(2) = 0. (Taylor et al.([49] (p. 923)))

$$u = \frac{1}{4}(u_I(1-s)(1-t) + u_J(1+s)(1-t) + u_K(1+s)(1+t) + u_L(1-s)(1+t)) + u_1(1-s^2) + u_2(1-t^2) \quad (11.132)$$

$$v = \frac{1}{4}(v_I(1-s) \ldots \text{(analogous to u)} \quad (11.133)$$

Equation 11.132 (p. 342) is adjusted for axisymmetric situations by removing the u_1 or u_2 term for elements near the centerline, in order to avoid holes or "doubled" material at the centerline.

11.7.5. 2-D and Axisymmetric 8-Node Quadrilateral Solids (Q8)

These shape functions are for the 2-D 8-node and axisymmetric quadrilateral elements such as PLANE77 and PLANE183:

$$u = \frac{1}{4}(u_I(1-s)(1-t)(-s-t-1) + u_J(1+s)(1-t)(s-t-1) + u_K(1+s)(1+t)(s+t-1) + u_L(1-s)(1+t)(-s+t-1)) + \frac{1}{2}(u_M(1-s^2)(1-t) + u_N(1+s)(1-t^2) + u_O(1-s^2)(1+t) + u_P(1-s)(1-t^2)) \quad (11.134)$$

$$v = \frac{1}{4}(v_I(1-s) \ldots \text{(analogous to u)} \quad (11.135)$$

$$w = \frac{1}{4}(w_I(1-s) \ldots \text{(analogous to u)} \quad (11.136)$$

$$A_z = \frac{1}{4}(A_{zI}(1-s) \ldots \text{(analogous to u)} \quad (11.137)$$

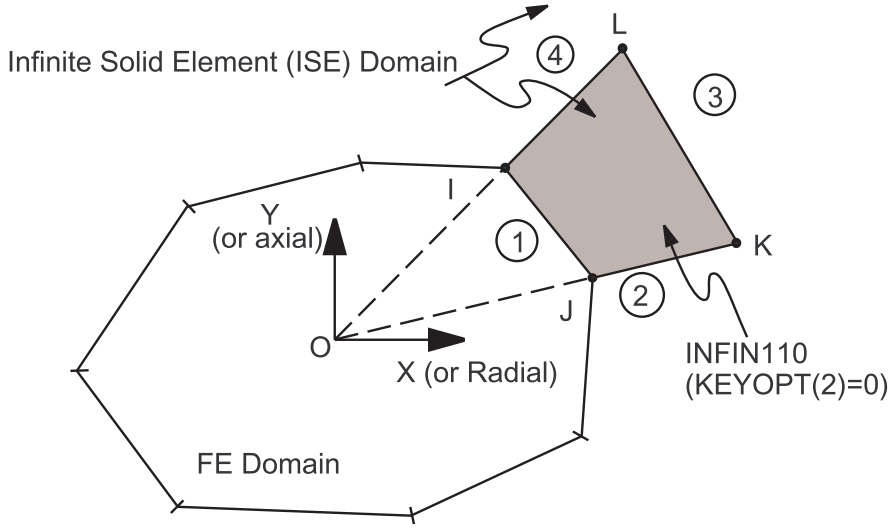
$$T = \frac{1}{4}(T_I(1-s) \ldots \text{(analogous to u)} \quad (11.138)$$

$$V = \frac{1}{4}(V_I(1-s) \ldots \text{(analogous to u)} \quad (11.139)$$

$$C = \frac{1}{4}(C_i(1\text{-}s) \ldots \text{(analogous to u)} \quad (11.140)$$

11.7.6. 2-D and Axisymmetric 4-Node Quadrilateral Infinite Solids

Figure 11.7: 4-Node Quadrilateral Infinite Solid Element



These Lagrangian isoparametric shape functions and "mapping" functions are for the 2-D and axisymmetric 4-node quadrilateral solid infinite elements such as INFIN110:

11.7.6.1. Lagrangian Isoparametric Shape Functions

$$A_z = \frac{1}{4}(A_{zI}(1-s)(t^2-t) + A_{zJ}(1+s)(t^2-t)) + \frac{1}{2}(A_{zK}(1+s)(1-t^2) + A_{zL}(1-s)(1-t^2)) \quad (11.141)$$

$$T = \frac{1}{4}(T_I(1-s) \ldots \text{(analogous to } A_z) \quad (11.142)$$

$$V = \frac{1}{4}(V_I(1-s) \ldots \text{(analogous to } A_z) \quad (11.143)$$

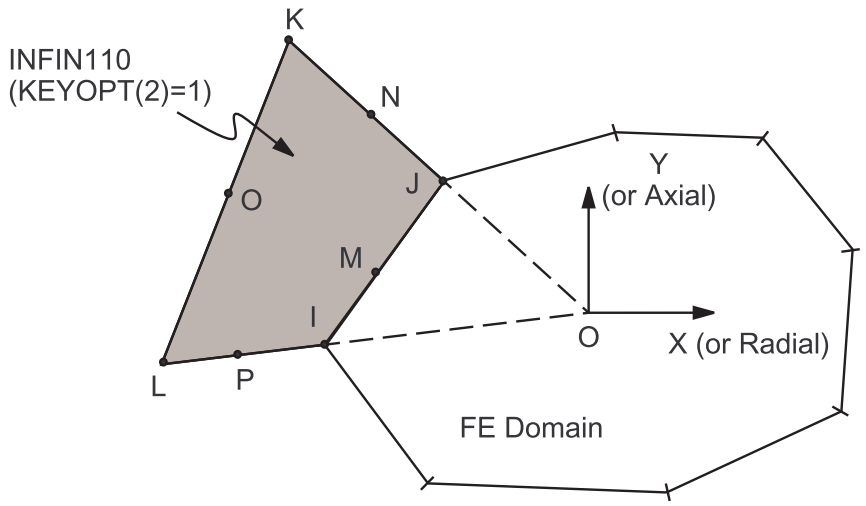
11.7.6.2. Mapping Functions

$$x = x_I(1-s)(-t)/(1-t) + x_J(1+s)(-t)/(1-t) + \frac{1}{2}X_K(1+s)(1+t)/(1-t) + \frac{1}{2}X_L(1-s)(1+t)/(1-t) \quad (11.144)$$

$$y = y_I(1-s) \ldots \text{(analogous to x)} \quad (11.145)$$

11.7.7. 2-D and Axisymmetric 8-Node Quadrilateral Infinite Solids

Figure 11.8: 8-Node Quadrilateral Infinite Solid Element



These Lagrangian isoparametric shape functions and "mapping" functions are for the 2-D and axisymmetric 8-node quadrilateral infinite solid elements such as INFIN110:

11.7.7.1. Lagrangian Isoparametric Shape Functions

$$A_z = \frac{1}{4}(A_{zI}(1-s)(1-t)(-1-s-t)) + \frac{1}{2}(A_{zJ}(1-s^2)(1-t)) + \frac{1}{4}(A_{zK}(1+s)(1-t)(-1+s-t)) + \frac{1}{2}(A_{zL}(1+s)(1-t^2)) + \frac{1}{2}(A_{zM}(1-s)(1-t^2)) \quad (11.146)$$

$$T = (T_I(1-s) \ldots \text{(analogous to } A_z) \quad (11.147)$$

$$V = (V_I(1-s) \ldots \text{(analogous to } A_z) \quad (11.148)$$

11.7.7.2. Mapping Functions

$$x = x_I(1-s)(-1-s-t)/(1-t) + 2x_J(1-s^2)/(1-t) + x_K(1+s)(-1+s-t)/(1-t) + \frac{1}{2}x_L(1+s)(1+t)/(1-t) + \frac{1}{2}x_M(1+s)(1+t)/(1-t) \quad (11.149)$$

$$y = y_I(1-s) \ldots \text{(analogous to x)} \quad (11.150)$$

The shape and mapping functions for the nodes N, O and P are deliberately set to zero.

11.8. Axisymmetric Harmonic Solids

This section contains shape functions for axisymmetric harmonic solid elements. These elements are available in a number of configurations, including certain combinations of the following features:

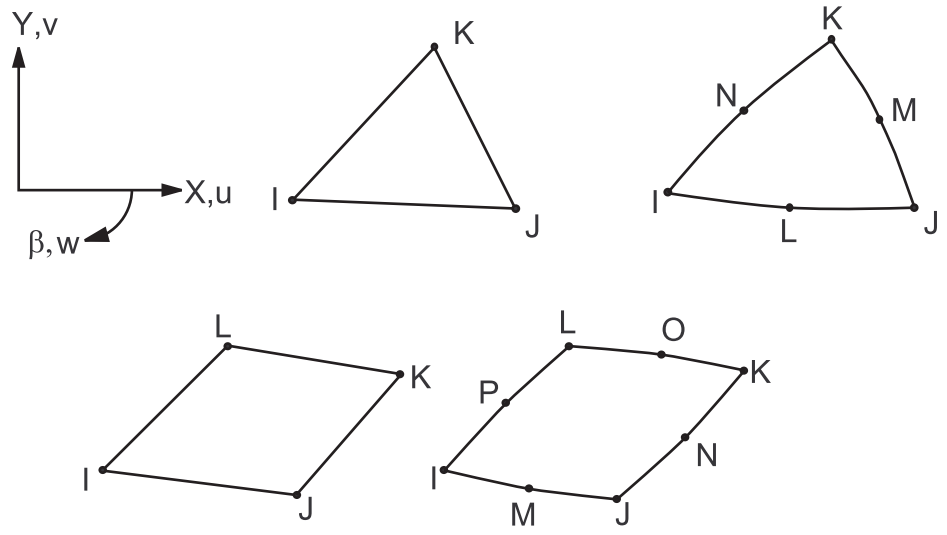
- triangular or quadrilateral.

  - if quadrilateral, with or without extra shape functions (ESF).

- with or without midside nodes.

The shape functions of this section use the quantities $\sin \ell \beta$ and $\cos \ell \beta$ (where ℓ = input as MODE on the **MODE** command). $\sin \ell \beta$ and $\cos \ell \beta$ are interchanged if I_s = -1 (where I_s = input as ISYM on the **MODE** command). If ℓ = 0, $\sin \ell \beta = \cos \ell \beta = 1.0$.

Figure 11.9: Axisymmetric Harmonic Solid Elements



11.8.1. Axisymmetric Harmonic 3-Node Triangular Solids

These shape functions are for the 3-node axisymmetric triangular solid elements, such as PLANE25 with only 3 nodes input:

$$u = (u_I L_1 + u_J L_2 + u_K L_3)\cos \ell \beta \quad (11.151)$$

$$v = (v_I L_1 + v_J L_2 + v_K L_3)\cos \ell \beta \quad (11.152)$$

$$w = (w_I L_1 + w_J L_2 + w_K L_3)\sin \ell \beta \quad (11.153)$$

$$T = (T_I L_1 + T_J L_2 + T_K L_3)\cos \ell \beta \quad (11.154)$$

11.8.2. Axisymmetric Harmonic 6-Node Triangular Solids

These shape functions are for the 6-node axisymmetric triangular solids elements, such as PLANE83 input as a triangle:

$$u = (u_I(2L_1 - 1)L_1 + u_J(2L_2 - 1)L_2 + u_K(2L_3 - 1) + u_L(4L_1L_2) + u_M(4L_2L_3) + u_N(4L_3L_1))\cos \ell \beta \quad (11.155)$$

$$v = (v_I(2L_1 - 1) \ldots \text{(analogous to u)} \ldots)\cos \ell \beta \quad (11.156)$$

$$w = (w_I(2L_1 - 1) \ldots \text{(analogous to u)} \ldots)\cos \ell \beta \quad (11.157)$$

$$T = (T_I(2L_1 - 1) \ldots \text{(analogous to u)}\ldots)\cos \ell\beta \tag{11.158}$$

11.8.3. Axisymmetric Harmonic 4-Node Quadrilateral Solids without ESF

These shape functions are for the 4-node axisymmetric harmonic quadrilateral solid elements without extra shape functions, such as PLANE25 with KEYOPT(2) = 1, or PLANE75:

$$\begin{aligned} u = \frac{1}{4}(&u_I(1-s)(1-t) + u_J(1+s)(1-t) \\ &+ u_K(1+s)(1+t) + u_L(1-s)(1+t))\cos \ell\beta \end{aligned} \tag{11.159}$$

$$v = \frac{1}{4}(v_I(1-s)\ldots \text{(analogous to u)}\ldots)\cos \ell\beta \tag{11.160}$$

$$w = \frac{1}{4}(w_I(1-s)\ldots \text{(analogous to u)}\ldots)\sin \ell\beta \tag{11.161}$$

$$T = \frac{1}{4}(T_I(1-s)\ldots \text{(analogous to u)}\ldots)\cos \ell\beta \tag{11.162}$$

11.8.4. Axisymmetric Harmonic 4-Node Quadrilateral Solids with ESF

These shape functions are for the 4-node axisymmetric harmonic quadrilateral elements with extra shape functions, such as PLANE25 with KEYOPT(2) = 0.

$$\begin{aligned} u = (\frac{1}{4}(&u_I(1-s)(1-t) + u_J(1+s)(1-t) + u_K(1+s)(1+t) \\ &+ u_L(1-s)(1+t)) + u_1(1-s^2) + u_2(1-t^2)))\cos \ell\beta \end{aligned} \tag{11.163}$$

$$v = (\frac{1}{4}(v_I(1-s)\ldots \text{(analogous to u)}\ldots)\cos \ell\beta \tag{11.164}$$

$$w = (\frac{1}{4}(w_I(1-s)\ldots \text{(analogous to u)}\ldots)\sin \ell\beta \tag{11.165}$$

Unless ℓ (MODE) = 1, u_1 or u_2 and w_1 or w_2 motions are suppressed for elements near the centerline.

11.8.5. Axisymmetric Harmonic 8-Node Quadrilateral Solids

These shape functions are for the 8-node axisymmetric harmonic quadrilateral solid elements such as PLANE78 or PLANE83.

$$\begin{aligned} u = (\frac{1}{4}(&u_I(1-s)(1-t)(-s-t-1) + u_J(1+s)(1-t)(s-t-1) \\ &+ u_K(1+s)(1+t)(s+t-1) + u_L(1-s)(1+t)(-s+t-1)) \\ &+ \frac{1}{2}(u_M(1-s^2)(1-t) + u_N(1+s)(1-t^2) \\ &+ u_O(1-s^2)(1+t) + u_P(1-s)(1-t^2)))\cos \ell\beta \end{aligned} \tag{11.166}$$

$$v = (\frac{1}{4}(v_I(1-s)\ldots \text{(analogous to u)}\ldots)\cos \ell\beta \tag{11.167}$$

$$w = (\frac{1}{4}(w_I(1-s)\ldots\text{(analogous to u)}\ldots)\sin\ell\beta \quad (11.168)$$

$$T = \frac{1}{4}(T_I(1-s)\ldots\text{(analogous to u)}\ldots)\cos\ell\beta \quad (11.169)$$

11.9. 3-D Solids

This section contains shape functions for 3-D solid elements. These elements are available in a number of configurations, including certain combinations of the following features:

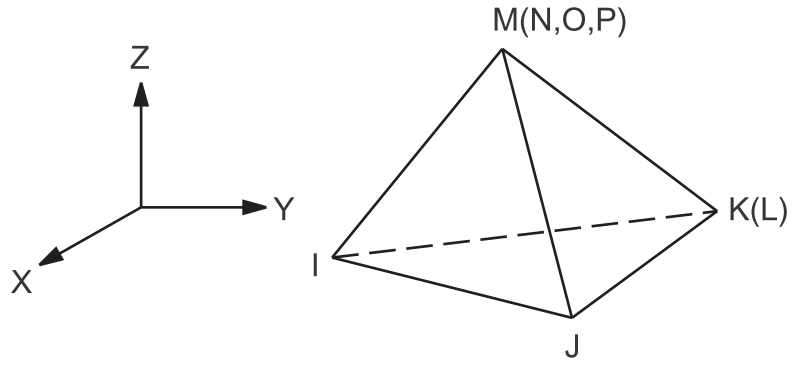
- Element shapes may be tetrahedra, pyramids, wedges, or bricks (hexahedra).
  - If wedges or bricks, with or without extra shape functions (ESF)
- With or without rotational degrees of freedom (RDOF)
- With or without midside nodes

The wedge elements with midside nodes (15-node wedges) are either a condensation of the 20-node brick element or are based on wedge shape functions.

11.9.1. 4-Node Tetrahedra

These shape functions are either a direct 4-node tetrahedral such as SOLID285 or a condensation of an 8-node brick element such as SOLID5, FLUID30, or SOLID98.

Figure 11.10: 3-D Solid Elements



The resulting effective shape functions are:

$$u = u_I L_1 + u_J L_2 + u_K L_3 + u_M L_4 \quad (11.170)$$

$$v = v_I L_1 + v_J L_2 + v_K L_3 + v_M L_4 \quad (11.171)$$

$$w = w_I L_1 + w_J L_2 + w_K L_3 + w_M L_4 \quad (11.172)$$

$$V_x = V_{xI} L_1 + w_J L_2 + w_K L_3 + w_M L_4 \quad (11.173)$$

$$V_y = V_{yI} L_1 + w_J L_2 + w_K L_3 + w_M L_4 \quad (11.174)$$

$$V_z = V_{zI} L_1 + w_J L_2 + w_K L_3 + w_M L_4 \quad (11.175)$$

$$P = P_I L_1 + P_J L_2 + P_K L_3 + P_M L_4 \quad (11.176)$$

$$T = T_I L_1 + T_J L_2 + T_K L_3 + T_M L_4 \quad (11.177)$$

$$V = V_I L_1 + V_J L_2 + V_K L_3 + V_M L_4 \quad (11.178)$$

$$\phi = \phi_I L_1 + \phi_J L_2 + \phi_K L_3 + \phi_M L_4 \quad (11.179)$$

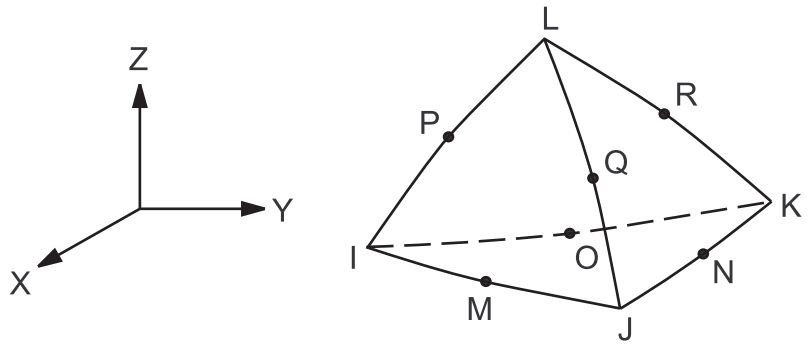
$$E^K = E_I^K L_1 + E_J^K L_2 + E_K^K L_3 + E_M^K L_4 \quad (11.180)$$

$$E^D = E_I^D L_1 + E_J^D L_2 + E_K^D L_3 + E_M^D L_4 \quad (11.181)$$

11.9.2. 10-Node Tetrahedra

These shape functions are for 10-node tetrahedron elements such as SOLID98 and SOLID227, or by condensation for SOLID90.

Figure 11.11: 10-Node Tetrahedra Element



$$\begin{aligned} u = {} & u_I(2L_1 - 1)L_1 + u_J(2L_2 - 1)L_2 + u_K(2L_3 - 1)L_3 \\ & + u_L(2L_4 - 1)L_4 + 4u_M L_1 L_2 + u_N L_2 L_3 + u_O L_1 L_3 \\ & + u_P L_1 L_4 + u_Q L_2 L_4 + u_R L_3 L_4 \end{aligned} \quad (11.182)$$

$$v = v_I(2L_1 - 1)L_1 + \ \ldots \text{ (analogous to u)} \quad (11.183)$$

$$w = w_I(2L_1 - 1)L_1 + \ \ldots \text{ (analogous to u)} \quad (11.184)$$

$$T = T_I(2L_1 - 1)L_1 + \ \ldots \text{ (analogous to u)} \quad (11.185)$$

$$V = V_I(2L_1 - 1)L_1 + \ \ldots \text{ (analogous to u)} \quad (11.186)$$

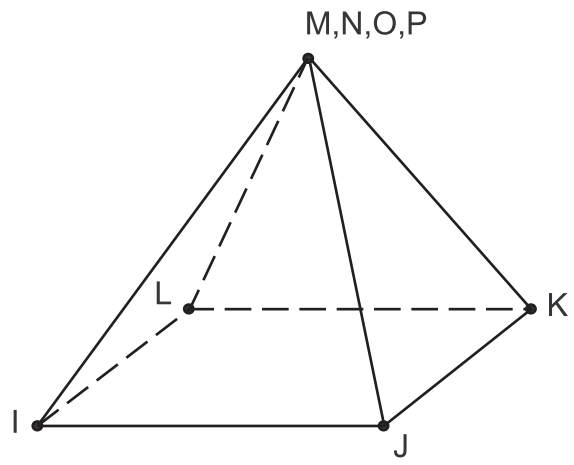
$$\phi = \phi_I(2L_1 - 1)L_1 + \ \ldots \text{ (analogous to u)} \quad (11.187)$$

$$C = C_I(2L_1 - 1)L_1 + \ \ldots \text{ (analogous to u)} \quad (11.188)$$

11.9.3. 5-Node Pyramids

This element is a condensation of an 8-node brick element.

Figure 11.12: 8-Node Brick Element



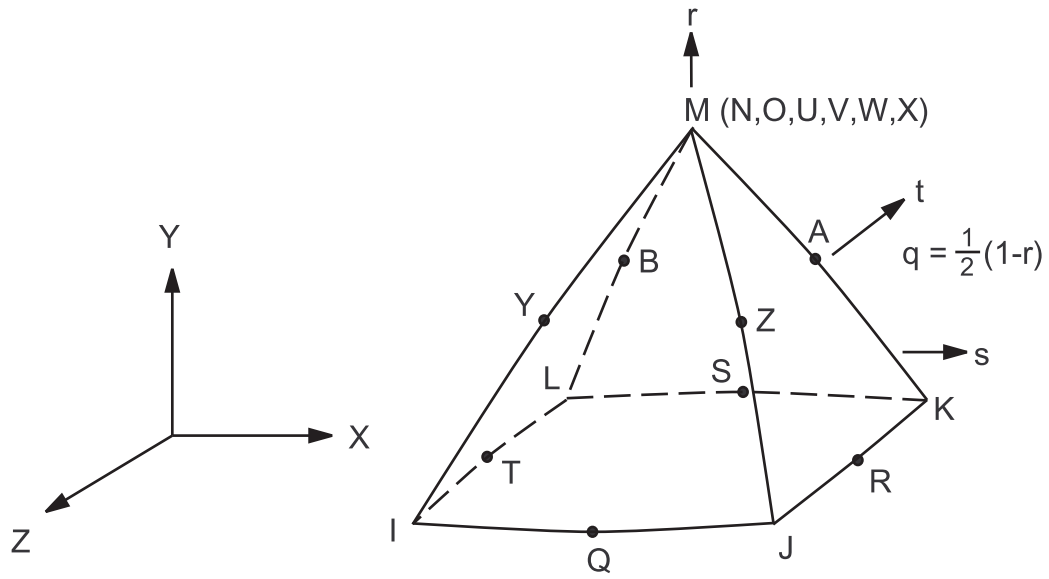
The resulting effective shape functions are:

$$T = \frac{1}{8}T_I(1-s)(1-t)(1-r) + T_J(1+s)(1-t)(1-r) + T_K(1+s)(1+t)(1-r) + T_L(1-s)(1+t)(1-r) + \frac{1}{2}T_M(1+r) \quad (11.189)$$

11.9.4. 13-Node Pyramids

These shape functions are for 13-node pyramid elements which are based on a condensation of a 20-node brick element:

Figure 11.13: 13-Node Pyramid Element



$$u = \frac{q}{4}(u_I(1-s)(1-t)(-1-qs-qt) + u_J(1+s)(1-t)(-1+qs-qt) + u_K(1+s)(1+t)(-1+sq+qt) + u_L(1-s)(1+t)(-1-qs+qt)) + u_M(1-q)(1-2q) + \frac{q^2}{2}(u_Q(1-t)(1-s^2) + u_R(1+s)(1-t^2) + u_S(1+t)(1-s^2) + u_T(1-s)(1-t^2)) + q(1-q)(u_Y(1-s-t+st) + u_Z(1+s-t-st) + u_A)(1+s+t+st) + u_B(1-s+t-st)) \quad (11.190)$$

$$v = \frac{q}{4}(v_I(1-s)\ldots \text{(analogous to u)} \quad (11.191)$$

$$w = \frac{q}{4}(w_I(1-s)\ldots \text{(analogous to u)} \quad (11.192)$$

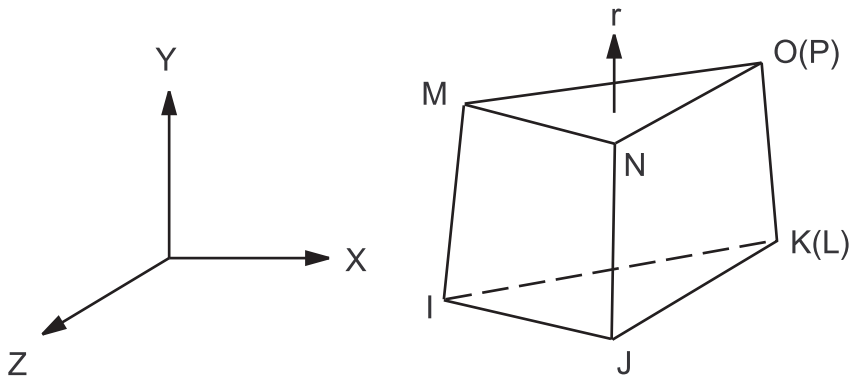
$$T = \frac{q}{4}(T_I(1-s)\ldots \text{(analogous to u)} \quad (11.193)$$

$$V = \frac{q}{4}(V_I(1-s)\ldots \text{(analogous to u)} \quad (11.194)$$

$$C = \frac{q}{4}(C_I(1-s)\ldots \text{(analogous to u)} \quad (11.195)$$

11.9.5. 6-Node Wedges without ESF

Figure 11.14: 6-Node Wedge Element



The 6-node wedge elements are a condensation of an 8-node brick such as SOLID5 or FLUID30. These shape functions are for 6-node wedge elements without extra shape functions:

$$u = \frac{1}{2}u_I L_1(1-r) + u_J L_2(1-r) + u_K L_3(1-r) + u_M L_1(1+r) + u_N L_2(1+r) + u_O L_3(1+r) \quad (11.196)$$

$$v = \frac{1}{2}(v_I L_1(1-r)\ldots \text{(analogous to u)} \quad (11.197)$$

$$w = \frac{1}{2}(w_I L_1(1-r). \ . \ . \text{ (analogous to u)} \tag{11.198}$$

$$P = \frac{1}{2}(P_I L_1(1-r). \ . \ . \text{ (analogous to u)} \tag{11.199}$$

$$T = \frac{1}{2}(T_I L_1(1-r). \ . \ . \text{ (analogous to u)} \tag{11.200}$$

$$V = \frac{1}{2}(V_I L_1(1-r). \ . \ . \text{ (analogous to u)} \tag{11.201}$$

$$\phi = \frac{1}{2}(\phi_I L_1(1-r). \ . \ . \text{ (analogous to u)} \tag{11.202}$$

11.9.6. 6-Node Wedges with ESF

The 6-node wedge elements are a condensation of an 8-node brick such as SOLID5. (See Figure 11.14: 6-Node Wedge Element (p. 350).) These shape functions are for 6-node wedge elements with extra shape functions:

$$u = \frac{1}{2}(u_I L_1(1-r) + u_J L_2(1-r) + u_K L_3(1-r) + u_M L_1(1+r) + u_N L_2(1+r) + u_O L_3(1+r) + u_1(1-r^2)) \tag{11.203}$$

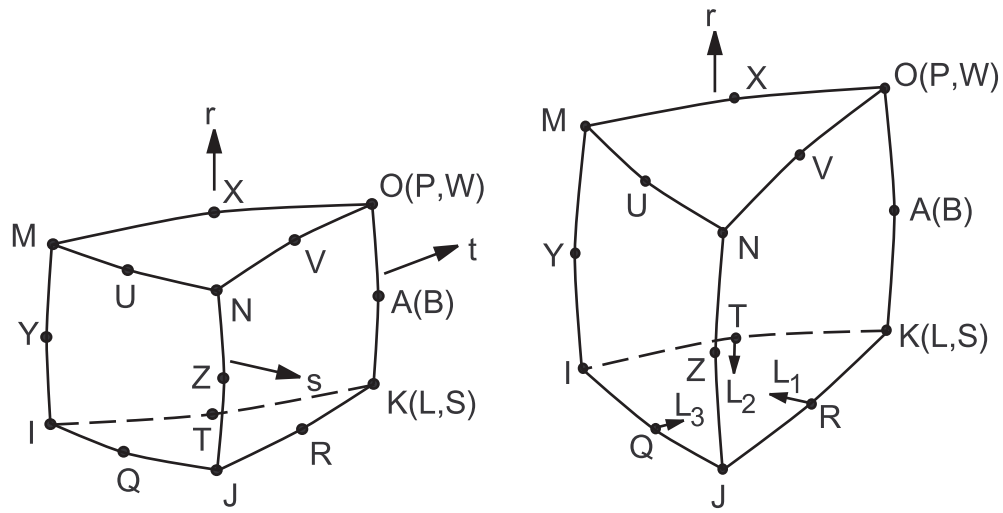
$$v = \frac{1}{2}(v_I L_1(1-r). \ . \ . \text{ (analogous to u)} \tag{11.204}$$

$$w = \frac{1}{2}(w_I L_1(1-r). \ . \ . \text{ (analogous to u)} \tag{11.205}$$

11.9.7. 15-Node Wedges

These shape functions are for 15-node wedge elements such as SOLID90 that are based on a condensation of a 20-node brick element Equation 11.230 (p. 354). or are computed directly.

Figure 11.15: 15-Node Wedge Element



Elements in a wedge configuration use shape functions based on triangular coordinates and the r coordinate going from -1.0 to +1.0.

$$\begin{aligned} u = \frac{1}{2}(&u_I(L_1(2L_1-1)(1-r)-L_1(1-r^2))+u_J(L_2-1)(1-r)-L_2(1-r^2)) \\ &+u_K(L_3(2L_3-1)(1-r)-L_3(1-r^2))+u_M(L_1(2L_1-1)(1+r) \\ &-L_1(1-r^2))+u_N(L_2(2L_2-1))(1+r)-L_2(1-r^2)) \\ &+u_O(L_3(2L_3-1)(1+r)-L_3(1-r^2))+2(u_Q L_1L_2(1-r)) \\ &+u_R L_2L_3(1-r)+u_T L_3L_1(1-r)+u_U L_1L_2(1+r) \\ &+u_V L_2L_3(1+r)+u_X L_3L_1(1+r))+u_Y L_1(1-r^2) \\ &+u_Z L_2(1-r^2)+u_A L_3(1-r^2) \end{aligned} \tag{11.206}$$

$$v = \frac{1}{2}(v_I L_1(2L_1-1)\ldots \text{(analogous to u)} \tag{11.207}$$

$$w = \frac{1}{2}(w_I L_1(2L_1-1)\ldots \text{(analogous to u)} \tag{11.208}$$

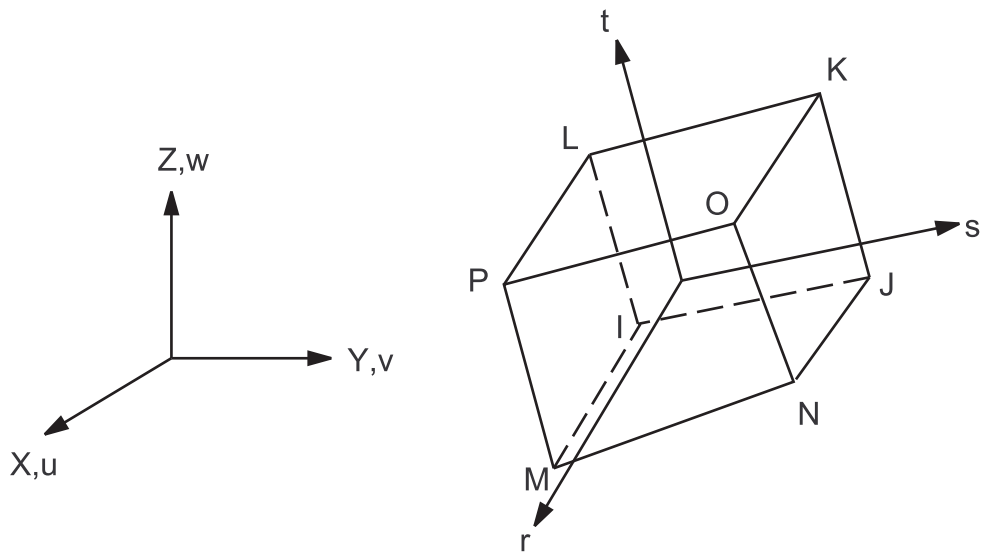
$$T = \frac{1}{2}(T_I L_1(2L_1-1)\ldots \text{(analogous to u)} \tag{11.209}$$

$$V = \frac{1}{2}(V_I L_1(2L_1-1)\ldots \text{(analogous to u)} \tag{11.210}$$

$$C = \frac{1}{2}(C_I L_1(2L_1\text{-}1)\ldots \text{(analogous to u)} \tag{11.211}$$

11.9.8. 8-Node Bricks without ESF

Figure 11.16: 8-Node Brick Element



These shape functions are for 8-node brick elements without extra shape functions such as SOLID5 with KEYOPT(3) = 1 or FLUID30:

$$u = \frac{1}{8}(u_I(1-s)(1-t)(1-r) + u_J(1+s)(1-t)(1-r) + u_K(1+s)(1+t)(1-r) + u_L(1-s)(1+t)(1-r) + u_M(1-s)(1-t)(1+r) + u_N(1+s)(1-t)(1+r) + u_O(1+s)(1+t)(1+r) + u_P(1-s)(1+t)(1+r)) \quad (11.212)$$

$$v = \frac{1}{8}(v_I(1-s)\ldots \text{(analogous to u)} \quad (11.213)$$

$$w = \frac{1}{8}(w_I(1-s)\ldots \text{(analogous to u)} \quad (11.214)$$

$$A_x = \frac{1}{8}(A_{xI}(1-s)\ldots \text{(analogous to u)} \quad (11.215)$$

$$A_y = \frac{1}{8}(A_{yI}(1-s)\ldots \text{(analogous to u)} \quad (11.216)$$

$$A_z = \frac{1}{8}(A_{zI}(1-s)\ldots \text{(analogous to u)} \quad (11.217)$$

$$V_x = \frac{1}{8}(V_{xI}(1-s)\ldots \text{(analogous to u)} \quad (11.218)$$

$$V_y = \frac{1}{8}(V_{yI}(1-s)\ldots \text{(analogous to u)} \quad (11.219)$$

$$V_z = \frac{1}{8}(V_{zI}(1-s)\ldots \text{(analogous to u)} \quad (11.220)$$

$$P = \frac{1}{8}(P_I(1-s)\ldots \text{(analogous to u)} \quad (11.221)$$

$$T = \frac{1}{8}(T_I(1-s)\ldots \text{(analogous to u)} \quad (11.222)$$

$$V = \frac{1}{8}(V_I(1-s)\ldots \text{(analogous to u)} \quad (11.223)$$

$$\phi = \frac{1}{8}(\phi_I(1-s)\ldots \text{(analogous to u)} \quad (11.224)$$

$$E^K = \frac{1}{8}(E_I^K(1-s)\ldots \text{(analogous to u)} \quad (11.225)$$

$$E^D = \frac{1}{8}(E_I^D(1-s)\ldots \text{(analogous to u)} \quad (11.226)$$

11.9.9. 8-Node Bricks with ESF

(Please see Figure 11.16: 8-Node Brick Element (p. 352)) These shape functions are for 8-node brick elements with extra shape functions such as SOLID5 with KEYOPT(3) = 0:

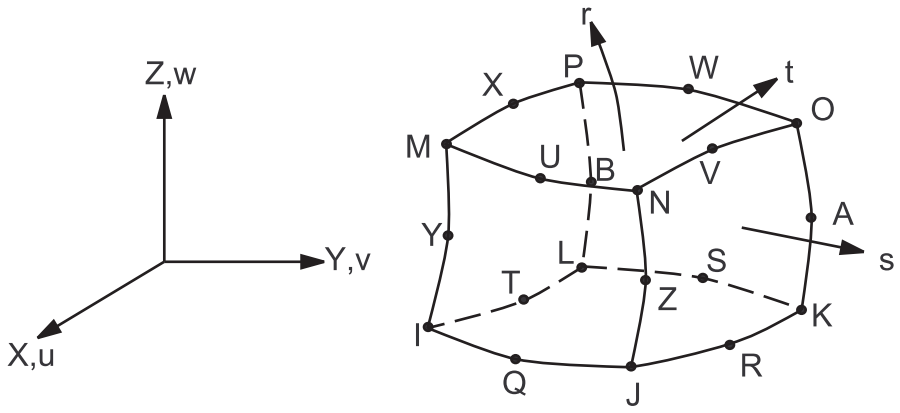
$$u = \frac{1}{8}(u_I(1-s)(1-t)(1-r) + u_J(1+s)(1-t)(1-r) + u_K(1+s)(1+t)(1-r) + u_L(1-s)(1+t)(1-r) + u_M(1-s)(1-t)(1+r) + u_N(1+s)(1-t)(1+r) + u_O(1+s)(1+t)(1+r) + u_P(1-s)(1+t)(1+r)) + u_1(1-s^2) + u_2(1-t^2) + u_3(1-r^2) \quad (11.227)$$

$$v = \frac{1}{8}(v_I(1-s)\ldots \text{(analogous to u)} \quad (11.228)$$

$$w = \frac{1}{8}(w_I(1-s)\ldots \text{(analogous to u)} \quad (11.229)$$

11.9.10. 20-Node Bricks

Figure 11.17: 20-Node Brick Element



These shape functions are used for 20-node solid elements such as SOLID90:

$$u = \frac{1}{8}(u_I(1-s)(1-t)(1-r)(-s-t-r-2) + u_J(1+s)(1-t)(1-r)(s-t-r-2) + u_K(1+s)(1+t)(1-r)(s+t-r-2) + u_L(1-s)(1+t)(1-r)(-s+t-r-2) + u_M(1-s)(1-t)(1+r)(-s-t+r-2) + u_N(1+s)(1-t)(1+r)(s-t+r-2) + u_O(1+s)(1+t)(1+r)(s+t+r-2) + u_P(1-s)(1+t)(1+r)(-s+t+r-2)) + \frac{1}{4}(u_Q(1-s^2)(1-t)(1-r) + u_R(1+s)(1-t^2)(1-r) + u_S(1-s^2)(1+t)(1-r) + u_T(1-s)(1-t^2)(1-r) + u_U(1-s^2)(1-t)(1+r) + u_V(1+s)(1-t^2)(1+r) + u_W(1-s^2)(1+t)(1+r) + u_X(1-s)(1-t^2)(1+r) + u_Y(1-s)(1-t)(1-r^2) + u_Z(1+s)(1-t)(1-r^2) + u_A(1+s)(1+t)(1-r^2) + u_B(1-s)(1+t)(1-r^2)) \quad (11.230)$$

$$v = \frac{1}{8}(v_I(1-s)\ldots \text{(analogous to u)} \quad (11.231)$$

$$w = \frac{1}{8}(w_I(1-s)\ldots \text{(analogous to u)} \tag{11.232}$$

$$T = \frac{1}{8}(T_I(1-s)\ldots \text{(analogous to u)} \tag{11.233}$$

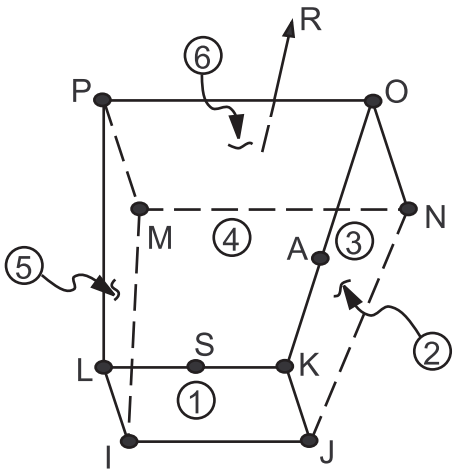
$$V = \frac{1}{8}(V_I(1-s)\ldots \text{(analogous to u)} \tag{11.234}$$

$$\phi = \frac{1}{8}(\phi_I(1-s)\ldots \text{(analogous to u)} \tag{11.235}$$

$$C = \frac{1}{8}(C_I(1\text{-}s)\ldots \text{(analogous to u)} \tag{11.236}$$

11.9.11. 8-Node Infinite Bricks

Figure 11.18: 3-D 8-Node Brick Element



These Lagrangian isoparametric shape functions and "mapping" functions are for the 3-D 8-node solid brick infinite elements such as INFIN111:

11.9.11.1. Lagrangian Isoparametric Shape Functions

$$\begin{aligned} A_x = {} & \frac{1}{8}(A_{xI}(1-s)(1-t)(r^2-r) \\ & + A_{xJ}(1+s)(1-t)(r^2-r) \\ & + A_{xK}(1+s)(1+t)(r^2-r) \\ & + A_{xL}(1-s)(1+t)(r^2-r)) \\ & + \frac{1}{4}(A_{xM}(1-s)(1-t)(1-r^2) \\ & + A_{xN}(1+s)(1-t)(1-r^2) \\ & + A_{xO}(1+s)(1+t)(1-r^2) \\ & + A_{xP}(1-s)(1+t)(1-r^2)) \end{aligned} \tag{11.237}$$

$$A_y = \frac{1}{8}(A_{xl}(1 - s) \ldots \text{(analogous to } A_x) \tag{11.238}$$

$$A_z = \frac{1}{8}(A_{zl}(1 - s) \ldots \text{(analogous to } A_x) \tag{11.239}$$

$$T = \frac{1}{8}(A_{Tl}(1 - s) \ldots \text{(analogous to } A_x) \tag{11.240}$$

$$V = \frac{1}{8}(A_{Vl}(1 - s) \ldots \text{(analogous to } A_x) \tag{11.241}$$

$$\phi = \frac{1}{8}(\phi_l(1 - s) \ldots \text{(analogous to } A_x) \tag{11.242}$$

11.9.11.2. Mapping Functions

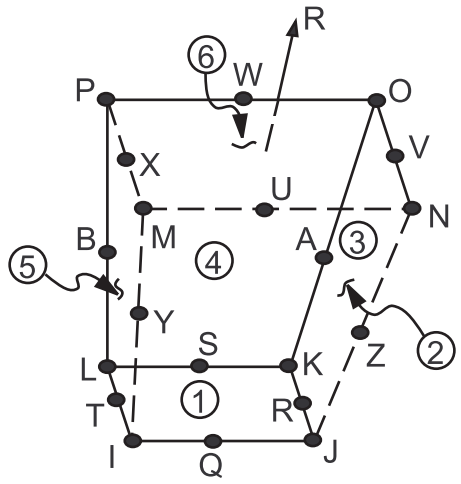
$$\begin{aligned} x = {} & \frac{1}{2}(x_I(1-s)(1-t)(-r)/(1-r) \\ & + x_J(1+s)(1-t)(-r)/(1-r) \\ & + x_K(1+s)(1+t)(-r)/(1-r) \\ & + x_L(1-s)(1+t)(-r)/(1-r)) \\ & + \frac{1}{4}(x_M(1-s)(1-t)(1+r)/(1-r) \\ & + x_N(1+s)(1-t)(1+r)/(1-r) \\ & + x_O(1+s)(1+t)(1+r)/(1-r) \\ & + x_P(1-s)(1+t)(1+r)/(1-r)) \end{aligned} \tag{11.243}$$

$$y = \frac{1}{2}(y_I(1-s) \ldots \text{(analogous to x)} \tag{11.244}$$

$$z = \frac{1}{2}(z_I(1-s) \ldots \text{(analogous to x)} \tag{11.245}$$

11.9.12. 3-D 20-Node Infinite Bricks

Figure 11.19: 20-Node Solid Brick Infinite Element



These Lagrangian isoparametric shape functions and "mapping" functions are for the 3-D 20-node solid brick infinite elements such as INFIN111:

11.9.12.1. Lagrangian Isoparametric Shape Functions

$$
\begin{aligned}
A_x = {} & \frac{1}{8}(A_{xI}(1-s)(1-t)(1-r)(-s-t-r-2)) \\
& + \frac{1}{4}(A_{xJ}(1-s^2)(1-t)(1-r)) \\
& + \frac{1}{8}(A_{xK}(1+s)(1-t)(1-r)(s-t-r-2)) \\
& + \frac{1}{4}(A_{xL}(1+s)(1-t^2)(1-r)) \\
& + \frac{1}{8}(A_{xM}(1+s)(1+t)(1-r)(s+t-r-2)) \\
& + \frac{1}{4}(A_{xN}(1-s^2)(1+t)(1-r)) \\
& + \frac{1}{8}(A_{xO}(1-s)(1+t)(1-r)(-s+t-r-2)) \\
& + \frac{1}{4}(A_{xP}(1-s)(1-t^2)(1-r)) \\
& + \frac{1}{4}(A_{xQ}(1-s)(1-t)(1-r^2)) \\
& + \frac{1}{4}(A_{xR}(1+s)(1-t)(1-r^2)) \\
& + \frac{1}{4}(A_{xS}(1+s)(1+t)(1-r^2)) \\
& + \frac{1}{4}(A_{xT}(1-s)(1+t)(1-r^2))
\end{aligned}
\tag{11.246}
$$

$$A_y = \frac{1}{8}(A_{yI}(1 - s). . . \text{(analogous to } A_x) \quad (11.247)$$

$$A_z = \frac{1}{8}(A_{zI}(1 - s). . . \text{(analogous to } A_x) \quad (11.248)$$

$$T = \frac{1}{8}(A_{TI}(1 - s). . . \text{(analogous to } A_x) \quad (11.249)$$

$$V = \frac{1}{8}(A_{VI}(1 - s). . . \text{(analogous to } A_x) \quad (11.250)$$

$$\phi = \frac{1}{8}(\phi_I(1 - s). . . \text{(analogous to } A_x) \quad (11.251)$$

11.9.12.2. Mapping Functions

$$\begin{aligned} x = {} & x_I(1-s)(1-t)(-s-t-r-2)/(2(1-r)) \\ & + x_J(1-s^2)(1-t)/(1-r) \\ & + x_K(1+s)(1-t)(-s-t-r-2)/(2(1-r)) \\ & + x_L(1+s)(1-t^2)/(1-r) \\ & + x_M(1+s)(1+t)(s+t-r-2)/(2(1-r)) \\ & + x_N(1-s^2)(1+t)/(1-r) \\ & + x_O(1-s)(1+t)(-s+t-r-2)/(2(1-r)) \\ & + x_P(1-s)(1-t^2)/(1-r) \\ & + x_Q(1-s)(1-t)(1+r)/(4(1-r)) \\ & + x_R(1+s)(1-t)(1+r)/(4(1-r)) \\ & + x_S(1+s)(1+t)(1+r)/(4(1-r)) \\ & + x_T(1-s)(1+t)(1+r)/(4(1-r)) \end{aligned} \quad (11.252)$$

$$y = y_I(1-s) \text{ . . . (analogous to x)} \quad (11.253)$$

$$z = z_I(1-s). . . \text{(analogous to x)} \quad (11.254)$$

The shape and mapping functions for the nodes U, V, W, X, Y, Z, A, and B are deliberately set to zero.

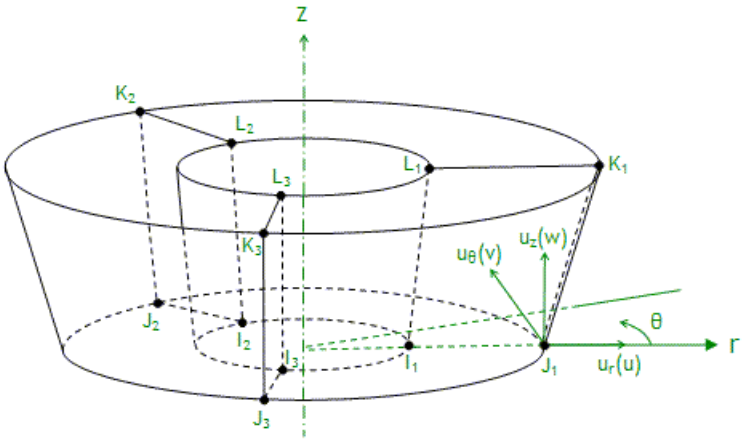
11.9.13. General Axisymmetric Solids

This section contains shape functions for general axisymmetric solid elements. These elements are available in a number of configurations, including certain combinations of the following features:

- A quadrilateral, or a degenerated triangle shape to simulate an irregular area, on the master plane (the plane on which the quadrilaterals or triangles are defined)
- With or without midside nodes
- A varying number of node planes (N_{np}) in the circumferential direction (defined via KEYOPT(2))

The elemental coordinates are cylindrical coordinates and displacements are defined and interpolated in that coordinate system, as shown in Figure 11.20: General Axisymmetric Solid Elements (when N_{np} = 3) (p. 359).

Figure 11.20: General Axisymmetric Solid Elements (when N_{hp} = 3)



When N_{np} is an odd number, the interpolation function used for displacement is:

$$u_i = h_i(s,t)(c_i + \sum_{m=1}^{\frac{NP-1}{2}} (a_i^m \cos m\theta + b_i^m \sin m\theta)) \tag{11.255}$$

where:

$i = r, \theta, z$

h_i (s, t) = regular Lagrangian polynominal interpolation functions like Equation 11.120 (p. 341) or Equation 11.134 (p. 342).

c_i, a_i^m, b_i^m = coefficients for the Fourier terms.

When N_{np} is an even number, the interpolation function is:

$$u_i = h_i(s,t)(c_i + \sum_{m=1}^{\frac{NP-2}{2}} (a_i^m \cos m\theta + b_i^m \sin m\theta) + a_i^{\frac{NP}{2}} (\cos \frac{NP}{2}\theta + \sin \frac{NP}{2}\theta)) \tag{11.256}$$

The temperatures are interpolated by Lagrangian polynominal interpolations in s, t plane, and linearly interpolated with θ in circumferential (θ) direction as:

$$T = T_n(s,t) + \left(\frac{\theta - \theta_n}{\theta_{n+1} - \theta_n}\right)(T_{n+1}(s,t) - T_n(s,t)) \tag{11.257}$$

where:

$\theta_n \le \theta \le \theta_{n+1}$

$n \le NP$ = node plane number in circumferential direction

T_n = same as Equation 11.128 (p. 341) and Equation 11.138 (p. 342).

11.9.13.1. General Axisymmetric Solid with 4 Base Nodes

All of the coefficients in Equation 11.255 (p. 359) and Equation 11.256 (p. 359) can be expressed by node displacements. Using $u_r = u$, $u_j = v$, $u_z = w$, and take $N_{np} = 3$ as an example.

$$u = \left(\frac{1}{4}(u_{I_1}(1-s)(1-t) + u_{J_1}(1+s)(1-t) + u_{K_1}(1+s)(1+t) + u_{L_1}(1-s)(1+t))\right)\left(\frac{1}{3} + \frac{2}{3}\cos\theta\right)$$
$$+\left(\frac{1}{4}(u_{I_2}(1-s)(1-t) + u_{J_2}(1+s)(1-t) + u_{K_2}(1+s)(1+t) + u_{L_2}(1-s)(1+t))\right)\left(\frac{1}{3} - \frac{1}{3}\cos\theta + \frac{1}{\sqrt{3}}\sin\theta\right) \quad (11.258)$$
$$+\left(\frac{1}{4}(u_{I_3}(1-s)(1-t) + u_{J_3}(1+s)(1-t) + u_{K_3}(1+s)(1+t) + u_{L_3}(1-s)(1+t))\right)\left(\frac{1}{3} - \frac{1}{3}\cos\theta - \frac{1}{\sqrt{3}}\sin\theta\right)$$

$$v = \left(\frac{1}{4}(v_{I_1}(1-s) \ldots \text{(analogous to u)} \quad (11.259)$$

$$w = \left(\frac{1}{4}(w_{I_1}(1-s) \ldots \text{(analogous to u)} \quad (11.260)$$

11.9.13.2. General Axisymmetric Solid with 3 Base Nodes

$$u = \left(u_{I_1}L_1 + u_{J_1}L_2 + u_{K_1}L_3\right)\left(\frac{1}{3} + \frac{2}{3}\cos\theta\right)$$
$$+\left(u_{I_2}L_1 + u_{J_2}L_2 + u_{K_2}L_3\right)\left(\frac{1}{3} - \frac{1}{3}\cos\theta + \frac{1}{\sqrt{3}}\cos\theta\right) \quad (11.261)$$
$$+\left(u_{I_3}L_1 + u_{J_3}L_2 + u_{K_3}L_3\right)\left(\frac{1}{3} - \frac{1}{3}\cos\theta - \frac{1}{\sqrt{3}}\sin\theta\right)$$

$$v = \left(v_{I_1}L_1 \ldots \text{(analogous to u)} \quad (11.262)$$

$$w = \left(w_{I_1}L_1 \ldots \text{(analogous to u)} \quad (11.263)$$

11.9.13.3. General Axisymmetric Solid with 8 Base Nodes

Similar to the element with 4 base nodes, the u, v, and w are expressed as:

$$u = \left(\begin{array}{l} \frac{1}{4}(u_{I_1}(1-s)(1-t)(-s-t-1) + u_{J_1}(1+s)(1-t)(s-t-1) \\ \quad + u_{K_1}(1+s)(1+t)(s+t-1) + u_{L_1}(1-s)(1+t)(-s+t-1)) \\ \quad + \frac{1}{2}(u_{M_1}(1-s^2)(1-t) + u_{N_1}(1+s)(1-t^2) \\ \quad + u_{O_1}(1-s^2)(1+t) + u_{P_1}(1-s)(1-t^2)) \end{array} \right) \left(\frac{1}{3} + \frac{2}{3}\cos\theta \right)$$
$$+ \left(\begin{array}{l} \frac{1}{4}(u_{I_2}(1-s)(1-t)(-s-t-1) + u_{J_2}(1+s)(1-t)(s-t-1) \\ \quad + u_{K_2}(1+s)(1+t)(s+t-1) + u_{L_2}(1-s)(1+t)(-s+t-1)) \\ \quad + \frac{1}{2}(u_{M_2}(1-s^2)(1-t) + u_{N_2}(1+s)(1-t^2) \\ \quad + u_{O_2}(1-s^2)(1+t) + u_{P_2}(1-s)(1-t^2)) \end{array} \right) \left(\frac{1}{3} - \frac{1}{3}\cos\theta + \frac{1}{\sqrt{3}}\sin\theta \right) \quad (11.264)$$
$$+ \left(\begin{array}{l} \frac{1}{4}(u_{I_3}(1-s)(1-t)(-s-t-1) + u_{J_3}(1+s)(1-t)(s-t-1) \\ \quad + u_{K_3}(1+s)(1+t)(s+t-1) + u_{L_3}(1-s)(1+t)(-s+t-1)) \\ \quad + \frac{1}{2}(u_{M_3}(1-s^2)(1-t) + u_{N_3}(1+s)(1-t^2) \\ \quad + u_{O_3}(1-s^2)(1+t) + u_{P_3}(1-s)(1-t^2)) \end{array} \right) \left(\frac{1}{3} - \frac{1}{3}\cos\theta - \frac{1}{\sqrt{3}}\sin\theta \right)$$

$$v = (\frac{1}{4}(v_{I}(1-s) \ldots \text{(analogous to u)} \quad (11.265)$$

$$w = (\frac{1}{4}(w_{I}(1-s) \ldots \text{(analogous to w)} \quad (11.266)$$

11.9.13.4. General Axisymmetric Solid with 6 Base Nodes

$$u = \left(\begin{array}{l} u_{I_1}(2L_1-1)L_1 + u_{J_1}(2L_2-1)L_2 + u_{K_1}(2L_3-1) \\ \quad + u_{L_1}(4L_1L_2) + u_{M_1}(4L_2L_3) + u_{N_1}(4L_3L_1) \end{array} \right) \left(\frac{1}{3} + \frac{2}{3}\cos\theta \right)$$
$$+ \left(\begin{array}{l} u_{I_2}(2L_1-1)L_1 + u_{J_2}(2L_2-1)L_2 + u_{K_2}(2L_3-1) \\ \quad + u_{L_2}(4L_1L_2) + u_{M_2}(4L_2L_3) + u_{N_2}(4L_3L_1) \end{array} \right) \left(\frac{1}{3} - \frac{1}{3}\cos\theta + \frac{1}{\sqrt{3}}\sin\theta \right) \quad (11.267)$$
$$+ \left(\begin{array}{l} u_{I_3}(2L_1-1)L_1 + u_{J_3}(2L_2-1)L_2 + u_{K_3}(2L_3-1) \\ \quad + u_{L_3}(4L_1L_2) + u_{M_3}(4L_2L_3) + u_{N_3}(4L_3L_1) \end{array} \right) \left(\frac{1}{3} - \frac{1}{3}\cos\theta - \frac{1}{\sqrt{3}}\sin\theta \right)$$

$$v = (v_{I_1}(2L_1-1)L_1 + \ \ldots \text{(analogous to u)} \quad (11.268)$$

$$w = (w_{I_1}(2L_1-1)L_1 + \ \ldots \text{(analogous to u)} \quad (11.269)$$

11.10. Electromagnetic Tangential Vector Elements

In electromagnetics, we encounter serious problems when node-based elements are used to represent vector electric or magnetic fields. (See Electric Scalar Potential (p. 190).) Node-based elements require

special treatment for enforcing boundary conditions of electromagnetic field at material interfaces, conducting surfaces and geometric corners. Tangentially continuous vector elements or edge elements, whose degrees of freedom are associated with the edges of the finite element mesh, have been shown to be free of such shortcomings. ([413] (p. 944))

11.10.1. Tetrahedral Elements

The tetrahedral element is the simplest tessellated shape and is able to model arbitrary 3-D geometric structures. It is also well suited for automatic mesh generation. The tetrahedral element, by far, is the most popular element shape for 3-D applications in FEA.

For the 1st-order tetrahedral element (KEYOPT(1) = 1), the degrees of freedom (DOF) are at the edges of element i.e., (DOFs = 6) (Figure 11.21: 1st-Order Tetrahedral Element (p. 362)). In terms of volume coordinates, the vector basis functions are defined as:

$$\vec{W}_{IJ} = h_{IJ}(\lambda_I \nabla\lambda_J - \lambda_J \nabla\lambda_I) \tag{11.270}$$

$$\vec{W}_{JK} = h_{JK}(\lambda_J \nabla\lambda_K - \lambda_K \nabla\lambda_J) \tag{11.271}$$

$$\vec{W}_{KI} = h_{KI}(\lambda_K \nabla\lambda_I - \lambda_I \nabla\lambda_K) \tag{11.272}$$

$$\vec{W}_{IL} = h_{IL}(\lambda_I \nabla\lambda_L - \lambda_L \nabla\lambda_I) \tag{11.273}$$

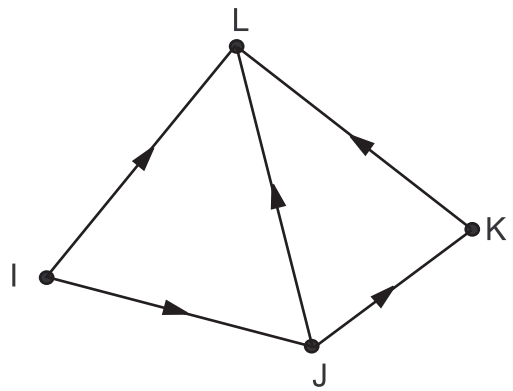
$$\vec{W}_{JL} = h_{JL}(\lambda_J \nabla\lambda_L - \lambda_L \nabla\lambda_J) \tag{11.274}$$

$$\vec{W}_{KL} = h_{KL}(\lambda_K \nabla\lambda_L - \lambda_L \nabla\lambda_K) \tag{11.275}$$

where:

h_{IJ} = edge length between node I and J
$\lambda_I, \lambda_J, \lambda_K, \lambda_L$ = volume coordinates ($\lambda_K = 1 - \lambda_I - \lambda_J - \lambda_L$)
$\nabla\lambda_I, \nabla\lambda_J, \nabla\lambda_K, \nabla\lambda_L$ = the gradient of volume coordinates

Figure 11.21: 1st-Order Tetrahedral Element



The tangential component of the approximated field is constant along the edge. The normal component of field varies linearly.

11.10.2. Hexahedral Elements

Tangential vector bases for hexahedral elements can be derived by carrying out the transformation mapping a hexahedral element in the global xyz coordinate to a brick element in local str coordinates.

For the 1st-order brick element (KEYOPT(1) = 1), the degrees of freedom (DOF) are at the edges of element (DOFs = 12) (Figure 11.22: 1st-Order Brick Element (p. 363)). The vector basis functions are cast in the local coordinate

$$\vec{W}_s^e = \frac{h_s}{8}(1 \pm t)(1 \pm r)\nabla s \qquad \text{parallel to s-axis} \qquad (11.276)$$

$$\vec{W}_t^e = \frac{h_t}{8}(1 \pm r)(1 \pm s)\nabla t \qquad \text{parallel to t-axis} \qquad (11.277)$$

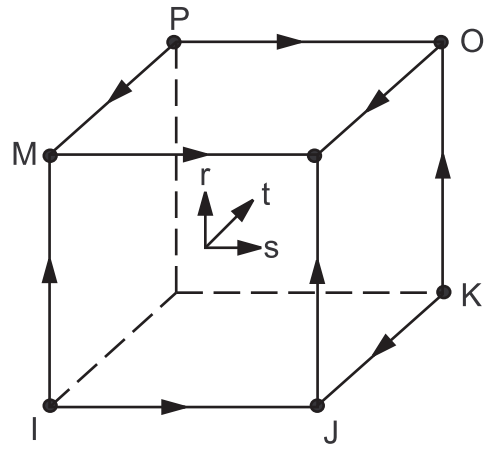
$$\vec{W}_r^e = \frac{h_r}{8}(1 \pm s)(1 \pm t)\nabla r \qquad \text{parallel to r-axis} \qquad (11.278)$$

where:

h_s, h_t, h_r = length of element edge

∇s, ∇t, ∇r = gradient of local coordinates

Figure 11.22: 1st-Order Brick Element



Chapter 12: Element Tools

The following element tools are available:

12.1. Element Shape Testing

12.1.1. Overview

All continuum elements (2-D and 3-D solids, 3-D shells) are tested for acceptable shape as they are defined by the **E**, **EGEN**, **AMESH**, **VMESH**, or similar commands. This testing, described in the following sections, is performed by computing shape parameters (such as Jacobian ratio) which are functions of geometry, then comparing them to element shape limits whose default values are functions of element type and settings (but can be modified by the user on the **SHPP** command with *Lab* = MODIFY as described below). When defined, an element may generate no warnings, one or more warnings, or the element may be rejected with an error.

12.1.2. 3-D Solid Element Faces and Cross-Sections

Some shape testing of 3-D solid elements (bricks [hexahedra], wedges, pyramids, and tetrahedra) is performed indirectly. Aspect ratio, parallel deviation, and maximum corner angle are computed for 3-D solid elements using the following steps:

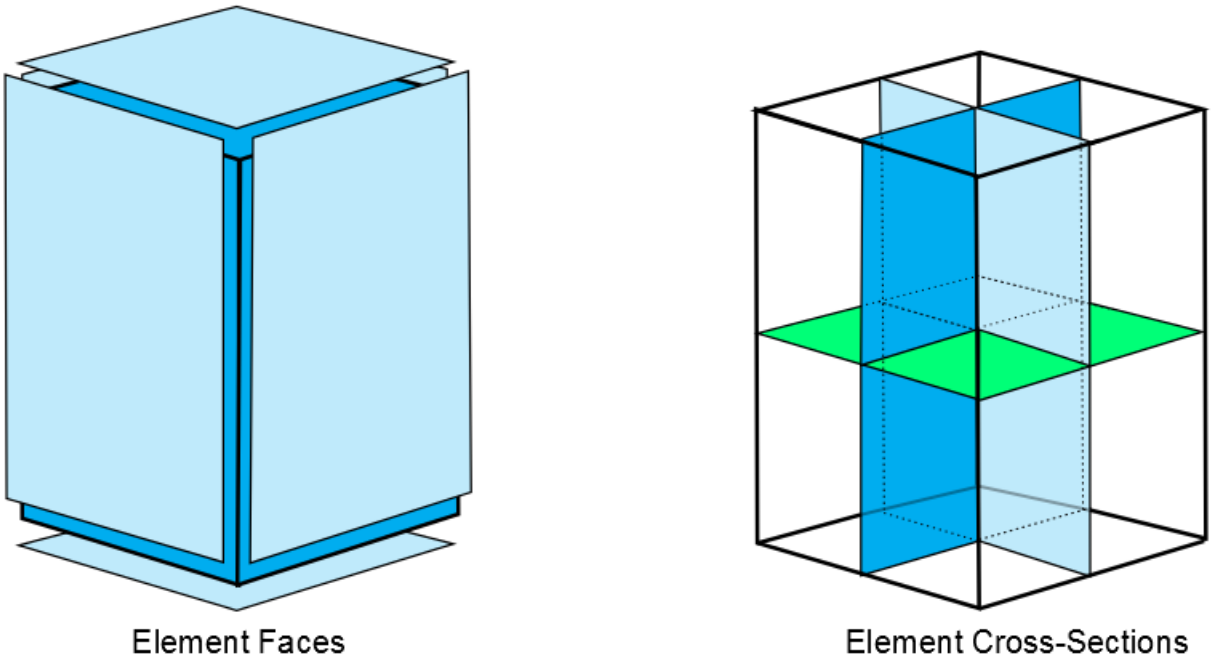
1. Each of these 3 quantities is computed, as applicable, for each face of the element as though it were a quadrilateral or triangle in 3-D space, by the methods described in sections Aspect Ratio (p. 368), Parallel Deviation (p. 371), and Maximum Corner Angle (p. 372).

2. Because some types of 3-D solid element distortion are not revealed by examination of the faces, cross-sections through the solid are constructed. Then, each of the 3 quantities is computed, as applicable, for each cross-section as though it were a quadrilateral or triangle in 3-D space.

3. The metric for the element is assigned as the worst value computed for any face or cross-section.

A brick element has 6 quadrilateral faces and 3 quadrilateral cross-sections (Figure 12.1: Brick Element (p. 366)). The cross-sections are connected to midside nodes, or to edge midpoints where midside nodes are not defined.

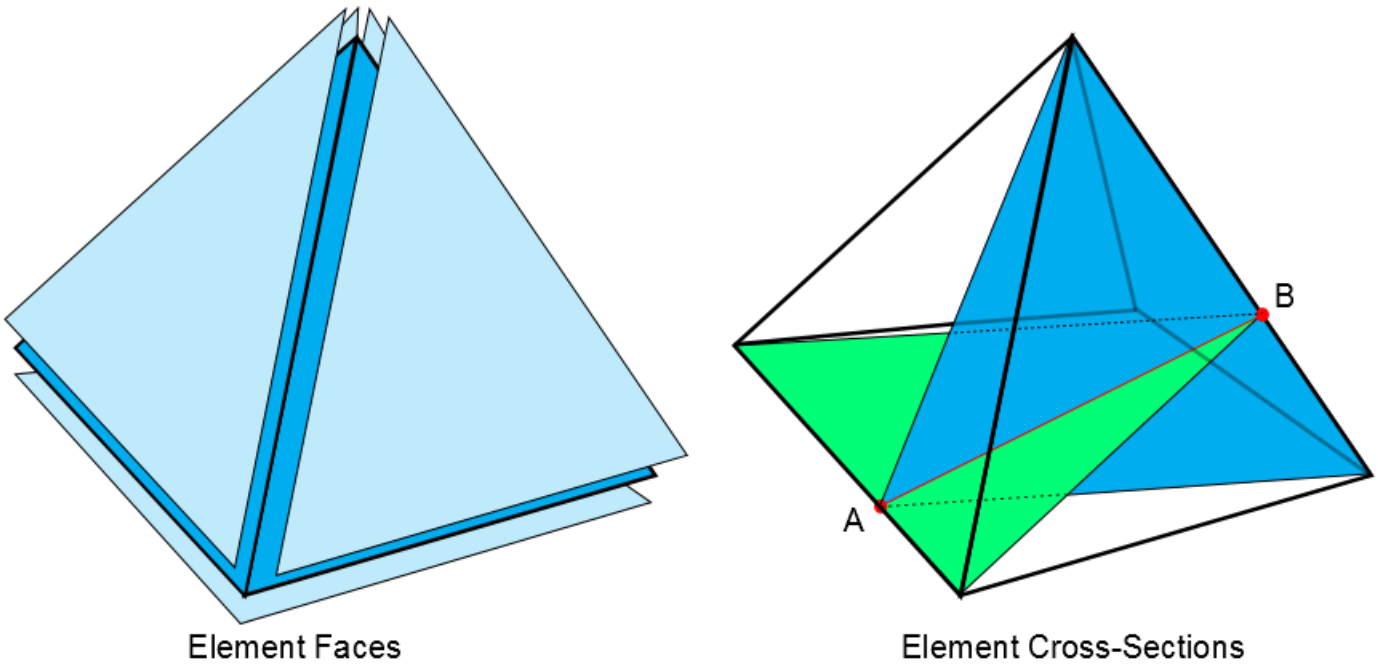
Figure 12.1: Brick Element



A pyramid element has 1 quadrilateral face and 4 triangle faces, as shown on the left side of Figure 12.2: Pyramid Element (p. 366). The pyramid element has 8 triangle cross-sections, constructed in 4 pairs. One of the 4 pairs is shown on the right side of Figure 12.2: Pyramid Element (p. 366).

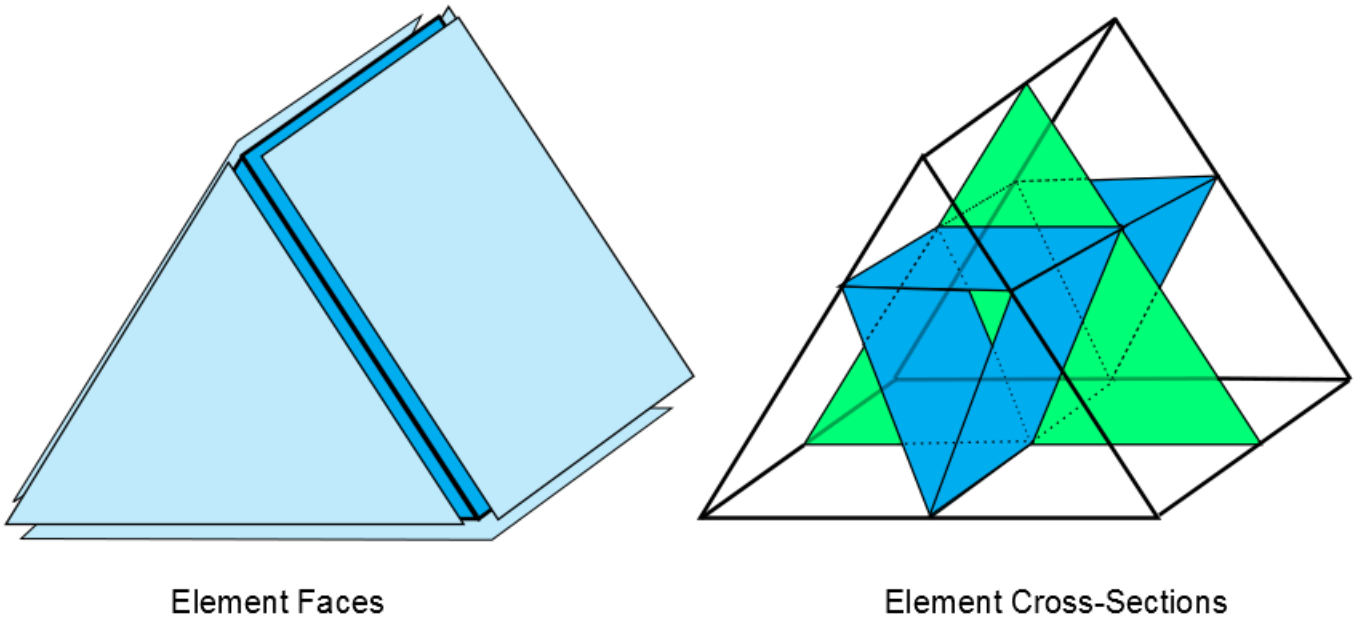
For each pair, the closest points (A and B in the figure) between the lines containing a quadrilateral face edge and the right-ward opposite leg are found. The paired cross-sections are the triangle containing the quadrilateral face edge and point B and the triangle containing the right-ward opposite leg and point A. The remaining 6 cross-sections are found in a similar manner using the other 3 quadrilateral face edges and their corresponding right-ward legs.

Figure 12.2: Pyramid Element



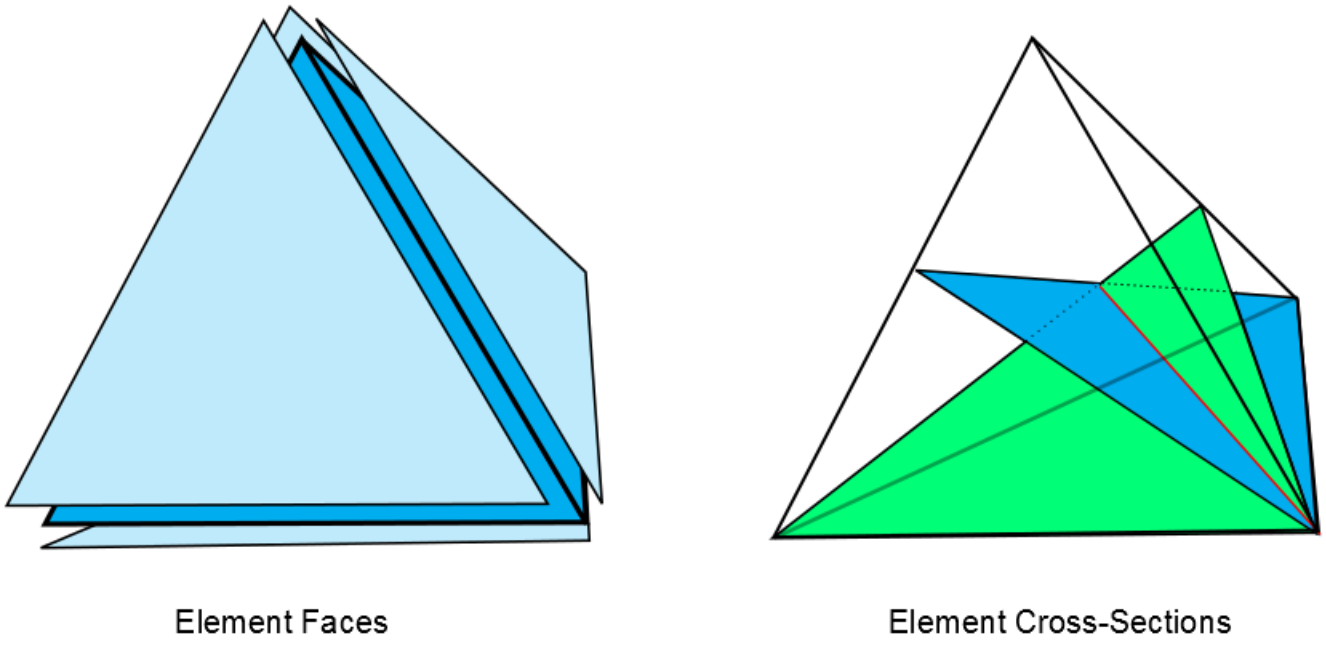
A wedge element has 3 quadrilateral and 2 triangle faces, and has 3 quadrilateral and 1 triangle cross-sections. As shown in Figure 12.3: Wedge Element (p. 367), the cross-sections are connected to midside nodes, or to edge midpoints where midside nodes are not defined.

Figure 12.3: Wedge Element



A tetrahedron element has 4 triangle faces and 6 triangle cross-sections (Figure 12.4: Tetrahedron Element (p. 367)). Each cross-section is the triangle containing one edge and the closest point on the line containing the opposite edge. Two of these cross-sections are shown on the right side of Figure 12.4: Tetrahedron Element (p. 367).

Figure 12.4: Tetrahedron Element



12.1.3. Aspect Ratio

An aspect ratio is computed and tested for all elements except Emag elements (see Table 12.1: Aspect Ratio Limits (p. 370)). Aspect ratio as a shape measure has been reported in finite element literature for decades (Robinson [121] (p. 927)) and is one of the easiest ones to understand. For triangles, equilateral triangles have an aspect ratio of 1 and the aspect ratio increases the more the triangle is stretched from an equilateral shape. For quadrilaterals, squares have an aspect ratio of 1 and the aspect ratio increases the more the square is stretched to an oblong shape. Some analysts want to be warned about high aspect ratio so they can verify that the creation of any stretched elements was intentional. Many other analysts routinely ignore it.

Unless elements are so stretched that numeric round off could become a factor (aspect ratio > 1000), aspect ratio alone has little correlation with analysis accuracy. Finite element meshes should be tailored to the physics of the given problem; i.e., fine in the direction of rapidly changing field gradients, relatively coarse in directions with less rapidly changing fields. Sometimes this calls for elements having aspect ratios of 10, 100, or in extreme cases 1000. (Examples include shell or thin coating analyses using solid elements, thermal shock "skin" stress analyses, and fluid boundary layer analyses.) Attempts to artificially restrict aspect ratio could compromise analysis quality in some cases.

12.1.4. Aspect Ratio Calculation for Triangles

For the benefit of computational efficiency, the aspect ratio of a triangle is computed based on the angle measures at the corner nodes rather than the height and width of the triangle. For the calculation, midside nodes are ignored and the triangle is treated as connecting nodes I, J, and K with straight lines:

1. For the angle θ, at node I, a quantity AM_I is calculated as follows:

   $$\text{for } 0^\circ < \theta \le 60^\circ, AM_I = \frac{0.5}{1-\cos\theta}$$

   $$\text{for } 60^\circ < \theta < 180^\circ, AM_I = \frac{1.5}{1+\cos\theta}$$

   The quantity AM_I is 1 when θ = 60°. As θ deviates from 60° in either direction, AM_I increases (to a maximum of ∞ as θ nears 0° or 180°).

2. The quantity AM is calculated for each of the remaining nodes J and K.

3. A preliminary aspect ratio, ASPRAT, is calculated by averaging the three values of AM, then taking the square root:

   $$ASPRAT = \sqrt{\frac{AM_I + AM_J + AM_K}{3}}$$

4. Finally, the aspect ratio is modified to produce a distribution of values similar to those that would be produced if the aspect ratio were calculated directly from the height and width:

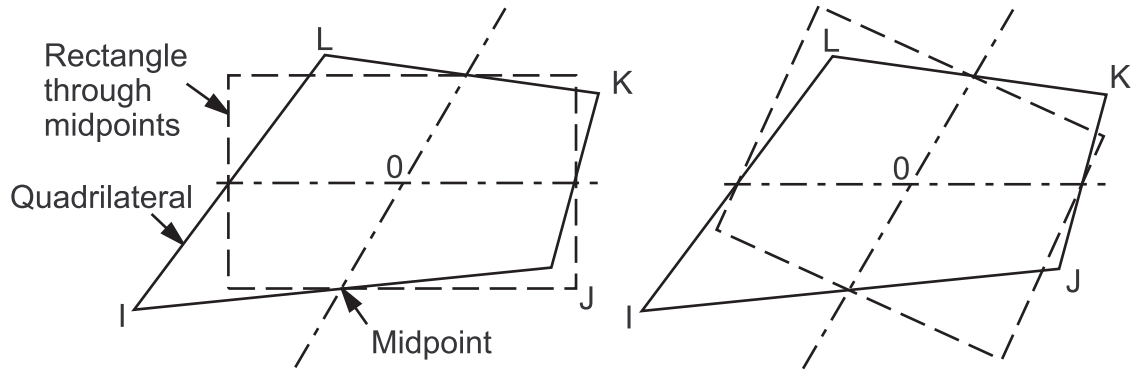
   $$ASPRAT = (ASPRAT - 1) * 2 + 1$$

The best possible triangle aspect ratio, for an equilateral triangle, is 1. Larger aspect ratios are calculated as the triangle is stretched. Triangles having aspect ratios of 1 and 20 are shown in Figure 12.5: Aspect Ratios for Triangles (p. 369).

Figure 12.5: Aspect Ratios for Triangles

12.1.5. Aspect Ratio Calculation for Quadrilaterals

Figure 12.6: Quadrilateral Aspect Ratio Calculation



The aspect ratio for a quadrilateral is computed by the following steps, using only the corner nodes of the element (Figure 12.6: Quadrilateral Aspect Ratio Calculation (p. 369)):

1. If the element is not flat, the nodes are projected onto a plane passing through the average of the corner locations and perpendicular to the average of the corner normals. The remaining steps are performed on these projected locations.

2. Two lines are constructed that bisect the opposing pairs of element edges and which meet at the element center. In general, these lines are not perpendicular to each other or to any of the element edges.

3. Rectangles are constructed centered about each of the 2 lines, with edges passing through the element edge midpoints. The aspect ratio of the quadrilateral is the ratio of a longer side to a shorter side of whichever rectangle is most stretched.

4. The best possible quadrilateral aspect ratio, for a square, is one. A quadrilateral having an aspect ratio of 20 is shown in Figure 12.7: Aspect Ratios for Quadrilaterals (p. 370).

Figure 12.7: Aspect Ratios for Quadrilaterals

Table 12.1: Aspect Ratio Limits

Command to modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,1	warning	20	Elements this stretched look to many users like they deserve warnings.	Disturbance of analysis results has not been proven It is difficult to avoid warnings even with a limit of 20.
SHPP,MODIFY,2	error	10^6	Informal testing has demonstrated solution error attributable to computer round off at aspect ratios of 1,000 to 100,000.	Threshold of round off problems depends on what computer is being used. Valid analyses should not be blocked.

12.1.6. Angle Deviation

Angle deviation from 90° corner angle is computed and tested only for the SHELL28 shear/twist panel quadrilateral (see Table 12.2: Angle Deviation Limits (p. 371)). It is an important measure because the element derivation assumes a rectangle.

12.1.7. Angle Deviation Calculation

The angle deviation is based on the angle between each pair of adjacent edges, computed using corner node positions in 3-D space. It is simply the largest deviation from 90° of any of the 4 corner angles of the element.

The best possible deviation is 0° (Figure 12.8: Angle Deviations for SHELL28 (p. 371)). Figure 12.8: Angle Deviations for SHELL28 (p. 371) also shows angle deviations of 5° and 30°, respectively.

Figure 12.8: Angle Deviations for SHELL28

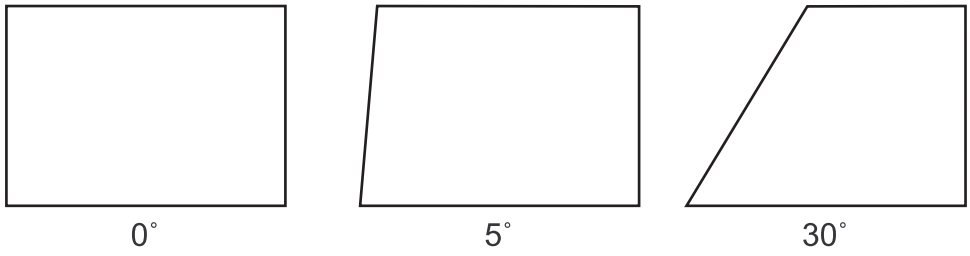


Table 12.2: Angle Deviation Limits

Command to Modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,7	warning	5°	Results degrade as the element deviates from a rectangular shape.	It is difficult to avoid warnings even with a limit of 5°
SHPP,MODIFY,8	error	30°	Pushing the limit further does not seem prudent.	Valid analyses should not be blocked.

12.1.8. Parallel Deviation

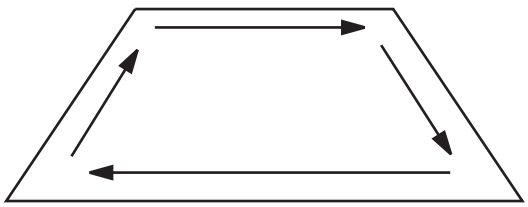
Parallel deviation is computed and tested for all quadrilaterals or 3-D solid elements having quadrilateral faces or cross-sections, except Emag elements (see Table 12.3: Parallel Deviation Limits (p. 372)). Formal testing has demonstrated degradation of stress convergence in linear displacement quadrilaterals as opposite edges become less parallel to each other.

12.1.9. Parallel Deviation Calculation

Parallel deviation is computed using the following steps:

1. Ignoring midside nodes, unit vectors are constructed in 3-D space along each element edge, adjusted for consistent direction, as demonstrated in Figure 12.9: Parallel Deviation Unit Vectors (p. 371).

   Figure 12.9: Parallel Deviation Unit Vectors

2. For each pair of opposite edges, the dot product of the unit vectors is computed, then the angle (in degrees) whose cosine is that dot product. The parallel deviation is the larger of these 2 angles. (In the illustration above, the dot product of the 2 horizontal unit vectors is 1, and acos (1) = 0°. The dot product

of the 2 vertical vectors is 0.342, and acos (0.342) = 70°. Therefore, this element's parallel deviation is 70°.)

3. The best possible deviation, for a flat rectangle, is 0°. Figure 12.10: Parallel Deviations for Quadrilaterals (p. 372) shows quadrilaterals having deviations of 0°, 70°, 100°, 150°, and 170°.

Figure 12.10: Parallel Deviations for Quadrilaterals

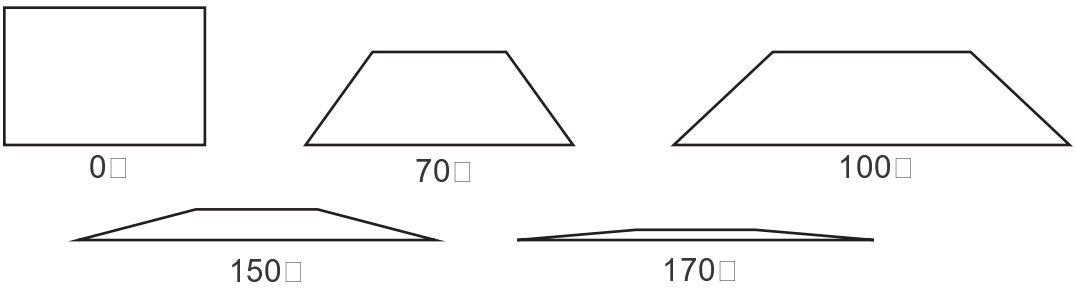


Table 12.3: Parallel Deviation Limits

Command to Modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,11	warning for elements without midside nodes	70°	Testing has shown results are degraded by this much distortion	It is difficult to avoid warnings even with a limit of 70°
SHPP,MODIFY,12	error for elements without midside nodes	150°	Pushing the limit further does not seem prudent	Valid analyses should not be blocked.
SHPP,MODIFY,13	warning for elements with midside nodes	100°	Elements having deviations > 100° look like they deserve warnings.	Disturbance of analysis results for quadratic elements has not been proven.
SHPP,MODIFY,14	error for elements with midside nodes	170°	Pushing the limit further does not seem prudent	Valid analyses should not be blocked.

12.1.10. Maximum Corner Angle

Maximum corner angle is computed and tested for all except Emag elements (see Table 12.4: Maximum Corner Angle Limits (p. 373)). Some in the finite element community have reported that large angles (approaching 180°) degrade element performance, while small angles don't.

12.1.11. Maximum Corner Angle Calculation

The maximum angle between adjacent edges is computed using corner node positions in 3-D space. (Midside nodes, if any, are ignored.) The best possible triangle maximum angle, for an equilateral triangle, is 60°. Figure 12.11: Maximum Corner Angles for Triangles (p. 373) shows a triangle having a maximum

corner angle of 165°. The best possible quadrilateral maximum angle, for a flat rectangle, is 90°. Figure 12.12: Maximum Corner Angles for Quadrilaterals (p. 373) shows quadrilaterals having maximum corner angles of 90°, 140° and 180°.

Figure 12.11: Maximum Corner Angles for Triangles

Figure 12.12: Maximum Corner Angles for Quadrilaterals

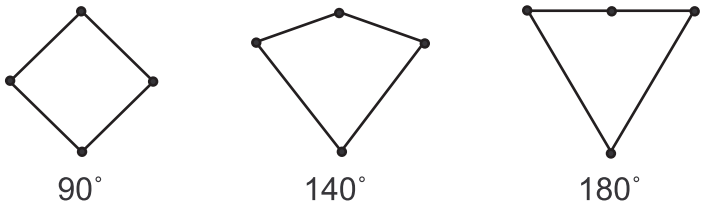


Table 12.4: Maximum Corner Angle Limits

Command to Modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,15	warnings for triangles	165°	Any element this distorted looks like it deserves a warning.	Disturbance of analysis results has not been proven. It is difficult to avoid warnings even with a limit of 165°.
SHPP,MODIFY,16	error for triangles	179.9°	We can not allow 180°	Valid analyses should not be blocked.
SHPP,MODIFY,17	warning for quadrilaterals without midside nodes	155°	Any element this distorted looks like it deserves a warning.	Disturbance of analysis results has not been proven. It is difficult to avoid warnings even with a limit of 155°.

Command to Modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,18	error for quadrilaterals without midside nodes	179.9°	We can not allow 180°	Valid analyses should not be blocked.
SHPP,MODIFY,19	warning for quadrilaterals with midside nodes	165°	Any element this distorted looks like it deserves a warning.	Disturbance of analysis results has not been proven. It is difficult to avoid warnings even with a limit of 165°.
SHPP,MODIFY,20	error for quadrilaterals with midside nodes	179.9°	We can not allow 180°	Valid analyses should not be blocked.

12.1.12. Jacobian Ratio

Jacobian ratio is computed and tested for all elements except triangles and tetrahedra that (a) are linear (have no midside nodes) or (b) have perfectly centered midside nodes (see Table 12.5: Jacobian Ratio Limits (p. 376)). A high ratio indicates that the mapping between element space and real space is becoming computationally unreliable.

12.1.12.1. Jacobian Ratio Calculation

An element's Jacobian ratio is computed by the following steps, using the full set of nodes for the element:

1. At each sampling location listed in the table below, the determinant of the Jacobian matrix is computed and called R_J. R_J at a given point represents the magnitude of the mapping function between element natural coordinates and real space. In an ideally-shaped element, R_J is relatively constant over the element, and does not change sign.

Element Shape	R_J Sampling Locations
10-node tetrahedra - **SHPP**,LSTET,OFF	corner nodes
10-node tetrahedra - **SHPP**,LSTET,ON	integration points
5-node or 13-node pyramids	base corner nodes and near apex node (apex R_J factored so that a pyramid having all edges the same length will produce a Jacobian ratio of 1)
8-node quadrilaterals	corner nodes and centroid
20-node bricks	all nodes and centroid
all other elements	corner nodes

2. The Jacobian ratio of the element is the ratio of the maximum to the minimum sampled value of R_J. If the maximum and minimum have opposite signs, the Jacobian ratio is arbitrarily assigned to be -100 (and the element is clearly unacceptable).

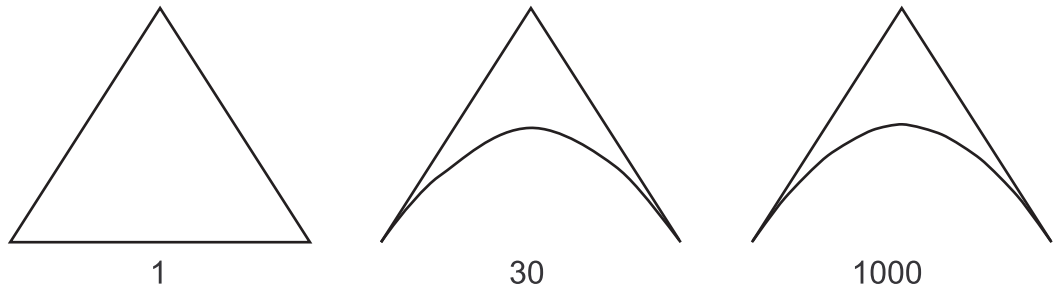
3. If the element is a midside-node tetrahedron, an additional R_J is computed for a fictitious straight-sided tetrahedron connected to the 4 corner nodes. If that R_J differs in sign from any nodal R_J (an extremely rare occurrence), the Jacobian ratio is arbitrarily assigned to be -100.

4. The sampling locations for midside-node tetrahedra depend upon the setting of the linear stress tetrahedra key on the **SHPP** command. The default behavior (**SHPP**,LSTET,OFF) is to sample at the corner nodes, while the optional behavior (**SHPP**,LSTET.ON) is to sample at the integration points (similar to what was done for the DesignSpace product). Sampling at the integration points will result in a lower Jacobian ratio than sampling at the nodes, but that ratio is compared to more restrictive default limits (see Table 12.5: Jacobian Ratio Limits (p. 376) below). Nevertheless, some elements which pass the LSTET,ON test fail the LSTET,OFF test - especially those having zero R_J at a corner node. Testing has shown that such elements have no negative effect on linear elastic stress accuracy. Their effect on other types of solutions has not been studied, which is why the more conservative test is recommended for general usage. Brick elements (such as SOLID186) degenerated into tetrahedra are tested in the same manner as are 'native' tetrahedra (SOLID187). In most cases, this produces conservative results. However, for SOLID185 and SOLID186 when using the non-recommended tetrahedron shape, it is possible that such a degenerate element may produce an error during solution, even though it produced no warnings during shape testing.

5. If the element is a line element having a midside node, the Jacobian matrix is not square (because the mapping is from one natural coordinate to 2-D or 3-D space) and has no determinant. For this case, a vector calculation is used to compute a number which behaves like a Jacobian ratio. This calculation has the effect of limiting the arc spanned by a single element to about 106°

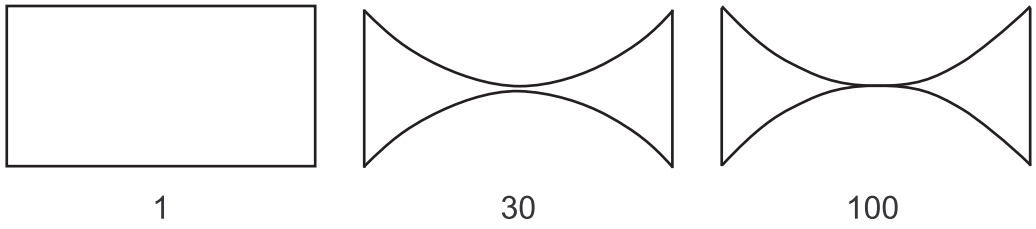
A triangle or tetrahedron has a Jacobian ratio of 1 if each midside node, if any, is positioned at the average of the corresponding corner node locations. This is true no matter how otherwise distorted the element may be. Hence, this calculation is skipped entirely for such elements. Moving a midside node away from the edge midpoint position will increase the Jacobian ratio. Eventually, even very slight further movement will break the element (Figure 12.13: Jacobian Ratios for Triangles (p. 375)). We describe this as “breaking” the element because it suddenly changes from acceptable to unacceptable- “broken”.

Figure 12.13: Jacobian Ratios for Triangles



Any rectangle or rectangular parallelepiped having no midside nodes, or having midside nodes at the midpoints of its edges, has a Jacobian ratio of 1. Moving midside nodes toward or away from each other can increase the Jacobian ratio. Eventually, even very slight further movement will break the element (Figure 12.14: Jacobian Ratios for Quadrilaterals (p. 376)).

Figure 12.14: Jacobian Ratios for Quadrilaterals



A quadrilateral or brick has a Jacobian ratio of 1 if (a) its opposing faces are all parallel to each other, and (b) each midside node, if any, is positioned at the average of the corresponding corner node locations. As a corner node moves near the center, the Jacobian ratio climbs. Eventually, any further movement will break the element (Figure 12.15: Jacobian Ratios for Quadrilaterals (p. 376)).

Figure 12.15: Jacobian Ratios for Quadrilaterals

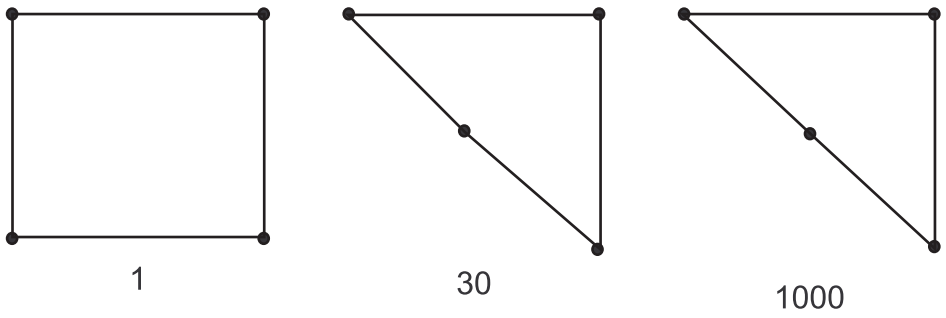


Table 12.5: Jacobian Ratio Limits

Command to modify	Type of limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,31	warning for h-elements	30 if **SHPP**, LSTET,OFF	A ratio this high indicates that the mapping between element and real space is becoming computationally unreliable.	Disturbance of analysis results has not been proven. It is difficult to avoid warnings even with a limit of 30.
		10 if **SHPP**, LSTET,ON		
SHPP,MODIFY,32	error for h-elements	1,000 if **SHPP**, LSTET,OFF	Pushing the limit further does not seem prudent.	Valid analyses should not be blocked.
		40 if **SHPP**, LSTET,ON		

12.1.13. Warping Factor

Warping factor is computed and tested for some quadrilateral shell elements, and the quadrilateral faces of bricks, wedges, and pyramids. (See Warping Factor (p. 376).) A high factor may indicate a condition the underlying element formulation cannot handle well, or may simply hint at a mesh generation flaw.

The following warping factor topics are available:

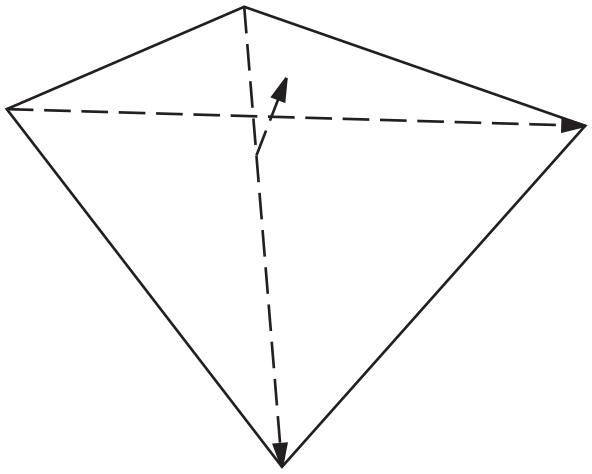
Also see Table 12.6: Warping Factor Limits (p. 380).

12.1.13.1. Warping Factor Calculation for Quadrilateral Shell Elements

A quadrilateral element's warping factor is computed from its corner node positions and other available data by the following steps:

1. An average element normal is computed as the vector (cross) product of the 2 diagonals (Figure 12.16: Shell Average Normal Calculation (p. 377)).

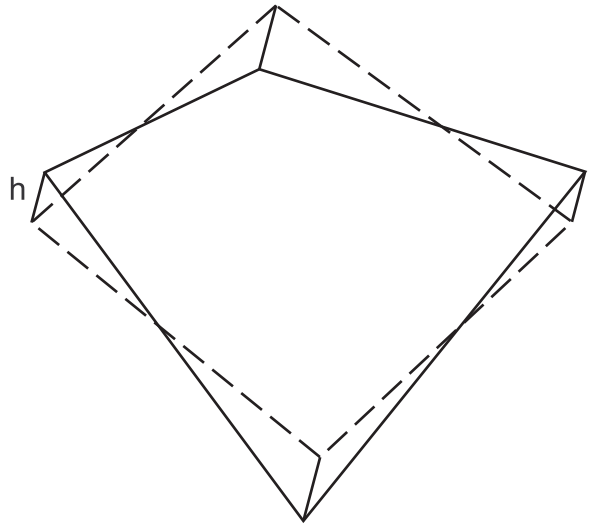
   Figure 12.16: Shell Average Normal Calculation

2. The projected area of the element is computed on a plane through the average normal (the dotted outline on Figure 12.17: Shell Element Projected onto a Plane (p. 378)).

3. The difference in height of the ends of an element edge is computed, parallel to the average normal. In Figure 12.17: Shell Element Projected onto a Plane (p. 378), this distance is 2h. Because of the way the average normal is constructed, h is the same at all four corners. For a flat quadrilateral, the distance is zero.

Figure 12.17: Shell Element Projected onto a Plane



4. The "area warping factor" (F_a^w) for the element is computed as the edge height difference divided by the square root of the projected area.

5. For all shells except those in the "membrane stiffness only" group, if the thickness is available, the "thickness warping factor" is computed as the edge height difference divided by the average element thickness. This could be substantially higher than the area warping factor computed in 4 (above).

6. The warping factor tested against warning and error limits (and reported in warning and error messages) is the larger of the area factor and, if available, the thickness factor.

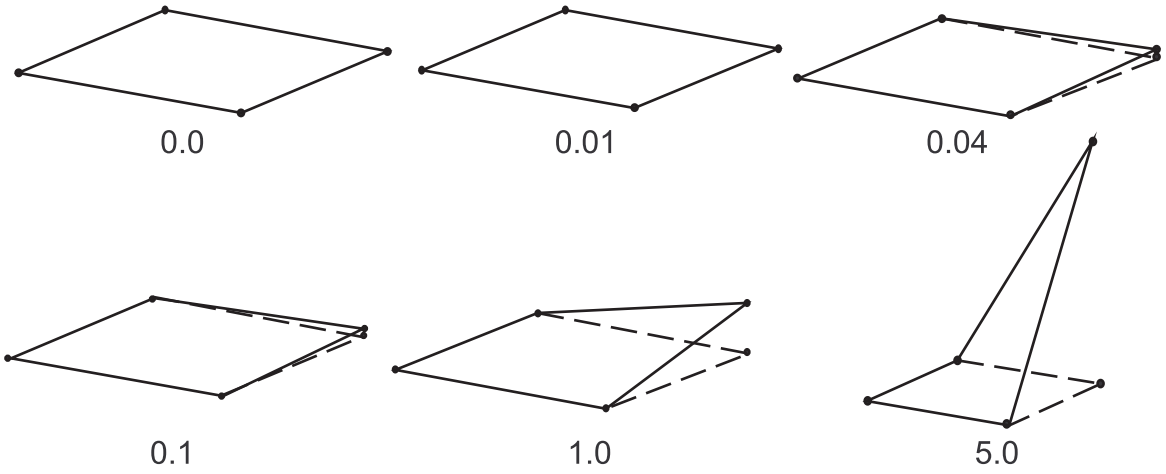
7. The best possible quadrilateral warping factor, for a flat quadrilateral, is zero.

Figure 12.18: Quadrilateral Shell Having Warping Factor (p. 379) shows a "warped" element plotted on top of a flat one. Only the right-hand node of the upper element is moved. The element is a unit square, with a real constant thickness of 0.1.

When the upper element is warped by a factor of 0.01, it cannot be visibly distinguished from the underlying flat one.

When the upper element is warped by a factor of 0.04, it just begins to visibly separate from the flat one.

Figure 12.18: Quadrilateral Shell Having Warping Factor



Warping of 0.1 is visible given the flat reference, but seems trivial; however, it is well beyond the error limit for a membrane shell. Warping of 1.0 is visually unappealing. This is the error limit for most shells.

Warping beyond 1.0 would appear to be obviously unacceptable; however, SHELL181 permits even this much distortion. Furthermore, the warping factor calculation seems to peak at about 7.0. Moving the node further off the original plane, even by much larger distances than shown here, does not further increase the warping factor for this geometry. Users are cautioned that manually increasing the error limit beyond its default of 5.0 for these elements could mean no real limit on element distortion.

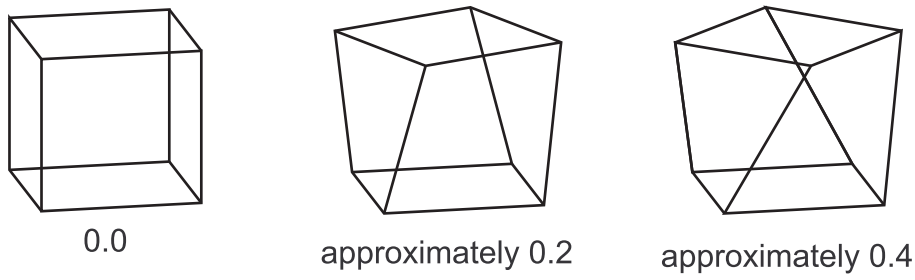
12.1.13.2. Warping Factor Calculation for 3-D Solid Elements

The warping factor for a 3-D solid element face is computed as though the 4 nodes make up a quadrilateral shell element with no real constant thickness available, using the square root of the projected area of the face as described in 4 (above).

The warping factor for the element is the largest of the warping factors computed for the 6 quadrilateral faces of a brick, 3 quadrilateral faces of a wedge, or 1 quadrilateral face of a pyramid.

Any brick element having all flat faces has a warping factor of zero (Figure 12.19: Warping Factor for Bricks (p. 379)).

Figure 12.19: Warping Factor for Bricks



Twisting the top face of a unit cube by 22.5° and 45° relative to the base produces warping factors of about 0.2 and 0.4, respectively.

Table 12.6: Warping Factor Limits

Command to modify	Type of limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,51	warning for "bending with high warping limit" shells (e.g., SHELL181)	1	Elements having warping factors > 1 look like they deserve warnings	Element formulation derived from 8-node solid isn't disturbed by warping. Disturbance of analysis results has not been proven
SHPP,MODIFY,52	same as above, error limit	5	Pushing this limit further does not seem prudent	Valid analyses should not be blocked.
SHPP,MODIFY,53	warning for "non-stress" shells or "bending stiffness included" shells without geometric nonlinearities (e.g., SHELL131)	0.1	The element formulation is based on flat shell theory, with rigid beam offsets for moment compatibility. Informal testing has shown that result error became significant for warping factor > 0.1.	It is difficult to avoid these warnings even with a limit of 0.1.
SHPP,MODIFY,54	same as above, error limit	1	Pushing this limit further does not seem prudent.	Valid analyses should not be blocked.
SHPP,MODIFY,55	warning for "membrane stiffness only" shells (e.g., SHELL41)	0.02	The element formulation is based on flat shell theory, without any correction for moment compatibility. The element cannot handle forces not in the plane of the element.	Informal testing has shown that the effect of warping < 0.02 is negligible.

Command to modify	Type of limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,56	same as above, error limit	0.2	Pushing this limit further does not seem prudent	Valid analyses should not be blocked.
SHPP,MODIFY,57	warning for "shear / twist" shells (e.g., SHELL28)	0.1	The element formulation is based on flat shell theory, with rigid beam offsets for moment compatibility. Informal testing has shown that result error became significant for warping factor > 0.1.	It is difficult to avoid these warnings even with a limit of 0.1.
SHPP,MODIFY,58	same as above, error limit	1	Pushing this limit further does not seem prudent	Valid analyses should not be blocked.
SHPP,MODIFY,60	same as above, error limit	0.01	Pushing this limit further does not seem prudent	Valid analyses should not be blocked.
SHPP,MODIFY,67	warning for 3-D solid element quadrilateral face	0.2	A warping factor of 0.2 corresponds to about a 22.5° rotation of the top face of a unit cube. Brick elements distorted this much look like they deserve warnings.	Disturbance of analysis results has not been proven.
SHPP,MODIFY,68	same as above, error limit	0.4	A warping factor of 0.4 corresponds to about a 45° rotation of the top face of a unit cube. Pushing this limit further does not seem prudent.	Valid analyses should not be blocked.

12.2. Integration Point Locations

The ANSYS program makes use of both standard and nonstandard numerical integration formulas. The particular integration scheme used for each matrix or load vector is given with each element description in Element Library (p. 411). Both standard and nonstandard integration formulas are described in this section. The numbers after the subsection titles are labels used to identify the integration point rule.

For example, line (1, 2, or 3 points) represents the 1, 2, and 3 point integration schemes along line elements. Midside nodes, if applicable, are not shown in the figures in this section.

12.2.1. Lines (1, 2, or 3 Points)

The standard 1-D numerical integration formulas which are used in the element library are of the form:

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{\ell} H_i f(x_i) \tag{12.1}$$

where:

$f(x)$ = function to be integrated
H_i = weighting factor (see Table 12.7: Gauss Numerical Integration Constants (p. 382))
x_i = locations to evaluate function (see Table 12.7: Gauss Numerical Integration Constants (p. 382); these locations are usually the s, t, or r coordinates)
ℓ = number of integration (Gauss) points

Table 12.7: Gauss Numerical Integration Constants

No. Integration Points	Integration Point Locations (x_i)	Weighting Factors (H_i)
1	0.00000.00000.00000	2.00000.00000.00000
2	±0.57735 02691 89626	1.00000.00000.00000
3	±0.77459 66692 41483	0.55555 55555 55556
	0.00000.00000.00000	0.88888 88888 88889

For some integrations of multi-dimensional regions, the method of Equation 12.1 (p. 382) is simply expanded, as shown below.

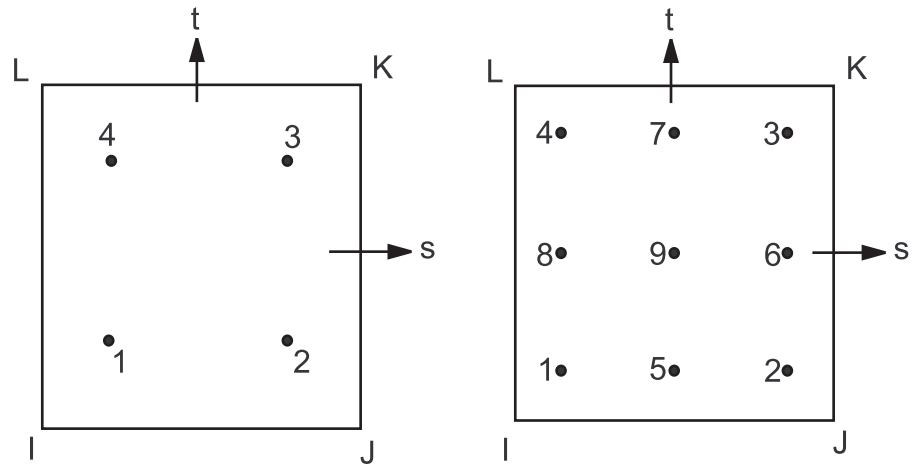
12.2.2. Quadrilaterals (2 x 2 or 3 x 3 Points)

The numerical integration of 2-D quadrilaterals gives:

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)dxdy = \sum_{j=1}^{m}\sum_{i=1}^{\ell} H_j H_i f(x_i, y_j) \tag{12.2}$$

and the integration point locations are shown in Figure 12.20: Integration Point Locations for Quadrilaterals (p. 383). The locations and weighting factors can be calculated using Table 12.7: Gauss Numerical Integration Constants (p. 382) two times.

Figure 12.20: Integration Point Locations for Quadrilaterals



One element models with midside nodes (such as PLANE183) using a 2 x 2 mesh of integration points have been seen to generate spurious zero energy (hourglassing) modes.

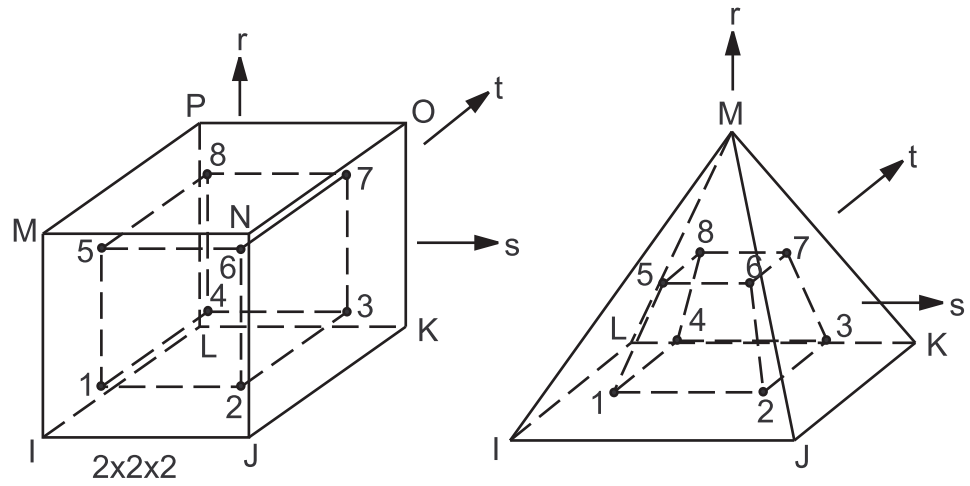
12.2.3. Bricks and Pyramids (2 x 2 x 2 Points)

The 3-D integration of bricks and pyramids gives:

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(x,y,z)dxdydz = \sum_{k=1}^{n}\sum_{j=1}^{m}\sum_{i=1}^{\ell} H_k H_j H_i f(x_i, y_j, z_k) \tag{12.3}$$

and the integration point locations are shown in Figure 12.21: Integration Point Locations for Bricks and Pyramids (p. 383). The locations and weighting factors can be calculated using Table 12.7: Gauss Numerical Integration Constants (p. 382) three times.

Figure 12.21: Integration Point Locations for Bricks and Pyramids



One element models with midside nodes using a 2 x 2 x 2 mesh of integration points have been seen to generate spurious zero energy (hourglassing) modes.

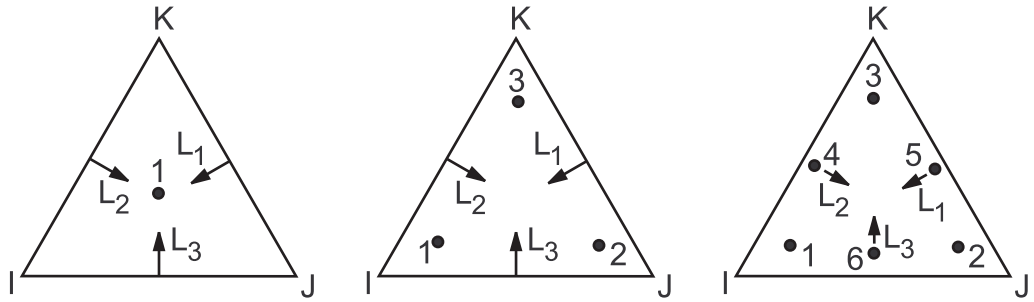
12.2.4. Triangles (1, 3, or 6 Points)

The integration points used for these triangles are given in Table 12.8: Numerical Integration for Triangles (p. 384) and appear as shown in Figure 12.22: Integration Point Locations for Triangles (p. 384). L varies from 0.0 at an edge to 1.0 at the opposite vertex.

Table 12.8: Numerical Integration for Triangles

Type		Integration Point Location	Weighting Factor
1 Point Rule		$L_1=L_2=L_3=.3333333$	1.000000
3 Point Rule		$L_1=.66666\ 66666\ 66666$ $L_2=L_3=.16666\ 66666\ 66666$ Permute L_1, L_2, and L_3 for other locations)	0.33333 33333 33333
6 Point Rule	Corner Points	$L_1=0.81684\ 75729\ 80459$ $L_2=L_3=0.09157\ 62135\ 09661$ Permute L_1, L_2, and L_3 for other locations)	0.10995 17436 55322
	Edge Center Points	$L_1=0.10810\ 30181\ 6807$ $L_2=L_3=0.44594\ 84909\ 15965$ Permute L_1, L_2, and L_3 for other locations)	0.22338 15896 78011

Figure 12.22: Integration Point Locations for Triangles



12.2.5. Tetrahedra (1, 4, 5, or 11 Points)

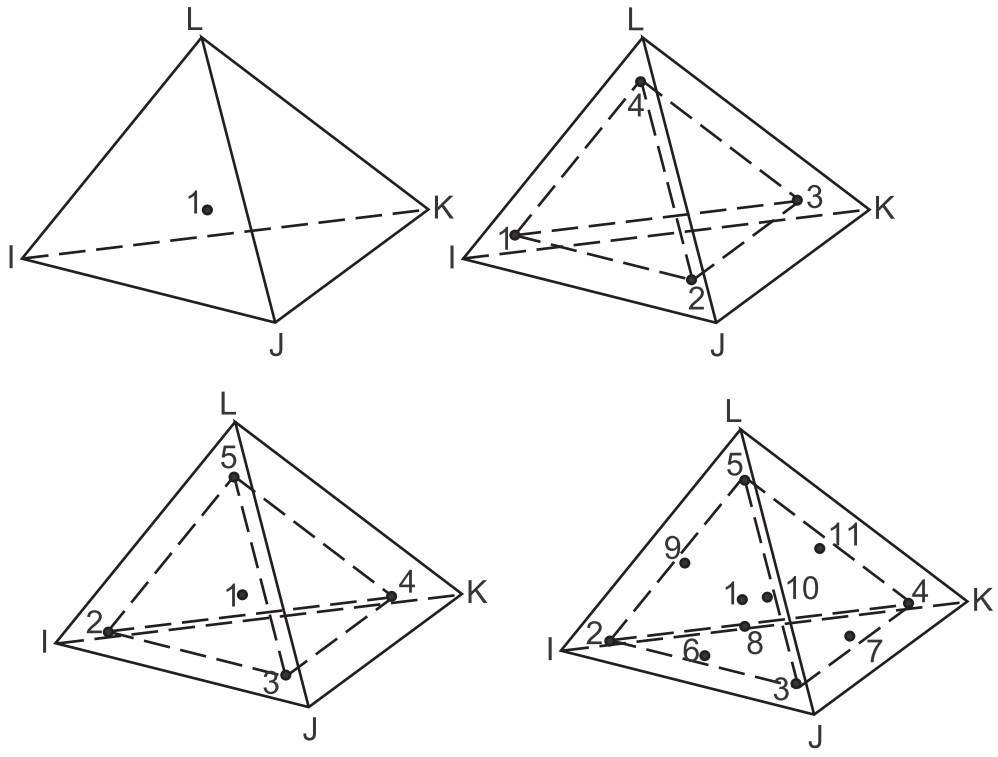
The integration points used for tetrahedra are given in Table 12.9: Numerical Integration for Tetrahedra (p. 385).

Table 12.9: Numerical Integration for Tetrahedra

Type		Integration Point Location	Weighting Factor
1 Point Rule	Center Point	$L_1=L_2=L_3=L_4$=.25000 00000 00000	1.00000 00000 00000
4 Point Rule	Corner Points	L_1=.58541 01966 24968 $L_2=L_3=L_4$=.13819 66011 25010 Permute L_1, L_2, L_3, and L_4 for other locations)	0.25000 00000 00000
5 Point Rule	Center Point	$L_1=L_2=L_3=L_4$=.25000 00000 00000	-0.80000 00000 00000
	Corner Points	L_1=.50000 00000 00000 $L_2=L_3=L_4$=.16666 66666 66666 Permute L_1, L_2, L_3, and L_4 for other locations)	0.45000 00000 00000
11 Point Rule	Center Point	$L_1=L_2=L_3=L_4$=.25000 00000 00000	0.01315 55555 55555
	Corner Point	$L_1=L_2=L_3$=.0714285714285714 L_4=.78571 42857 14286 (Permute L_1, L_2, L_3 and L_4 for other three locations)	0.00762 22222 22222
	Edge Center Points	$L_1=L_2$=0.39940 35761 66799 $L_3=L_4$=0.10059 64238 33201 Permute L_1, L_2, L_3 and L_4 such that two of L_1, L_2, L_3 and L_4 equal 0.39940 35761 66799 and the other two equal 0.10059 64238 33201 for other five locations	0.02488 88888 88888

These appear as shown in Figure 12.23: Integration Point Locations for Tetrahedra (p. 386). L varies from 0.0 at a face to 1.0 at the opposite vertex.

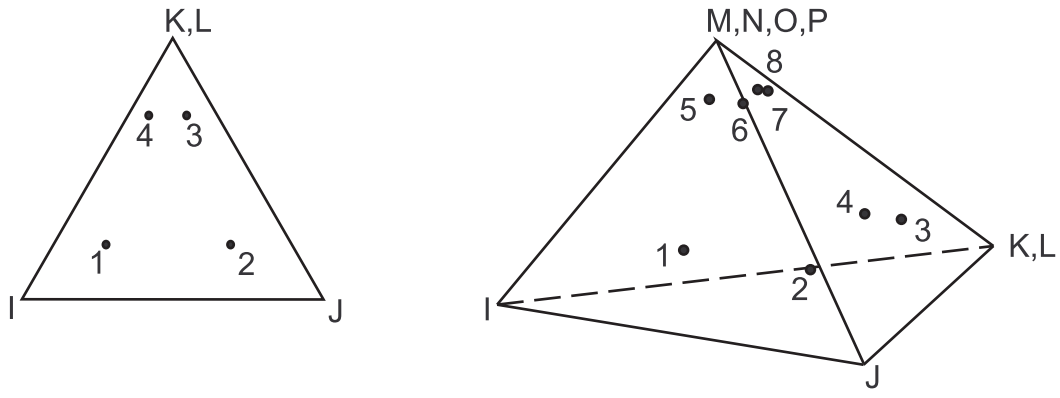
Figure 12.23: Integration Point Locations for Tetrahedra



12.2.6. Triangles and Tetrahedra (2 x 2 or 2 x 2 x 2 Points)

These elements use the same integration point scheme as for 4-node quadrilaterals and 8-node solids, as shown in Figure 12.24: Integration Point Locations for Triangles and Tetrahedra (p. 386). The locations and weighting factors can be calculated using Table 12.7: Gauss Numerical Integration Constants (p. 382) two or three times.

Figure 12.24: Integration Point Locations for Triangles and Tetrahedra

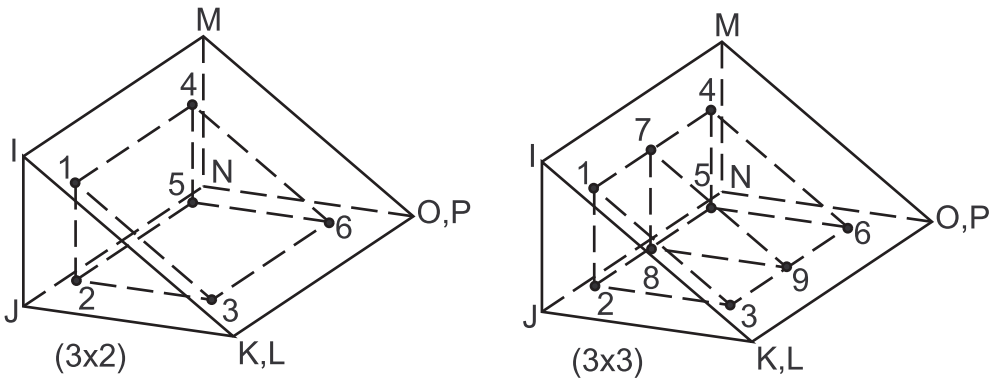


3x3 and 3x3x3 cases are handled similarly.

12.2.7. Wedges (3 x 2 or 3 x 3 Points)

These wedge elements use an integration scheme that combines linear and triangular integrations, as shown in Figure 12.25: 6 and 9 Integration Point Locations for Wedges (p. 387). The locations and weighting factors can be calculated using Table 12.7: Gauss Numerical Integration Constants (p. 382) and Table 12.8: Numerical Integration for Triangles (p. 384).

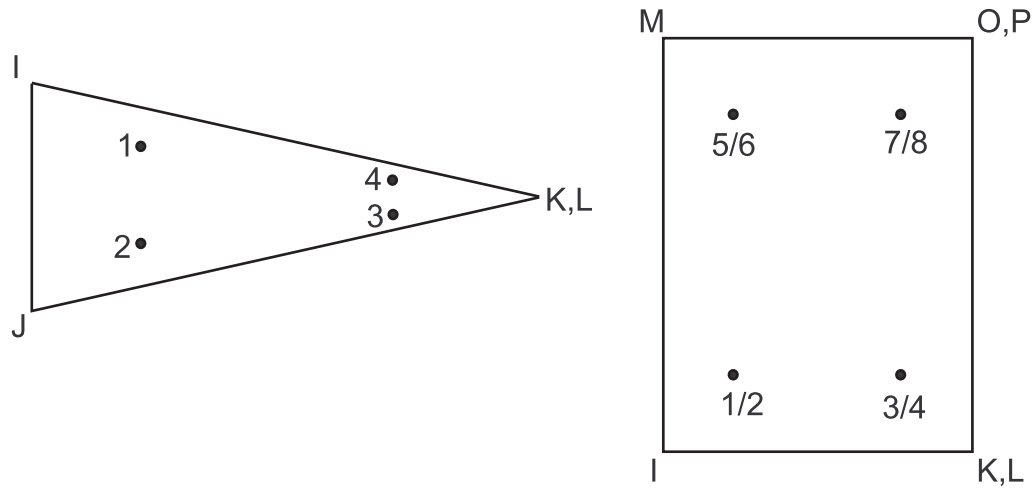
Figure 12.25: 6 and 9 Integration Point Locations for Wedges



12.2.8. Wedges (2 x 2 x 2 Points)

These wedge elements use the same integration point scheme as for 8-node solid elements as shown by two orthogonal views in Figure 12.26: 8 Integration Point Locations for Wedges (p. 387). The locations and weighting factors can be calculated using Table 12.7: Gauss Numerical Integration Constants (p. 382) three times.

Figure 12.26: 8 Integration Point Locations for Wedges



12.2.9. Bricks (14 Points)

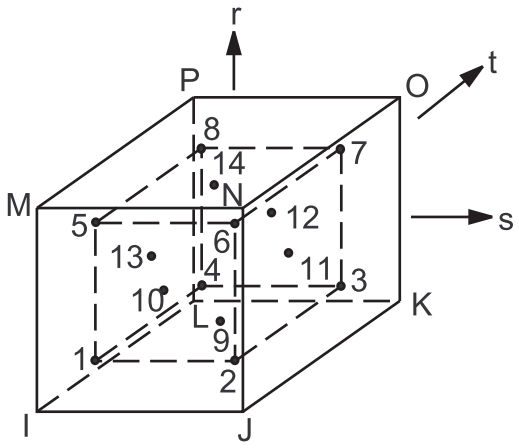
The 20-node solid uses a different type of integration point scheme. This scheme places points close to each of the 8 corner nodes and close to the centers of the 6 faces for a total of 14 points. These locations are given in Table 12.10: Numerical Integration for 20-Node Brick (p. 388):

Table 12.10: Numerical Integration for 20-Node Brick

Type		Integration Point Location	Weighting Factor
14 Point Rule	Corner Points	s = ±.75868 69106 39328 t = ±.75878 69106 39329 r = ±.75878 69106 39329	.33518 00554 01662
	Center Points	s = ±.79582 24257 54222, t=r=0.0 t = ±.79582 24257 54222, s=r=0.0 r = ±.79582 24257 54222, s=t=0.0	.88642 65927 97784

and are shown in Figure 12.27: Integration Point Locations for 14 Point Rule (p. 388).

Figure 12.27: Integration Point Locations for 14 Point Rule



12.2.10. Nonlinear Bending (5 Points)

Both beam and shell elements that have nonlinear materials must have their effects accumulated thru the thickness. This uses nonstandard integration point locations, as both the top and bottom surfaces have an integration point in order to immediately detect the onset of the nonlinear effects.

Table 12.11: Thru-Thickness Numerical Integration

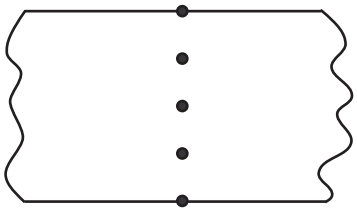
Type	Integration Point Location[1]	Weighting Factor
5	±0.500	0.1250000

Type	Integration Point Location[1]	Weighting Factor
	±0.300	0.5787036
	0.000	0.5925926

1. Thickness coordinate going from -0.5 to 0.5.

These locations are shown in Figure 12.28: Nonlinear Bending Integration Point Locations (p. 389).

Figure 12.28: Nonlinear Bending Integration Point Locations



12.2.11. General Axisymmetric Elements

The numerical integration of general axisymmetric elements gives:

$$\int_0^{2\pi}\int_{-1}^{1}\int_{-1}^{1} f(r,z,\theta)dr\,dz\,2\pi r d\theta = \sum_{k=1}^{n}\sum_{k=1}^{m}\sum_{k=1}^{l} H_k H_j H_i f(r_i, z_j, \theta_k) \quad (12.4)$$

H_i and H_j are weighting factors on the r_z plane, as shown in Figure 11.20: General Axisymmetric Solid Elements (when N_{np} = 3) (p. 359). The values are shown in Table 12.7: Gauss Numerical Integration Constants (p. 382). In circumferential direction θ:

$$\theta_k = (k-1)\frac{\pi}{NP} \qquad k = 1 \ldots 2NP \qquad H_k = \frac{\pi r_i}{NP} \quad (12.5)$$

12.3. Temperature-Dependent Material Properties

Temperature-dependent material properties are evaluated at each integration point. Elements for which this applies include SHELL181, PLANE182, PLANE183 , SOLID185, SOLID186 , SOLID187, SOLID272, SOLID273, SOLID285, SOLSH190, BEAM188, BEAM189, SHELL208, SHELL209, REINF264, SHELL281, PIPE288, PIPE289, and ELBOW290. Elements using a closed form solution (without integration points) have their material properties evaluated at the average temperature of the element. Elements for which this applies include SHELL28, and LINK180 .

Other cases:

- For the structural elements PLANE13 and SOLID65, the nonlinear material properties (**TB** commands) are evaluated at the integration points, but the linear material properties (**MP** commands) are evaluated at the average element temperature.
- Numerically integrated structural elements PLANE25, SHELL41, SHELL61, and PLANE83 have their linear material properties evaluated at the average element temperature.

- Non-structural elements have their material properties evaluated only at the average element temperature, except for the specific heat (C_p) which is evaluated at each integration point.

Whether shape functions are used or not, materials are evaluated at the temperature given, i.e. no account is made of the temperature offset (**TOFFST** command).

For a stress analysis, the temperatures used are based directly on the input. As temperature is the unknown in a heat transfer analysis, the material property evaluation cannot be handled in the same direct manner. For the first iteration of a heat transfer analysis, the material properties are evaluated at the uniform temperature (input on the **BFUNIF** command). The properties of the second iteration are based on the temperatures of the first iteration. The properties of the third iteration are based on the temperatures of the second iteration, etc.

See Temperature-Dependent Coefficient of Thermal Expansion (p. 10) for a special discussion about the coefficient of thermal expansion.

12.4. Positive Definite Matrices

By definition, a matrix [D] (as well as its inverse $[D]^{-1}$) is positive definite if the determinants of all submatrices of the series:

$$[D_{1,1}], \begin{bmatrix} D_{1,1} & D_{1,2} \\ D_{2,1} & D_{2,2} \end{bmatrix}, \begin{bmatrix} D_{1,1} & D_{1,2} & D_{1,3} \\ D_{2,1} & D_{2,2} & D_{2,3} \\ D_{3,1} & D_{3,2} & D_{3,3} \end{bmatrix}, \text{ etc.} \tag{12.6}$$

including the determinant of the full matrix [D], are positive. The series could have started out at any other diagonal term and then had row and column sets added in any order. Thus, two necessary (but not sufficient) conditions for a symmetric matrix to be positive definite are given here for convenience:

$$D_{i,i} > 0.0 \tag{12.7}$$

$$D_{i,j} < \sqrt{D_{i,i} D_{j,j}} \tag{12.8}$$

If any of the above determinants are zero (and the rest positive), the matrix is said to be positive semidefinite. If all of the above determinants are negative, the matrix is said to be negative definite.

12.4.1. Matrices Representing the Complete Structure

In virtually all circumstances, matrices representing the complete structure with the appropriate boundary conditions must be positive definite. If they are not, the message "NEGATIVE PIVOT . . ." appears. This usually means that insufficient boundary conditions were specified. An exception is a piezoelectric analysis, which works with negative definite matrices, but does not generate any error messages.

12.4.2. Element Matrices

Element matrices are often positive semidefinite, but sometimes they are either negative or positive definite. For most cases where a negative definite matrix could inappropriately be created, the program will abort with a descriptive message.

12.5. Lumped Matrices

Some of the elements allow their consistent mass or specific heat matrices to be reduced to diagonal matrices (accessed with the **LUMPM**,ON command). This is referred to as "lumping".

12.5.1. Diagonalization Procedure

One of two procedures is used for the diagonalization, depending on the order of the element shape functions. The mass matrix is used as an example.

For lower order elements (linear or bilinear) the diagonalized matrix is computed by summing rows (or columns). The steps are:

1. Compute the consistent mass matrix $([M_e'])$ in the usual manner.
2. Compute:

$$S(i) = \sum_{j=1}^{n} M_e'(i,j) \quad \text{for } i = 1, n \tag{12.9}$$

   where:

   n = number of degrees of freedom (DOFs) in the element

3. Set

$$M_e(i,j) = 0.0 \qquad \text{for } i \neq j \tag{12.10}$$

$$M_e(i,j) = S(i) \qquad \text{for } i = 1, n \tag{12.11}$$

For higher order elements the procedure suggested by Hinton, et al.([45] (p. 923)), is used. The steps are:

1. Compute the consistent mass matrix $([M_e'])$ in the usual manner.
2. Compute:

$$S = \sum_{i=1}^{n} \sum_{j=1}^{n} M_e'(i,j) \tag{12.12}$$

$$D = \sum_{i=1}^{n} M_e'(i,i) \tag{12.13}$$

3. Set:

$$M_e(i,j) = 0.0 \qquad \text{if } i \neq j \tag{12.14}$$

$$M_e(i,i) = \frac{S}{D} M_e'(i,i) \tag{12.15}$$

Note that this method ensures that:

1. The element mass is preserved

2. The element mass matrix is positive definite

It may be observed that if the diagonalization is performed by simply summing rows or columns in higher order elements, the resulting element mass matrix is not always positive definite.

12.5.2. Limitations of Lumped Mass Matrices

Lumped mass matrices have the following limitations:

1. Elements containing both translational and rotational degrees of freedom will have mass contributions only for the translational degrees of freedom. Rotational degrees of freedom are included for:

   - SHELL181, SHELL208, SHELL209, SHELL281, and ELBOW290 unless an unbalanced laminate construction is used.
   - BEAM188, BEAM189, PIPE288, and PIPE289 if there are no offsets.

2. Lumping, by its very nature, eliminates the concept of mass coupling between degrees of freedom. Therefore, the following restrictions exist:

   - Lumping is not allowed for FLUID29, FLUID30, or FLUID38 elements.
   - Lumping is not allowed for the mass matrix option of MATRIX27 elements if it is defined with nonzero off-diagonal terms.
   - The use of lumping with constraint equations may effectively cause the loss of some mass for analyses that involve a mass matrix. For example, in modal analyses this typically results in higher frequencies. This loss of mass comes about because of the generation of off-diagonal terms by the constraint equations, which then are ignored.

     The exceptions to this are substructuring generation passes with the sparse solver and the PCG Lanczos mode extraction method in modal analyses. These exceptions contain the off-diagonal terms when lumped mass is used with constraint equations. It is important to note however, that the assembled mass matrix in a `jobname.FULL` file generated by the PCG Lanczos mode extraction method will not contain the off-diagonal mass terms for this case.

12.6. Reuse of Matrices

Matrices are reused automatically as often as possible in order to decrease running time. The information below is made available for use in running time estimates.

12.6.1. Element Matrices

For static (**ANTYPE**,STATIC) or full transient dynamic (**ANTYPE**,TRANS with **TRNOPT**,FULL) analyses, element stiffness/conductivity, mass, and damping/specific heat, matrices ($[K_e]$, $[M_e]$, $[C_e]$) are always reused from iteration to iteration, except when:

1. The full Newton-Raphson option (**NROPT**,FULL) is used, or for the first equilibrium iteration of a time step when the modified Newton-Raphson option (**NROPT**,MODI) is used and the element has either nonlinear materials or large deformation (**NLGEOM**,ON) is active.

2. The element is nonlinear (e.g. gap, radiation, or control element) and its status changes.

3. MODE or ISYM (**MODE** command) have changed from the previous load step for elements PLANE25, SHELL61, PLANE75, PLANE78, or PLANE83.

4. $[K_e^t]$ will be reformulated if a convective film coefficient (input on the **SF** or **SFE** commands) on an element face changes. Such a change could occur as a ramp (**KBC**,0) within a load step.

5. The materials or real constants are changed by new input, or if the material properties have changed due to temperature changes for temperature-dependent input.

Element stress stiffness matrices $[S_e]$ are never reused, as the stress normally varies from iteration to iteration.

12.6.2. Structure Matrices

The overall structure matrices are reused from iteration to iteration except when:

1. An included element matrix is reformed (see above).

2. The set of specified degrees of freedom (DOFs) is changed.

3. The integration time step size changes from that used in the previous substep for the transient (**ANTYPE**,TRANS) analysis.

4. The first iteration of a restart is being performed.

12.6.3. Override Option

The above tests are all performed automatically by the program. The user can select to override the program's decision with respect to whether the matrices should be reformed or not. For example, if the user has temperature-dependent input as the only cause which is forcing the reformulation of the matrices, and there is a load step where the temperature dependency is not significant, the user can select that the matrices will not be reformed at that load step (**KUSE**,1). (Normally, the user would want to return control back to the program for the following load step (**KUSE**,0)). On the other hand, the user can select that all element matrices are to be reformed each iteration (**KUSE**,-1).

12.7. Hydrostatic Loads

The following topics concerning hydrostatic loads are available:

12.7.1. Buckling of Pipe Cross-Sections

Hydrostatic effects may affect the outside and the inside of a pipe. Pressure on the outside crushes the pipe and buoyant forces on the outside tend to raise the pipe to the water surface. Pressure on the inside tends to stabilize the pipe cross-section.

The buoyant force for a completely submerged element acting in the positive z direction is:

$$\{F/L\}_b = C_b \rho_w \frac{\pi}{4} D_e^2 \{g\} \tag{12.16}$$

where: $\{F/L\}_b$ = vector of loads per-unit-length due to buoyancy

C_b = coefficient of buoyancy (input as *Cb* on the **OCDATA** command.
{g} = acceleration vector (input via the **ACEL** command with only *ACEL_Z* being nonzero)

Also, an adjustment for the added mass term is made.

The crushing pressure at a node is:

$$P_o^s = -\rho_w g(Z - Z_{msl}) + P_o^a \tag{12.17}$$

where:

P_o^s = crushing pressure due to hydrostatic effects
g = acceleration due to gravity (input as *ACEL_Z* on the **ACEL** command)
Z = vertical coordinate of the node
Z_{msl} = vertical offset from the global origin to the mean sea level (input on the **OCDATA** command)
P_o^a = input external pressure (input as face 2 on **SFE** command)

The internal (bursting) pressure is:

$$P_i = -\rho_o g(Z - Z_{fo}) + P_i^a \tag{12.18}$$

where:

P_i = internal pressure
ρ_o = internal fluid density
Z_{fo} = Z coordinate of free surface of fluid (input as face 3 on **SFE** command)
P_i^a = input internal pressure (input as face 1 on **SFE** command)

To ensure that the problem is physically possible as input, a check is made at the element midpoint to see if the cross-section collapses under the hydrostatic effects. The cross-section is assumed to be unstable if:

$$P_o^s - P_i > \frac{E}{4(1-\nu^2)}\left(\frac{2t_w}{D_o}\right)^3 \tag{12.19}$$

where:

E = Young's modulus (input as EY on **MP** command)
ν = Poisson's ratio (input as PRXY or NUXY on **MP** command)

12.7.2. Effect of Water Pressure on Elements

Elements such as PIPE288 have three direct stress components: axial, hoop, and radial. Elements such as LINK180 and BEAM188, however, have only an axial stress component; therefore, an adjustment is necessary to include the effect of the three stress components of a hydrostatic load.

Consider the first line of Equation 2.2 (p. 6), assuming isotropic materials and no thermal effect:

$$\varepsilon_x = \frac{1}{E}(\sigma_x - \nu\sigma_y - \nu\sigma_z) \tag{12.20}$$

where:

ε_x = axial strain
E = Young's modulus (input as EX on the **MP** command)
σ_x = axial stress
ν = Poisson's ratio (input as PRXY or NUXY on the **MP** command)
$\sigma_y = \sigma_z$ = stresses normal to the axial direction

For hydrostatic loads:

$$\sigma_h = \sigma_x = \sigma_y = \sigma_z = \rho g(Z - Z_{msl}) \tag{12.21}$$

where:

σ_h = hydrostatic pressure

Combining Equation 12.21 (p. 395) and Equation 12.22 (p. 395):

$$\varepsilon_x = \frac{\sigma_h}{E}(1 - 2\nu) \tag{12.22}$$

The factor (1 - 2ν) is used to give the correct strain. Thus, the following restrictions apply to this approach:

- Material must be isotropic.
- Material must be elastic.
- Fluid pressure is assumed to be hydrostatic everywhere (that is, CTUBE and HREC beam subtypes presume that the flooding option is applied, and there is no option for removing it).

It is of course preferable to use the pipe element rather than the beam element, as doing so avoids the limitations that apply to the beam element with subtype CTUBE.

Because hydrostatic pressure (stress) is normally much lower than the working stress, the restrictions described above typically need to be considered only for extremely deep applications.

12.8. Hydrodynamic Loads

Hydrodynamic effects may occur because the structure moves in a motionless fluid, the structure is fixed but fluid motion is present, or both the structure and fluid are moving.

Fluid motion consists of two parts: current and wave motions. Current is input by giving the current velocity and direction (input as W(i) and θ(i)) at up to eight different vertical stations (input as Z(i)). (All input quantities referred to in this section not otherwise identified come from the **OCTYPE**, **OCDATA**, and **OCTABLE** commands.

The velocity and direction are interpolated linearly between stations. The current is assumed to flow horizontally only. The wave may be input using one of the wave theories (K_w, input as *KWAVE* on the **OCDATA** command) in the following table :

Table 12.12: Wave Theory Table

Wave Theory	K_w
Small amplitude wave, unmodified (Airy wave theory), (Wheeler([35] (p. 922)))	0
Small amplitude wave, modified with empirical depth decay function, (Wheeler([35] (p. 922)))	1
Stokes fifth order wave, (Skjelbreia([31] (p. 922)))	2
Stream function wave, (Dean([59] (p. 923)), Dalrymple([417] (p. 944)))	3
Random wave	5
Shell new wave, (Atkins([391] (p. 943)))	6
Constrained new wave, (O'Neill([392] (p. 943)), Pinna([393] (p. 943)))	7
Diffracted wave	8

These related topics follow:

12.8.1. Regular Waves on Line Elements

Regular waves include both linear (Airy) and non-linear (Stoke's fifth order and Dean's stream function) waves.

12.8.1.1. Small Amplitude Linear (Airy) Wave (K_w = 0, 1)

The free surface of the wave is defined by:

$$\eta_s = \sum_{i=1}^{N_w} \eta_i = \sum_{i=1}^{N_w} \frac{H_i}{2} \cos\beta_i \qquad (12.23)$$

where:

η_s = total wave amplitude
η_w = number of waves
η_i = wave amplitude of component i
H_i = surface coefficient (component height) [input quantity H(i) on the **OCTABLE** command]

$$\beta_i = \begin{cases} 2\pi\left(\frac{R}{\lambda_i} - \frac{t}{\tau_i} + \frac{\phi_i}{360}\right) & \text{if } L_w = 0 \\ 0.0 & \text{if } L_w = 1 \\ \frac{\pi}{2} & \text{if } L_w = 2 \\ -\frac{\pi}{2} & \text{if } L_w = 3 \\ \pi & \text{if } L_w = 4 \end{cases}$$

R = radial distance to point on element from origin in the X-Y plane in the direction of the wave
λ_i = wave length (input as L(i) on the **OCTABLE** command if L(i) > 0.0, or derived from Equation 12.24 (p. 397) otherwise
t = time elapsed (input as *TIME* on the **TIME** command). The default value of *TIME* is usually not desired. If zero is needed, use ($1.0E^{-12}$) instead.
τ_i = wave period (input as T(i) on the **OCTABLE** command)
ϕ_i = phase shift (input as Ps(i) on the **OCDATA** command)
L_w = wave location (input as *WAVELOC* on the **OCDATA** command)

If K_w = 0 or 1, and N_w > 1, the wave is considered to be irregular.

If λ_i is not input (set to zero) and K_w = 0 or 1, λ_i is calculated iteratively from:

$$\lambda_i = \lambda_i^d \tanh\left(\frac{2\pi d}{\lambda_i}\right) \tag{12.24}$$

where:

λ_i = output quantity small amplitude wave length

$\lambda_i^d = \frac{g(\tau_i)^2}{2\pi}$ = output quantity deep water wave length

g = acceleration due to gravity (Z direction) (input via the **ACEL** command)
d = water depth (input as *DEPTH* on the **OCDATA** command)

Each component of wave height is checked to verify that it satisfies the Miche criterion. The check ensures that the wave is not a breaking wave, which the included wave theories do not cover. A breaking wave is one that spills over its crest, normally in shallow water. A warning message is issued if:

$$H_i > H_b \tag{12.25}$$

where:

$$H_b = 0.142\lambda_i \tanh\left(\frac{2\pi d}{\lambda_i}\right) = \text{height of breaking wave}$$

When using wave loading, an error check occurs to ensure that the input acceleration does not change after the first load step, as this would imply a change in the wave behavior between load steps.

The particle velocities at integration points are calculated as a function of depth from:

$$\vec{v}_R = \sum_{i=1}^{N_w} \frac{\cosh(k_i \bar{Z} f)}{\sinh(k_i d)} \frac{2\pi}{\tau_i} \eta_i + \vec{v}_D \tag{12.26}$$

$$\vec{v}_Z = \sum_{i=1}^{N_w} \frac{\sinh(k_i \bar{Z} f)}{\sinh(k_i d)} \dot{\eta}_i \tag{12.27}$$

where:

$\vec{v}_R$ = radial particle velocity

$\vec{v}_Z$ = vertical particle velocity

$k_i = 2\pi/\lambda_i$

$\bar{Z}$ = height of integration point above the ocean floor = d + Z

$\dot{\eta}_i$ = time derivative of η_i

$\vec{v}_D$ = drift velocity (input as *W* on the **OCTABLE** command)

$$f = \begin{cases} 1.0 & \text{if } K_w = 0 \text{ (small amplitude wave theory)} \\ \dfrac{d}{d+\eta_s} & \text{if } K_w = 1 \text{ (Wheeler(35))} \end{cases}$$

The particle accelerations are computed by differentiating $\vec{v}_R$ and $\vec{v}_Z$ with respect to time. Thus:

$$\dot{\vec{v}}_R = \sum_{i=1}^{N_w} \frac{\cosh(k_i \bar{Z} f)}{\sinh(k_i d)} \left(\frac{2\pi}{\tau_i}\right)(\eta_i - C\dot{\eta}_i) \tag{12.28}$$

$$\dot{\vec{v}}_Z = \sum_{i=1}^{N_w} \frac{\sinh(k_i \bar{Z} f)}{\sinh(k_i d)} \left(\frac{2\pi}{\tau_i}\right)\left(-\frac{2\pi}{\tau_i}\eta_i - C\dot{\eta}_i\left(\frac{\tau}{2\pi}\right)\right) \tag{12.29}$$

where:

$$C = \begin{cases} \dot{\eta}_s \dfrac{2\Pi}{\lambda_i} \dfrac{\bar{Z}d}{(d+\eta_s)^2} & \text{if } K_w = 0 \text{ (small amplitude wave theory)} \\ 0.0 & \text{if } K_w = 1 \text{ (Wheeler(35))} \end{cases}$$

Expanding equation 2.29 of the Shore Protection Manual([43] (p. 923)) for a multiple component wave, the wave hydrodynamic pressure is:

$$P_o^d = \rho_w g \sum_{i=1}^{N_w} \eta_i \frac{\cosh\left[2\pi \dfrac{\bar{Z}}{\lambda_i}\right]}{\cosh\left[2\pi \dfrac{d}{\lambda_i}\right]} \tag{12.30}$$

However, use of this equation leads to nonzero total pressure at the surface at the crest or trough of the wave. Thus, Equation 12.30 (p. 398) is modified to be:

$$P_o^d = \rho_w g \sum_{i=1}^{N_w} \eta_i \frac{\cosh\left[2\pi \frac{\bar{Z}d}{\lambda_i d + \eta_s}\right]}{\cosh\left[2\pi \frac{d}{\lambda_i}\right]} \tag{12.31}$$

which does result in a total pressure of zero at all points of the free surface. This dynamic pressure, which is calculated at the integration points during the stiffness pass, is extrapolated to the nodes for the stress pass. The hydrodynamic pressure for Stokes fifth order wave theory is:

$$P_o^d = \rho_w g \sum_{i=1}^{5} \eta_i \frac{\cosh\left(2\pi \frac{\bar{Z}}{\lambda_i}\right)}{\cosh\left(2\pi \frac{d}{\lambda_i}\right)} \tag{12.32}$$

12.8.1.2. Stoke's Fifth Order Wave ($K_w = 2$)

The Stokes fifth order wave theory is discussed by Skjelbreia ([31] (p. 922)). The modification suggested by Nishimura ([143] (p. 928)) has been included.

12.8.1.3. Dean's Stream Function Wave ($K_w = 3$)

The stream function wave theory is described by Dean ([59] (p. 923)). A solution algorithm that uses the Lagrange multiplier technique suggested by Dalrymple ([417] (p. 944)) is included as a base algorithm.

12.8.2. Irregular Waves on Line Elements ($K_w = 5$ through 7)

The random wave ($K_w = 5$), Shell new wave ($K_w = 6$) and constrained new wave ($K_w = 7$) models assume that the wave particle kinematics are generated from a wave spectrum using linear theory.

A wave spectrum is characterized by a curve of spectral density of wave energy $S(\omega_i)$. The wave spectrum can be defined in one of the three forms: Pierson-Moskowitz ([394] (p. 943)), JONSWAP ([395] (p. 943)), or user-defined.

These related topics follow:
12.8.2.1. Random Wave
12.8.2.2. Shell New Wave
12.8.2.3. Constrained New Wave

12.8.2.1. Random Wave

The random wave is created by adding the parameters (wave height, velocity, and acceleration) of a number of regular Airy waves (wave components) with amplitudes corresponding to the required spectrum and with random phases.

With the Pierson-Moskowitz and JONSWAP spectrum, the spectrum is divided into a number of equal energy strips (equal areas) based on the number of wave components specified (input as *NWC* on the **OCTABLE** command). Each of the strips is a wave component, the frequency of which is defined at the centroid of the strip. The amplitude is given by the square root of twice the area of the strip and, due to the equal-energy-based strips, remains constant for each wave component.

For a user-defined spectrum, the frequency and energy parameters to form the spectrum are input, so the amplitudes may vary for each component. The solution is most stable numerically when wave component amplitudes are kept constant throughout the definition.

The free surface of the wave can be defined by

$$\eta_s = \sum_{i=1}^{N_w} \eta_i = \sum_{i=1}^{N_w} \frac{H_i}{2} \cos\beta_i$$

but with revised definitions for some of the parameters, as follows:

N_W = number of wave components (input as *NWC* on the **OCTABLE** command)

H_i = surface coefficient = $2\sqrt{2\frac{m_0}{N_w}}$ where m_o is the area under the full spectrum

β_i = $2\pi(\frac{R_i}{\lambda_i} - \frac{t}{\tau_i} + \frac{\varphi_i}{360})$

λ_i = wave length derived from Equation 12.24 (p. 397)

τ_i = wave period = $\frac{2\pi}{\omega_i}$, where ω_i is the angular frequency for wave component *i*

φ_i = phase shift (based on a randomly generated number)

For a given initial seed value (input as *SEED* on the **OCDATA** command), the same sequence of random phases is generated in every run. Changing the initial seed generates a different set of random phases.

A form of Wheeler stretching, known as *delta stretching*, is introduced to provide the wave kinematics under the crest. The depth-mapping function modifies the conventional cosh and sinh terms in the wave kinematic equations to provide linear extrapolation of the velocities and accelerations above mean water level so that unrealistic terms are prevented from developing.

Similar to equations Equation 12.26 (p. 398) and Equation 12.27 (p. 398), the radial and vertical fluid particle velocities are given by:

$$\overrightarrow{v_R} = f_K \sum_{i=1}^{N_w} G_R \frac{2\pi}{\tau_i} \eta_i + \overrightarrow{v_D} \tag{12.33}$$

$$\overrightarrow{v_Z} = \sum_{i=1}^{N_w} G_Z \frac{2\pi}{\tau_i} \dot{n}_i \tag{12.34}$$

where:

G_R = radial depth-mapping function, computed as:

$$G_R = \frac{\cosh(k_i d)}{\sinh(k_i d)} + k_i Z' \quad \text{for } Z' \geq 0$$

$$G_R = \frac{\cosh(k_i (Z' + d))}{\sinh(k_i d)} \quad \text{for } Z' < 0$$

G_Z = vertical depth-mapping function, computed as:

$$G_Z = \frac{\sinh(k_i(Z'+d))}{\sinh(k_i d)} \qquad \text{for } Z' < 0$$

$$G_Z = \frac{\cosh(k_i d)}{\sinh(k_i d)} k_i Z' + 1 \qquad \text{for } Z' \geq 0$$

f_K = wave kinematics factor (input as *KINE* on the **OCDATA** command) that accounts for the effect of wave-spreading on the radial velocity and acceleration

k_i = wave number = $\frac{2\pi}{\lambda_i}$

The effective depth Z′ is used to compute the depth functions and is given by:

$$Z' = \frac{(d+\Delta\eta_s)}{(d+\eta_s)}(Z+d) - d \qquad \text{for } \eta_s > 0$$
$$Z' = Z \qquad \text{for } \eta_s \leq 0 \text{ or } Z < Z_{DS} \tag{12.35}$$

where:

Δ = delta-stretching parameter (input as *DELTA* on the **OCDATA** command)
$Z_{DS} = -f_{DS}D_s$ = depth under the mean water level in which stretching starts to become effective
f_{DS} = stretching depth factor (input as *DSA* on the **OCDATA** command)
D_S = significant wave height (input directly as *HS* on the **OCTABLE** command or calculated internally from the spectrum data on the **OCTABLE** command)

The particle accelerations are calculated by differentiating $\overrightarrow{v_R}$ and $\overrightarrow{v_Z}$ with respect to time.

12.8.2.2. Shell New Wave

The Shell new wave model is similar to random wave. It uses a statistically based linear superposition of linear wave components to define the wave profile and associated kinematics representing the most likely maximum condition of a real sea.

The free surface of the wave can again be defined by

$$\eta_s = \sum_{i=1}^{N_W} \eta_i = \sum_{i=1}^{N_W} \frac{H_i}{2} \cos\beta_i$$

but with revised definitions for some parameters, as follows:

N_W = Number of wave components (input as *NWC* on the **OCTABLE** command)

H_i = surface coefficient = $2A_C\left(\frac{S(\omega_i)\Delta\omega_i}{m_0}\right)$ where A_C is the wave crest amplitude (input as *AMPMAX* on the **OCDATA** command), $S_{(\omega i)}$ is value of the spectral density at angular frequency ωi, $\Delta\omega i$ is the frequency band which is assumed constant for this wave model, and m_o is the area under the full spectrum

$\beta_i = 2\pi\left(\frac{(R_i - R_0)}{\lambda_i} - \frac{(t - t_0)}{\tau_i}\right)$ for evolving wave (input as *EVOLVING* on the **OCDATA** command), or

$2\pi\left(\frac{(R_i - R_0)}{\lambda_i} - \frac{c_Z}{\lambda_i}(t - t_0)\right)$ for frozen wave

λ_i = wave length derived from Equation 12.24 (p. 397)

τ_i = wave period = $\frac{2\pi}{\omega_i}$, where ω_i is the angular frequency for wave component *i*

R_0 = position offset in the direction of the wave, relative to the structural origin, where the maximum wave crest occurs (input as *ROFF* on the **OCDATA** command)

t_0 = time at which the maximum wave crest occurs (input as *TOFF* on the **OCDATA** command)

c_Z = wave celerity (= λ/τ) associated with a wave having a wave period of T_Z, where T_Z is the zero-crossing period of the wave spectrum that can be computed from the spectral peak period.

The radial and vertical fluid particle velocities are given by Equation 12.33 (p. 400) and Equation 12.34 (p. 400) except that the wave kinematics factor f_K is now computed from:

$$\sqrt{(1-\alpha^2+\alpha^4)}$$

where:

α = wave-spreading angle in radians (input as *SPANGLE* on the **OCDATA** command)

Delta stretching is available for Shell new wave as described in Random Wave (p. 399). All definitions are similar except that D_s in Equation 12.35 (p. 401) is set to A_C here.

12.8.2.3. Constrained New Wave

Constrained new wave theory embeds a Shell new wave into a random wave so that the maximum crest amplitude as given by the new wave occurs at a specific time and position while the statistical nature of the random sea is preserved.

The free surface of a constrained new wave is given by:

$$\begin{aligned}\eta_s &= \eta_r(R,t) + (A_C - \eta_r(R_0,t_0))\eta_n(R,t) - (\dot{\eta}_r(R_0,t_0)/\ddot{\eta}_n(R_0,t_0))\dot{\eta}_n(R,t) \\ &= \eta_r(R,t) + K_1\eta_n(R,t) + K_2\dot{\eta}_n(R,t)\end{aligned} \tag{12.36}$$

where:

$\eta_r(R,t) = \sum_{i=1}^{N_w} \frac{H_i}{2}\cos\{2\pi\left(\frac{R}{\lambda_i} - \frac{t}{\tau_i} + \frac{\varphi_i}{360}\right)\}$ = surface elevation equation of a random wave as previously defined.

$\eta_n(R,t) = \sum_{i=1}^{N_w} \frac{H_i^n}{2}\cos\{2\pi\left(\frac{(R - R_0)}{\lambda_i} - \frac{(t - t_0)}{\tau_i}\right)\}$ = surface elevation equation of a new wave that yields unit crest amplitude at time t_0 and position R_0 (input as *TOFF* and *ROFF*, respectively, on the **OCDATA** command.

A_C = desired constrained wave crest amplitude (input as *AMPCONST* on the **OCDATA** command)

A dot or dots over a variable represent the usual time derivative(s) of the variable.

Equation 12.36 (p. 402) is set up such that it satisfies the conditions of:

$$\eta_s(R_0, t_0) = A_C$$

$$\dot{\eta}_s(R_0, t_0) = 0$$

Therefore, the desired crest amplitude is produced at (R_0, t_0) and the corresponding slope of the free surface is zero.

The height of the new wave components H^n_i is related to the random wave component height H_i. For the equal energy packets that are assumed in the implementation, the expression is simplified to:

$$H_i^n = \frac{2}{N_W}$$

The second and third terms in Equation 12.36 (p. 402) are related to new wave and can be further manipulated to a more familiar form for linear waves:

$$K_1\eta_n(R,t) + K_2\dot{\eta}_n(R,t) = \sum_{i=1}^{N_W} \frac{B_i^n}{2}\cos\{2\pi(\frac{(R-R_0)}{\lambda_i} - \frac{(t-t_0)}{\tau_i} + \frac{\delta_i}{360})\}$$

where:

$$B_i = \sqrt{K_1^2 + (K_2\omega_i)^2}H_i^n$$

$$\delta_i = -\tan^{-1}(\frac{K_2\omega_i}{K_1})$$

A constrained new wave can thus be considered to have two sets of superimposed random waves together; therefore, the wave particle kinematics can also be calculated by applying the formula derived for random waves. See Random Wave (p. 399) and refer to Equation 12.33 (p. 400) through Equation 12.35 (p. 401) for details.

12.8.3. Diffracted Wave on Line and Surface Elements (K_w = 8)

Hydrodynamic pressures and motions obtained from a frequency-based hydrodynamic diffraction analysis can be imported for further analysis. This feature is enabled by specifying a diffracted wave type (K_w = 8, input as *KWAVE* on the **OCDATA** command). The hydrodynamic data are imported via the **OCREAD** command.

Loading can be applied to a line or surface. Both cases are described in the following topics:

12.8.3.1. Diffracted Waves on Line Elements
12.8.3.2. Diffracted Waves on Surface Elements

12.8.3.1. Diffracted Waves on Line Elements

Hydrodynamic loads on structural line elements are calculated based on the results obtained for hydrodynamic line elements.

For hydrodynamic line elements, the wave data consist of the magnitudes and phase angles of the wave free surface, dynamic wave pressure, and velocities and accelerations at both ends of the line elements at a number of defined wave periods and directions.

If the required wave direction θ (input as *THETA* on the **OCDATA** command) and/or wave frequency ω_i ($2\pi/\tau$(i), where τ(i) is the input wave period on the **OCTABLE** command) differs from the hydrodynamic data values, the wave information is obtained via linear interpolations between data in neighboring hydrodynamic analyzed directions and frequencies.

The position of each structural integration point is mapped to the hydrodynamic mesh and the hydrodynamic line element containing this point is determined. The wave data at this location are interpolated linearly from the nodal values of the hydrodynamic line element. The mapping operation assumes that the hydrodynamic mesh is fixed in the original configuration; therefore, structural displacements must be small for this procedure to work accurately.

Finally, the wave free surface at the specified phase angle ϕ_i (input as ϕ(i) on the **OCTABLE** command) is given by:

$$\eta_s = \sum_{i=1}^{N_w} \frac{H_i}{2} (\eta_{real}(\theta, \omega_i) \cos\varphi_i + \eta_{imag}(\theta, \omega_i) \sin\varphi_i) \tag{12.37}$$

where:

H_i = height of component i (input as A(i) on the **OCTABLE** command)
$\eta_{real}(\theta,\omega_i)$ = real component of the surface elevation obtained from the hydrodynamic data
$\eta_{imag}(\theta,\omega_i)$ = imaginary component of the surface elevation obtained from the hydrodynamic data

The wave height is required because the hydrodynamic diffraction results are for unit wave amplitude and must be scaled to the required height.

The fluid particle velocities and accelerations are obtained in a similar fashion (that is, with η replaced by velocity V or acceleration V in Equation 12.37 (p. 404). Current effects, if specified, are added to the fluid velocities and Morison's equation is then applied to calculate the hydrodynamic loads.

12.8.3.2. Diffracted Waves on Surface Elements

Hydrodynamic loading acting on a surface is applied by mapping the pressure results on the hydrodynamic panel elements onto the structural SURF154 elements. Hydrodynamic loading is introduced by setting KEYOPT(8) = *OCID* in the SURF154 element, where *OCID* is the ocean load ID containing the hydrodynamic data.

For hydrodynamic panel elements, the wave data are the magnitudes and phase angles of the hydrodynamic pressure at the centroids of the panels at a number of defined wave periods and directions. If the required wave direction θ (input as *THETA* on the **OCDATA** command) and/or wave frequency ω_i ($2\pi/\tau$(i), where τ(i) is the input wave period on the **OCTABLE** command) differs from the hydrodynamic data values, the wave information is obtained by linear interpolations between data in neighboring hydrodynamic analyzed directions and frequencies.

The position of each structural integration point is mapped to the hydrodynamic mesh and the hydrodynamic panel element containing this point is determined. The real and imaginary wave pressures at this location are interpolated linearly from the nodal values of the hydrodynamic panel. Because the

pressure data are given at the centroidal position, it is necessary to first calculate the element nodal values from the centroidal values surrounding each node. The mapping operation assumes that the hydrodynamic mesh is fixed in the original configuration; therefore, structural displacements must be small for this procedure to work accurately.

Similar to Equation 12.37 (p. 404), the hydrodynamic pressure at the specified phase angle ϕ_i (input as ϕ(i) on the **OCTABLE** command) is given by:

$$P = \sum_{i=1}^{N_W} \frac{H_i}{2} (P_{real}(\theta, \omega_i) \cos \varphi_i + P_{imag}(\theta, \omega_i) \sin \varphi_i) \tag{12.38}$$

where:

H_i = height of component i (input as A(i) on the **OCTABLE** command)
$P_{real}(\theta,\omega_i)$ = real component of the pressure obtained from the hydrodynamic data
$P_{imag}(\theta,\omega_i)$ = imaginary component of the pressure obtained from the hydrodynamic data

The pressure from Equation 12.38 (p. 405) is added to other pressure (PRES) loads applied to the SURF154 element. Current is not considered in the pressure calculations, and static pressure is also excluded.

12.8.4. Presence of Both Waves and Current

If both waves and current are present, the question of wave-current interaction must be considered. Several options are made available via K_{cr} (input as *KCRC* on the **OCDATA** command):

- For K_{cr} = 0, the current velocity at all points above the mean sea level is simply set equal to W_o, where W_o is the input current velocity at Z = 0.0. All points below the mean sea level have velocities selected as though no wave exists.
- For K_{cr} = 1, the current velocity profile is "stretched" or "compressed" to fit the wave. In equation form, the Z coordinate location of current measurement is adjusted by

$$Z'(j) = Z(j) \frac{d + \eta_s}{d} + \eta_s \tag{12.39}$$

where:

Z(j) = Z coordinate location of current measurement (input as Z(j))
$Z'(j)$ = adjusted value of Z(j)

For K_{cr} = 2, the same adjustment as for K_{cr} = 1 is used, as well as a second change that accounts for "continuity." That is,

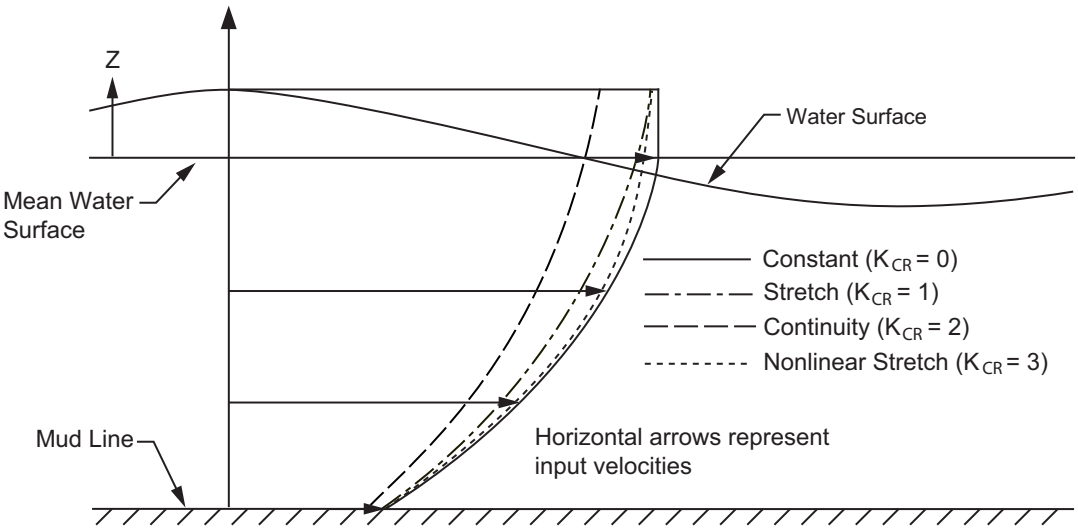
$$W'(j) = W(j) \frac{d}{d + \eta_s} \tag{12.40}$$

where:

W(j) = velocity of current at this location (input as W(j))
$W'(j)$ = adjusted value of W(j)

The velocity options are shown pictorially in Figure 12.29: Velocity Profiles for Wave-Current Interactions (p. 406).

Figure 12.29: Velocity Profiles for Wave-Current Interactions



An additional K_{cr} option is available for K*w* = 5, 6 or 7. For K_{cr} = 3, the current velocity is computed by a nonlinear stretching algorithm as recommended by the API RP2A codes of practice. The relationship is given by:

$$Z = Z' + \eta_s \frac{\sinh(\frac{2\pi}{\lambda_p}(Z'+d))}{\sinh(\frac{2\pi}{\lambda_p}d)}$$

where λ_p is the wave length of the wave components corresponding to the peak spectral period.

In the presence of current with wave, some codes of practice require the use of apparent wave period in the computation of wave kinematics. The apparent period can be calculated iteratively from:

$$\frac{\lambda}{T} = \frac{\lambda}{T_{app}} + V_I \tag{12.41}$$

$$gT_{app}^2 = \frac{2\pi\lambda}{\tanh(2\pi d/\lambda)} \tag{12.42}$$

where:

T = wave period as seen by a stationary observer
T_{app} = wave period as seen by an observer moving at velocity V_I
V_I = effective in-line current speed as recommended by the American Petroleum Institute ([396] (p. 943)): $\frac{4\pi/\lambda}{\sinh(4\pi d/\lambda)}\int_{-d}^{0} U_c(z)\cosh[\frac{4\pi(z+d)}{\lambda}]dz$, where $U_c(z)$ is the component of steady current profile at elevation z in the wave direction.

The apparent period calculation is available for K_w = 5, 6 or 7. The calculation is activated by setting *APC* = 1 on the **OCDATA** command

12.8.5. MacCamy-Fuchs Corrections

When K_w = 0 through 3, the MacCamy-Fuchs adjustment can be requested (**OCDATA**) to account for diffraction effects, especially for large-diameter objects with shorter wave lengths.

When the MacCamy-Fuchs option is applied, the following two adjustments occur:

- The coefficient of inertia is adjusted:

$$C'_m = C_m \frac{\frac{2}{\pi x^2}}{\sqrt{\left[J'_1(x)\right]^2 + \left[Y'_1(x)\right]^2}}$$

where:

$$x = \frac{\pi D_e}{\lambda_1}$$

$$J'_1(x) = J_o(x) - \frac{J_1(x)}{x}$$

$$Y'_1(x) = Y_o(x) - \frac{Y_1(x)}{x}$$

J_0 = zero order Bessel function of the first kind
J_1 = first-order Bessel function of the first kind
Y_0 = zero order Bessel function of the second kind
Y_1 = first-order Bessel function of the second kind

- The phase shift is added to φ_i (before the wave-location correction [input via *WAVELOC* on the **OCDATA** command], if used):

$$\varphi'_i = \varphi_i + \arctan\frac{J'(x)}{Y'(x)}$$

12.8.6. Morison's Equation

For line elements, the hydrodynamic loads are calculated using Morison's equation for a moving body:

$$\{F / L\} = \rho_w A\{\ddot{v}\} + C_a \rho_w A\{\ddot{v} - \ddot{u}\} + \frac{1}{2}\rho_w C_D D_e \left|\{\dot{v} - \dot{u}\}\right| \{\dot{v} - \dot{u}\} \tag{12.43}$$

where:

{F/L} = vector of loads per-unit-length due to hydrodynamic effects
C_D = drag coefficient
ρ_w = water density (mass/length^3)
D_e = outer pipe diameter when using PIPE288 or PIPE289. For BEAM188 or BEAM189, D_e is based on the average of the effective height and width (y or z,

respectively) (output via the **SLIST** command). This value can be augmented by the thicknesses of insulation and biofouling.

A = cross-sectional area of the body (length^2). For PIPE288 or PIPE289, this value is $\frac{\pi D_e^2}{4}$. For BEAM188 or BEAM189, A is the displaced fluid area (output via the **SLIST** command). This value can be augmented by the thickness of insulation.

C_a = added mass coefficient

$\dot{u}, \ddot{u}$ = velocity, acceleration of the structure (u = solved for unknown)

$\dot{v}, \ddot{v}$ = velocity, acceleration of water particle

The first term in Equation 12.43 (p. 407) is the Froude-Krylov force, the second term is the hydrodynamic mass force, and the third term is the drag force. The Froude-Krylov force and hydrodynamic mass force can be rearranged for nonrelative accelerations by using the relationship between C_a and C_M:

$$\{F / L\} = \rho_w C_a A\{-\ddot{u}\} + \rho_w C_M A\{\ddot{v}\} + \frac{1}{2}\rho_w C_D D_e \left|\{\dot{v} - \dot{u}\}\right| \{\dot{v} - \dot{u}\} \tag{12.44}$$

where C_M = inertia coefficient (= C_a +1)

The equations of motion for a submerged structure are expressed as:

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \rho_w C_a A\{-\ddot{u}\} + \rho_w C_M A\{\ddot{v}\} + \frac{1}{2}\rho_w C_D D_e \left|\{\dot{v} - \dot{u}\}\right| \{\dot{v} - \dot{u}\} \tag{12.45}$$

where [M], [C], [K] = structural mass, damping, and stiffness matrix per-unit-length, respectively.

The first term on the right side of Equation 12.45 (p. 408) can be added to the mass matrix, as it is a function of structural acceleration ($\ddot{u}$) only.

$$[M + Ma]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \rho_w C_M A\{\ddot{v}\} + \frac{1}{2}\rho_w C_D D_e \left|\{\dot{v} - \dot{u}\}\right| \{\dot{v} - \dot{u}\} \tag{12.46}$$

where [Ma] = added mass matrix $\left(= f\left(\rho_w C_a V\right)\right)$ per-unit-length, and C_a is input as *Cay* and *Caz* (**OCDATA**).

Finally, the right side of Equation 12.46 (p. 408) forms the hydrodynamic force vector per-unit-length $\{F_{hd}/L\}$. The force consists of two normal components and one tangential component in the element coordinate system. The tangential force includes the drag force only, as no inertial effect exists in that direction.

$$\begin{aligned}
\{F_{hd}/L\}_y &= \frac{1}{2}\rho_w C_{Dy} D_{ey} \left|\{\dot{v}_y - \dot{u}_y\}\right| \{\dot{v}_y - \dot{u}_y\} + \rho_w C_{My} A\{\ddot{v}_y\} \\
\{F_{hd}/L\}_z &= \frac{1}{2}\rho_w C_{Dz} D_{ez} \left|\{\dot{v}_z - \dot{u}_z\}\right| \{\dot{v}_z - \dot{u}_z\} + \rho_w C_{Mz} A\{\ddot{v}_z\} \\
\{F_{hd}/L\}_x &= \frac{1}{2}\rho_w C_T D_e \left|\{\dot{v}_x - \dot{u}_x\}\right| \{\dot{v}_x - \dot{u}_x\}
\end{aligned} \tag{12.47}$$

where:

$\{F_{hd}/L\}_{x,y,z}$ = load vector per-unit-length due to hydrodynamic effects in the element x, y, and z directions
C_{Dy}, C_{Dz}= coefficient of normal drag (input as *CDY* and *CDZ* [**OCTABLE**])
D_{ey} and D_{ez} = outer pipe diameter when using PIPE288 or PIPE289. For BEAM188 or BEAM189, this value is based on either the effective height or width (y or z, respectively) (output via the **SLIST** command).
C_{My}, C_{Mz} = coefficient of inertia in the element normal (y and z) direction (input as *CMY* and *CMZ* [**OCTABLE**])
$\{\ddot{v}_{y,z}\}$ = normal particle acceleration vector (length/time^2)
=
C_T = coefficient of tangential (x) drag (input as *CT* [**OCTABLE**])
$\{\dot{v}_{x,y,z}\}$ = tangential (x) and normal (y,z) particle velocity vector (length/time)
$\{\dot{u}_{x,y,z}\}$ = tangential (x) and normal (y,z) structure velocity vector (length/time)

To calculate the relative velocities and accelerations, the velocity and acceleration of the fluid particle and the structure must be available so that one can be subtracted from the other. The fluid particle velocity and acceleration are computed using relationships such as Equation 12.26 (p. 398) through Equation 12.29 (p. 398), as well as current effects. The structure velocity is available via the Newmark time integration logic. (See Transient Analysis (p. 763).)

The integration points along the length of the element match the underlying element and are used to generate the load vector. Integration points below the mud line are simply bypassed. For elements intersecting the free surface, the integration points are distributed along the wet length only.

The hydrodynamic coefficients (C_D, C_T, C_M) may be defined as fixed numbers (**OCTABLE**), or as function of Z level or Reynolds number (via multiple **OCTABLE** commands).

The definitions of Reynolds number can be expressed in the element coordinate system as:

$$Re_n = \{\dot{v}_n\}\frac{\rho_w D_e}{\mu}, \qquad Re_t = \{\dot{v}_t\}\frac{\rho_w D_e}{\mu}$$

where:

Re_n, Re_t = normal and tangential Reynolds number
$\{\dot{v}_n\},\{\dot{v}_t\}$ = vector of normal and tangential particle velocity
μ = viscosity (input as *VISC* [**MP**])

Temperature-dependent effects can be included via the **MP** command, or by specifying the temperature (*Te*) on the **OCTABLE** command with ocean current input (**OCTYPE**,CURR).

12.9. Nodal and Centroidal Data Evaluation

Area and volume elements normally compute results most accurately at the integration points. The location of these data, which includes structural stresses, elastic and thermal strains, field gradients, and fluxes, can then be moved to nodal or centroidal locations for further study. This is done with ex-

trapolation or interpolation, based on the element shape functions or simplified shape functions given in Table 12.13: Assumed Data Variation of Stresses (p. 410).

Table 12.13: Assumed Data Variation of Stresses

Geometry	No. Integration Points	Assumed Data Variation
Triangles	3	a + bs + ct
Quadrilaterals	4	a + bs + ct + dst
Tetrahedra	4	a + bs + ct + dr
Hexahedra	8	a + bs + ct + dr + est + ftr + gsr + hstr

where:

a, b, c, d, e, f, g, h = coefficients
s, t, r = element natural coordinates

The extrapolation is done or the integration point results are simply moved to the nodes, based on the user's request (input on the **ERESX** command). If material nonlinearities exist in an element, the least squares fit can cause inaccuracies in the extrapolated nodal data or interpolated centroidal data. These inaccuracies are normally minor for plasticity, creep, or swelling, but are more pronounced in elements where an integration point may change status, such as SHELL41, SOLID65, etc.

Adjustments and special cases exist, as follows:

- SOLID90 uses only the eight corner integration points.
- Uniform stress cases, like a constant stress triangle, do not require the above processing.

Chapter 13: Element Library

This chapter describes the theory underlying each ANSYS element. The explanations are augmented by references to other sections in this manual as well as external sources.

The table below the introductory figure of each element is complete, except that the Newton-Raphson load vector is omitted. This load vector always uses the same shape functions and integration points as the applicable stiffness, conductivity and/or coefficient matrix. Exceptions associated mostly with some nonlinear line elements are noted with the element description.

13.1. Reserved for Future Use

This section is reserved for future use.

13.2. Not Documented

No detail or element available at this time.

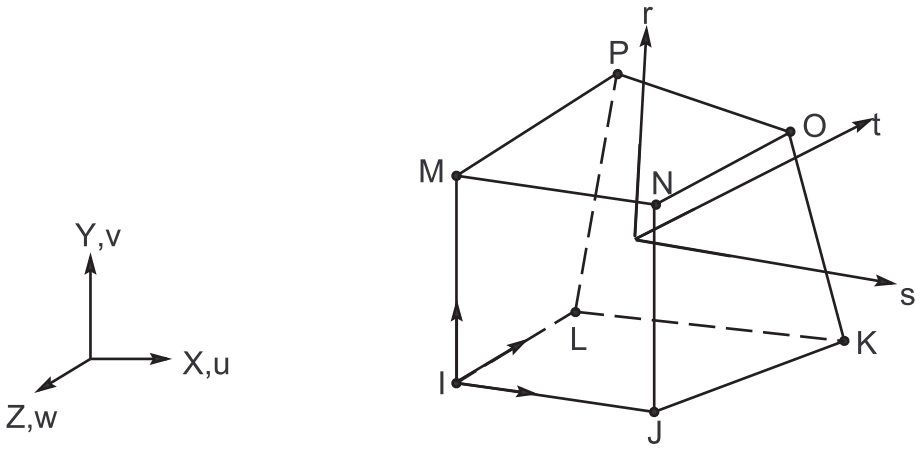
13.3. Reserved for Future Use

This section is reserved for future use.

13.4. Reserved for Future Use

This section is reserved for future use.

13.5. SOLID5 - 3-D Coupled-Field Solid



Matrix or Vector	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix	Equation 11.226 (p. 353)	2 x 2 x 2

Matrix or Vector	Shape Functions	Integration Points
Electrical Conductivity Matrix	Equation 11.225 (p. 353)	2 x 2 x 2
Thermal Conductivity Matrix	Equation 11.224 (p. 353)	2 x 2 x 2
Stiffness Matrix and Thermal Expansion Load Vector	Equation 11.212 (p. 353), Equation 11.213 (p. 353), and Equation 11.214 (p. 353) or, if modified extra shapes are included (KEYOPT(3) = 0), Equation 11.227 (p. 354), Equation 11.228 (p. 354), and Equation 11.229 (p. 354)	2 x 2 x 2
Piezoelectric Coupling Matrix	Same as combination of stiffness matrix and conductivity matrix.	2 x 2 x 2
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in 3-D Lines (p. 329)	2 x 2 x 2
Mass and Stress Stiffening Matrices	Equation 11.212 (p. 353), Equation 11.213 (p. 353), and Equation 11.214 (p. 353)	2 x 2 x 2
Load Vector due to Imposed Thermal and Electric Gradients, Heat Generation, Joule Heating, Magnetic Forces, Magnetism due to Source Currents and Permanent Magnets	Same as coefficient or conductivity matrix	2 x 2 x 2
Load Vector due to Convection Surfaces and Pressures	Same as stiffness or conductivity matrix specialized to the surface.	2 x 2 x 2

References: Wilson([38] (p. 922)), Taylor([49] (p. 923)), Coulomb([76] (p. 924)), Mayergoyz([119] (p. 927)), Gyimesi([141] (p. 928),[149] (p. 929))

13.5.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. Heat Flow (p. 227) describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. Derivation of Electromagnetic Matrices (p. 193) discusses the scalar potential method, which is used by this element. Piezoelectrics (p. 313) discusses the piezoelectric capability used by the element.

13.6. Reserved for Future Use

This section is reserved for future use.

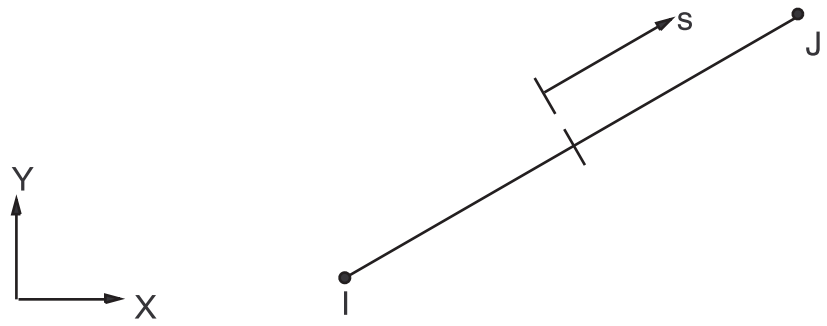
13.7. Reserved for Future Use

This section is reserved for future use.

13.8. Reserved for Future Use

This section is reserved for future use.

13.9. INFIN9 - 2-D Infinite Boundary



Matrix or Vector	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix or Thermal Conductivity Matrix	$A = C_1 + C_2x$	None

References: Kagawa, Yamabuchi and Kitagami([122] (p. 927))

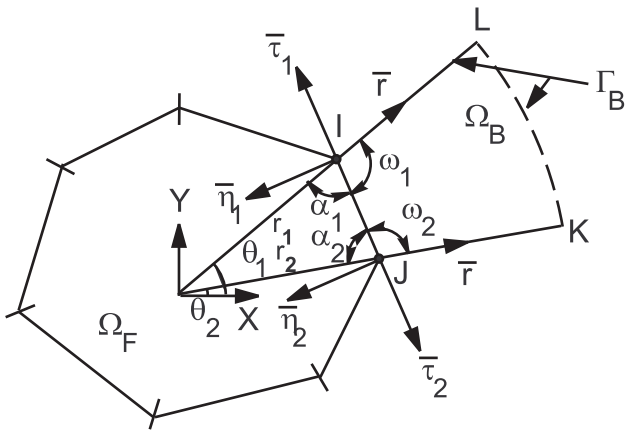
13.9.1. Introduction

This boundary element (BE) models the exterior infinite domain of the far-field magnetic and thermal problems. This element is to be used in combination with elements having a magnetic potential (AZ) or temperature (TEMP) as the DOF.

13.9.2. Theory

The formulation of this element is based on a first order infinite boundary element (IBE) that is compatible with first order quadrilateral or triangular shaped finite elements, or higher order elements with dropped midside nodes. For unbounded field problems, the model domain is set up to consist of an interior finite element domain, Ω_F, and a series of exterior BE subdomains, Ω_B, as shown in Figure 13.1: Definition of BE Subdomain and the Characteristics of the IBE (p. 414). Each subdomain, Ω_B, is treated as an ordinary BE domain consisting of four segments: the boundary element I-J, infinite elements J-K and I-L, and element K-L; element K-L is assumed to be located at infinity.

Figure 13.1: Definition of BE Subdomain and the Characteristics of the IBE



The approach used here is to write BE equations for Ω_B, and then convert them into equivalent load vectors for the nodes I and J. The procedure consists of four separate steps that are summarized below (see reference ([122] (p. 927)) for details).

First, a set of boundary integral equations is written for Ω_B. To achieve this, linear shape functions are used for the BE I-J:

$$N_1(s) = \frac{1}{2}(1-s) \tag{13.1}$$

$$N_2(s) = \frac{1}{2}(1+s) \tag{13.2}$$

Over the infinite elements J-K and I-L the potential (or temperature) ϕ and its derivative q (flux) are respectively assumed to be:

$$\phi(r) = \phi_i\left(\frac{r_i}{r}\right), \qquad i = I,J \tag{13.3}$$

$$q(r) = q_i\left(\frac{r_i}{r}\right)^2, \qquad i = I,J \tag{13.4}$$

The boundary integral equations are the same as presented in Equation 13.102 (p. 456) except that the Green's function in this case would be:

$$G(x,\xi) = \frac{1}{2\pi k}\ln\left(\frac{\sqrt{k}}{r}\right) \tag{13.5}$$

where:

x = field point in boundary element
ξ = source point

$$k = \begin{cases} \text{magnetic reluctivity (inverse of free space permeability input on } \textbf{EMUNIT} \text{ command) for AZ DOF (KEYOPT(1) = 0)} \\ \text{or} \\ \text{thermal conductivity (input as KXX on } \textbf{MP} \text{ command) for TEMPDOF (KEYOPT(1) = 1)} \end{cases}$$

Note that all the integrations in the present case are performed in closed form.

Second, in the absence of a source or sink in Ω_B, the flux q(r) is integrated over the boundary Γ_B of Ω_B and set to zero.

$$\int_{\Gamma_B} q d\Gamma = 0 \tag{13.6}$$

Third, a geometric constraint condition that exists between the potential ϕ and its derivatives $\frac{\partial \phi}{\partial n}$ and $\frac{\partial \phi}{\partial \tau} = q_\tau$ at the nodes I and J is written as:

$$q_{n_i} = q_{\tau_i} \cos\alpha_i + \phi_i \frac{\sin\alpha_i}{r_i} \qquad i = I,J \tag{13.7}$$

Fourth, the energy flow quantity from Ω_B is written as:

$$w = \int_{\Gamma_B} q\phi d\Gamma \tag{13.8}$$

This energy flow is equated to that due to an equivalent nodal {F} defined below.

The four steps mentioned above are combined together to yield, after eliminating q_n and q_τ,

$$[K]\{\phi\} = \{F\} \tag{13.9}$$

where:

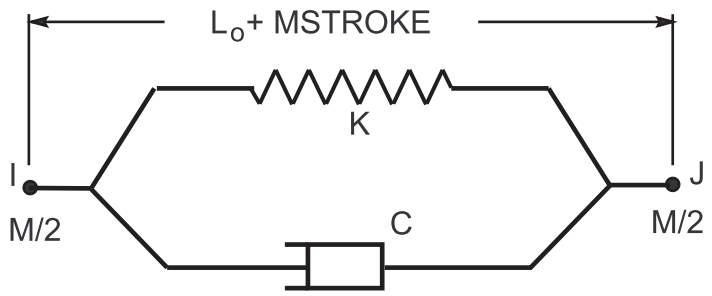
[K] = 2 x 2 equivalent unsymmetric element coefficient matrix
{ϕ} = 2 x 1 nodal DOFs, AZ or TEMP
{F} = 2 x 1 equivalent nodal force vector

For linear problems, the INFIN9 element forms the coefficient matrix [K] only. The load vector {F} is not formed. The coefficient matrix multiplied by the nodal DOF's represents the nodal load vector which brings the effects of the semi-infinite domain Ω_B onto nodes I and J.

13.10. Reserved for Future Use

This section is reserved for future use.

13.11. LINK11 - Linear Actuator



Matrix or Vector	Shape Functions	Integration Points
Stiffness and Damping Matrices	Equation 11.6 (p. 329)	None
Mass Matrix	None (lumped mass formulation)	None
Stress Stiffness Matrix	Equation 11.7 (p. 330) and Equation 11.8 (p. 330)	None

13.11.1. Assumptions and Restrictions

The element is not capable of carrying bending or twist loads. The force is assumed to be constant over the entire element.

13.11.2. Element Matrices and Load Vector

All element matrices and load vectors are described below. They are generated in the element coordinate system and are then converted to the global coordinate system. The element stiffness matrix is:

$$[K_\ell] = K \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{13.10}$$

where:

K = element stiffness (input as K on **R** command)

The element mass matrix is:

$$[M_\ell] = \frac{M}{2} \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{13.11}$$

where:

M = total element mass (input as M on **R** command)

The element damping matrix is:

$$[C_\ell] = C\left[\begin{array}{ccc:ccc} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{13.12}$$

where:

C = element damping (input as C on **R** command)

The element stress stiffness matrix is:

$$[S_\ell] = \frac{F}{L}\left[\begin{array}{ccc:ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{array}\right] \tag{13.13}$$

where:

F = the axial force in the element (output as FORCE)
L = current element length (output as CLENG)

The element load vector is:

$$\{F_\ell\} = \{F_\ell^{ap}\} - \{F_\ell^{nr}\} \tag{13.14}$$

where:

$\{F_\ell^{ap}\}$ = applied force vector

$\{F_\ell^{nr}\}$ = Newton-Raphson restoring force, if applicable

The applied force vector is:

$$\{F_\ell^{ap}\} = F'\left\lfloor -1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \right\rfloor^T \tag{13.15}$$

where:

F' = applied force thru surface load input using the PRES label

The Newton-Raphson restoring force vector is:

$$\{F_\ell^{nr}\} = F\left\lfloor -1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \right\rfloor^T \tag{13.16}$$

13.11.3. Force, Stroke, and Length

The element spring force is determined from

$$F = K(S_M - S_A) \quad (13.17)$$

where:

F = element spring force (output as FORCE)
S_A = applied stroke (output as STROKE) thru surface load input using the PRES label
S_M = computed or measured stroke (output as MSTROKE)

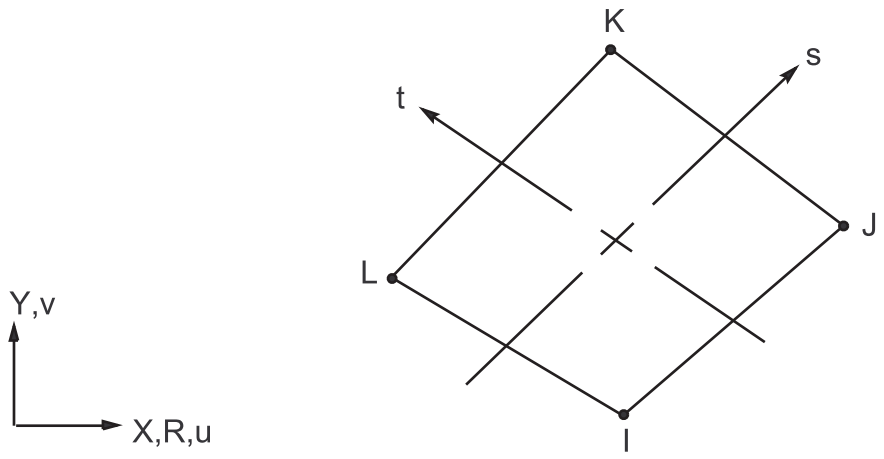
The lengths, shown in the figure at the beginning of this section, are:

L_o = initial length (output as ILEN)
$L_o + S_M$ = current length (output as CLEN)

13.12. Reserved for Future Use

This section is reserved for future use.

13.13. PLANE13 - 2-D Coupled-Field Solid



Matrix or Vector	Geo-metry	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix; and Permanent Magnet and Applied Current Load Vector	Quad	Equation 11.123 (p. 341)	2 x 2
	Triangle	Equation 11.103 (p. 340)	1 if planar 3 if axisymmetric
Thermal Conductivity Matrix	Quad	Equation 11.128 (p. 341)	Same as coefficient matrix
	Triangle	Equation 11.108 (p. 340)	
Stiffness Matrix; and Thermal and Magnetic Force Load Vector	Quad	Equation 11.120 (p. 341) and Equation 11.121 (p. 341) and, if modified extra shapes are included (KEY-	Same as coefficient matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
		OPT(2) = 0) and element has 4 unique nodes) Equation 11.132 (p. 342) and Equation 11.133 (p. 342).	
	Triangle	Equation 11.100 (p. 340) and Equation 11.101 (p. 340)	
Mass and Stress Stiffness Matrices	Quad	Equation 11.120 (p. 341) and Equation 11.121 (p. 341)	Same as coefficient matrix
	Triangle	Equation 11.100 (p. 340) and Equation 11.101 (p. 340)	
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Lumped Matrices (p. 391)		Same as coefficient matrix
Damping (Eddy Current) Matrix	Quad	Equation 11.123 (p. 341) and Equation 11.129 (p. 341)	Same as coefficient matrix
	Triangle	Equation 11.103 (p. 340) and Equation 11.109 (p. 340)	
Convection Surface Matrix and Load Vector	Same as conductivity matrix, specialized to the surface		2
Pressure Load Vector	Same as mass matrix specialized to the face		2

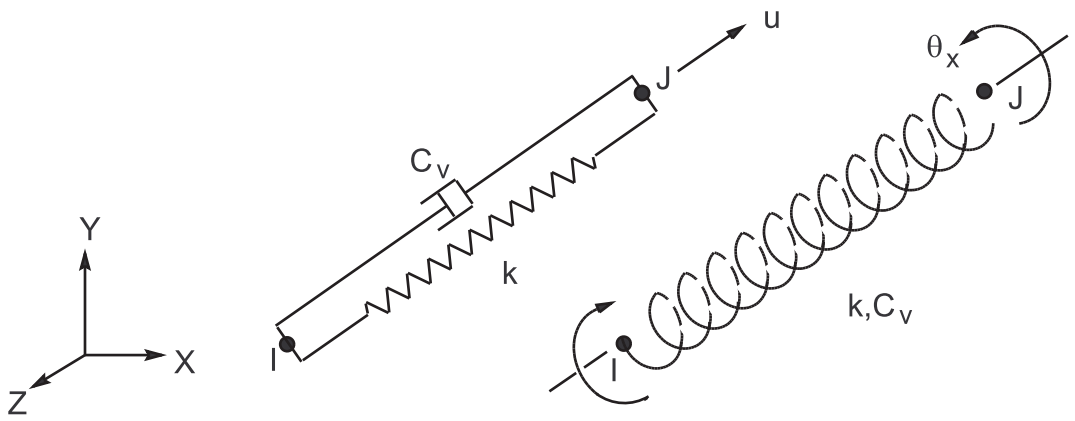
Load Type	Distribution
Current Density	Bilinear across element
Current Phase Angle	Bilinear across element
Heat Generation	Bilinear across element
Pressure	Linear along each face

References: Wilson([38] (p. 922)), Taylor, et al.([49] (p. 923)), Silvester, et al.([72] (p. 924)),Weiss, et al.([94] (p. 925)), Garg, et al.([95] (p. 925))

13.13.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. Heat Flow (p. 227) describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. Derivation of Electromagnetic Matrices (p. 193) and Electromagnetic Field Evaluations (p. 200) discuss the magnetic vector potential method, which is used by this element. The diagonalization of the specific heat matrix is described in Lumped Matrices (p. 391).

13.14. COMBIN14 - Spring-Damper



Matrix or Vector	Option	Shape Functions[1]	Integration Points
Stiffness and Damping Matrices	Longitudinal	Equation 11.6 (p. 329)	None
	Torsional	Equation 11.18 (p. 330)	None
Stress Stiffening Matrix	Longitudinal	Equation 11.7 (p. 330), and Equation 11.8 (p. 330)	None

1. There are no shape functions used if the element is input on a one DOF per node basis (KEYOPT(2) > 0) as the nodes may be coincident.

13.14.1. Types of Input

COMBIN14 essentially offers two types of elements, selected with KEYOPT(2).

1. Single DOF per node (KEYOPT(2) > 0). The orientation is defined by the value of KEYOPT(2) and the two nodes are usually coincident.

2. Multiple DOFs per node (KEYOPT(2) = 0). The orientation is defined by the location of the two nodes; therefore, the two nodes must not be coincident.

13.14.2. Stiffness Pass

Consider the case of a single DOF per node first. The orientation is selected with KEYOPT(2). If KEYOPT(2) = 7 (pressure) or = 8 (temperature), the concept of orientation does not apply. The form of the element stiffness and damping matrices are:

$$[K_e] = k\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.18}$$

$$[C_e] = C_v\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.19}$$

where:

k = stiffness (input as K on **R** command)
$C_v = C_{v1} + C_{v2}\,|v|$

C_{v1} = constant damping coefficient (input as CV1 on **R** command)
C_{v2} = linear damping coefficient (input as CV2 on **R** command)
v = relative velocity between nodes computed from the nodal Newmark velocities

In a full harmonic analysis where the stiffness and constant damping coefficients are input as table parameters, the stiffness (k) and damping (Cv) are interpolated at each frequency step.

Next, consider the case of multiple DOFs per node. Only the case with three DOFs per node will be discussed, as the case with two DOFs per node is simply a subset. The stiffness, damping, and stress stiffness matrices in element coordinates are developed as:

$$[K_\ell] = k\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{13.20}$$

$$[C_\ell] = C_v\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{13.21}$$

$$[S_\ell] = \frac{F}{L}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \tag{13.22}$$

where subscript ℓ refers to element coordinates.

and where:

F = force in element from previous iteration
L = distance between the two nodes

There are some special notes that apply to the torsion case (KEYOPT(3) = 1):

1. Rotations are simply treated as a vector quantity. No other effects (including displacements) are implied.

2. In a large rotation problem (**NLGEOM**,ON), the coordinates do not get updated, as the nodes only rotate. (They may translate on other elements, but this does not affect COMBIN14 with KEYOPT(3) = 1). Therefore, there are no large rotation effects.

3. Similarly, as there is no axial force computed, no stress stiffness matrix is computed.

13.14.3. Output Quantities

The stretch is computed as:

$$\varepsilon_o = \begin{Bmatrix} \frac{A}{L} & \text{if KEYOPT(2) = 0} \\ u'_J - u'_I & \text{if KEYOPT(2) = 1} \\ v'_J - v'_I & \text{if KEYOPT(2) = 2} \\ w'_J - w'_I & \text{if KEYOPT(2) = 3} \\ \theta'_{xJ} - \theta'_{xI} & \text{if KEYOPT(2) = 4} \\ \theta'_{yJ} - \theta'_{yI} & \text{if KEYOPT(2) = 5} \\ \theta'_{zJ} - \theta'_{zI} & \text{if KEYOPT(2) = 6} \\ P_J - P_I & \text{if KEYOPT(2) = 7} \\ T_J - T_I & \text{if KEYOPT(2) = 8} \end{Bmatrix} = \text{output as STRETCH} \tag{13.23}$$

where:

$A = (X_J - X_I)(u_J - u_I) + (Y_J - Y_I)(v_J - v_I) + (Z_J - Z_I)(w_J - w_I)$
X, Y, Z = coordinates in global Cartesian coordinates
u, v, w = displacements in global Cartesian coordinates
u', v', w' = displacements in nodal Cartesian coordinates (UX, UY, UZ)
$\theta'_x, \theta'_y, \theta'_z$ = rotations in nodal Cartesian coordinates (ROTX, ROTY, ROTZ)
P = pressure (PRES)
T = temperatures (TEMP)

If KEYOPT(3) = 1 (torsion), the expression for A has rotation instead of translations, and ε_o is output as TWIST. Next, the static force (or torque) is computed:

$$F_s = k\varepsilon_o \tag{13.24}$$

where:

F_s = static force (or torque) (output as FORC (TORQ if KEYOPT(3) = 1))

Finally, if a nonlinear transient dynamic (**ANTYPE**,TRANS, with **TIMINT**,ON) analysis is performed, a damping force is computed:

$$F_D = C_v v \tag{13.25}$$

where:

F_D = damping force (or torque) (output as DAMPING FORCE (DAMPING TORQUE if KEYOPT(3) = 1))
v = relative velocity

relative velocity is computed using Equation 13.23 (p. 422), where the nodal displacements u, v, w, etc. are replaced with the nodal Newmark velocities $\dot{u}, \dot{v}, \dot{w},$ etc.

13.15. Reserved for Future Use

This section is reserved for future use.

13.16. Reserved for Future Use

This section is reserved for future use.

13.17. Reserved for Future Use

This section is reserved for future use.

13.18. Reserved for Future Use

This section is reserved for future use.

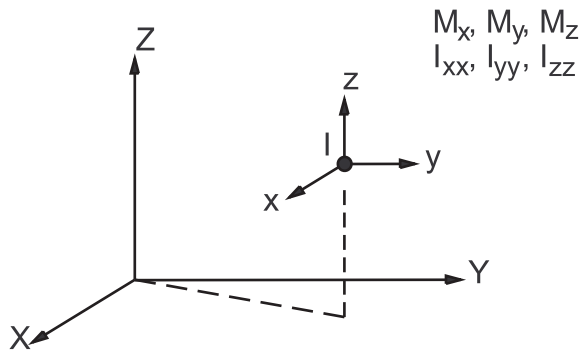
13.19. Reserved for Future Use

This section is reserved for future use.

13.20. Reserved for Future Use

This section is reserved for future use.

13.21. MASS21 - Structural Mass



Matrix or Vector	Shape Functions	Integration Points
Mass Matrix	None	None

The element mass matrix is:

$$[M_e] = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 \\ 0 & b & 0 & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 0 & 0 \\ 0 & 0 & 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 & e & 0 \\ 0 & 0 & 0 & 0 & 0 & f \end{bmatrix} \quad (13.26)$$

where:

$$\begin{Bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{Bmatrix} = \begin{cases} \begin{Bmatrix} a' \\ b' \\ c' \\ d' \\ e' \\ f' \end{Bmatrix} & \text{if KEYOPT(1) = 0} \\ \rho \begin{Bmatrix} a' \\ b' \\ c' \\ d' \\ e' \\ f' \end{Bmatrix} & \text{if KEYOPT(1) = 1} \end{cases}$$

ρ = density (input as DENS on **MP** command)

where a', b', c', d', e', and f' are user input (input on the **R** command) in the locations shown in the following table:

	KEYOPT(3) = 0	KEYOPT(3) = 2	KEYOPT(3) = 3	KEYOPT(3) = 4
a'	1	1	1	1
b'	2	1	1	1
c'	3	1	-	-
d'	4	-	-	-
e'	5	-	-	-
f'	6	-	2	-

For the mass summary, only the first real constant is used, regardless of which option of KEYOPT(3) is used. The precise mass summary used for three-dimensional models as well as analyses with inertial relief use the complete matrix.

13.22. Reserved for Future Use

This section is reserved for future use.

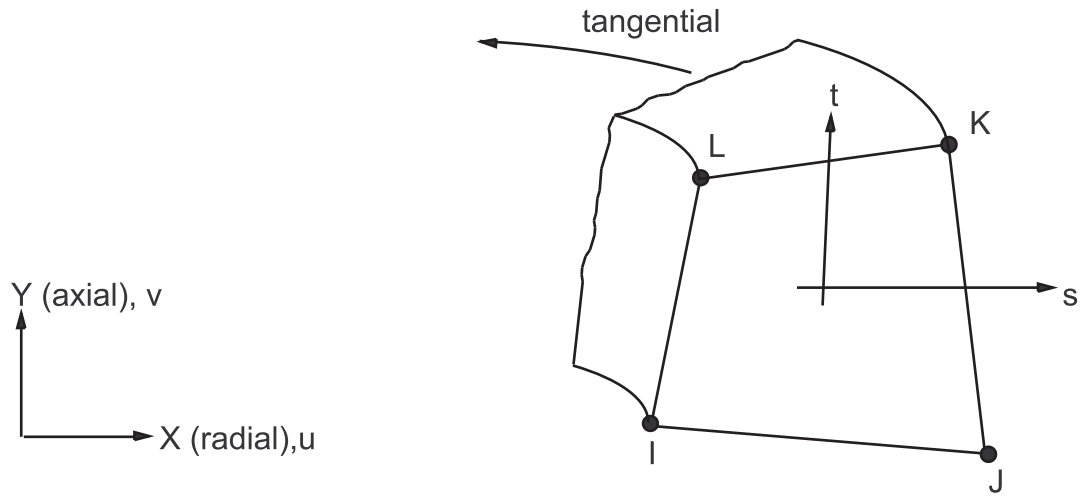
13.23. Reserved for Future Use

This section is reserved for future use.

13.24. Reserved for Future Use

This section is reserved for future use.

13.25. PLANE25 - Axisymmetric-Harmonic 4-Node Structural Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix and Thermal Load Vector	Quad	Equations Equation 11.159 (p. 346), Equation 11.160 (p. 346) , and Equation 11.161 (p. 346) or if modified extra shape functions are included (KEYOPT(2) = 0) and element has 4 unique nodes: Equation 11.163 (p. 346), Equation 11.164 (p. 346) , and Equation 11.165 (p. 346)	2 x 2
	Triangle	Equation 11.151 (p. 345), Equation 11.152 (p. 345) , and Equation 11.153 (p. 345)	3
Mass and Stress Stiffness Matrices	Quad	Equation 11.120 (p. 341), Equation 11.121 (p. 341) , and Equation 11.122 (p. 341)	2 x 2
	Triangle	Equation 11.100 (p. 340), Equation 11.101 (p. 340) , and Equation 11.102 (p. 340)	3
Pressure Load Vector	Same as stress stiffness matrix, specialized to the surface		2

Load Type	Distribution
Element Temperature	Bilinear across element, harmonic around circumference
Nodal Temperature	Bilinear across element, harmonic around circumference
Pressure	Linear along each face, harmonic around circumference

Reference: Wilson([38] (p. 922)), Zienkiewicz([39] (p. 922)), Taylor([49] (p. 923))

13.25.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations.

13.25.2. Assumptions and Restrictions

The material properties are assumed to be constant around the entire circumference, regardless of temperature dependent material properties or loading. For ℓ (input as MODE on **MODE** command) > 0, the extreme values for combined stresses are obtained by computing these stresses at every 10/ℓ degrees and selecting the extreme values.

13.25.3. Use of Temperature

In general, temperatures have two effects on a stress analysis:

1. Temperature dependent material properties.
2. Thermal expansion

In the case of ℓ = 0, there is no conflict between these two effects. However, if ℓ > 0, questions arise. As stated in the assumptions, the material properties may not vary around the circumference, regardless of the temperature. That is, one side cannot be soft and the other side hard. The input temperature for ℓ > 0 varies sinusoidally around the circumference. As no other temperatures are available to the element, the material properties are evaluated at T_{ref} (input on **TREF** command). The input temperature can therefore be used to model thermal bending. An approximate application of this would be a chimney subjected to solar heating on one side only. A variant on this basic procedure is provided by the temperature KEYOPT (KEYOPT(3) for PLANE25). This variant provides that the input temperatures be used only for material property evaluation rather than for thermal bending. This second case requires that α_x, α_y, and α_z (input on **MP** commands) all be input as zero. An application of the latter case is a chimney, which is very hot at the bottom and relatively cool at the top, subjected to a wind load.

13.26. Not Documented

No detail or element available at this time.

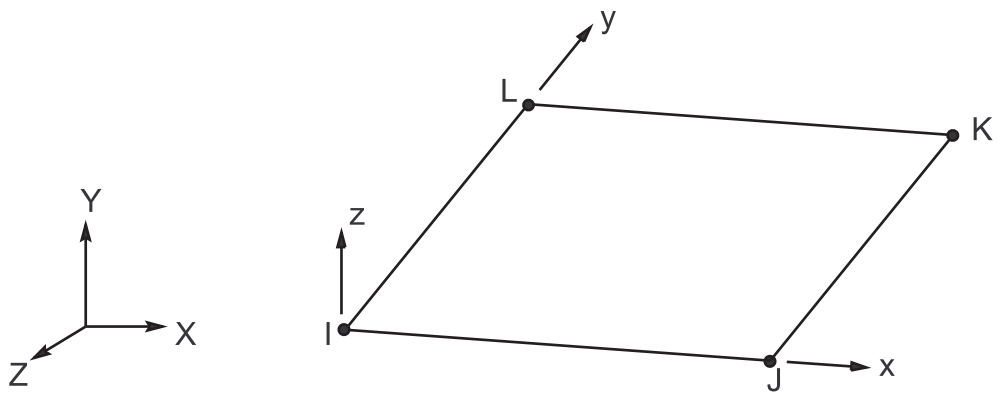
13.27. MATRIX27 - Stiffness, Damping, or Mass Matrix

Matrix or Vector	Shape Functions	Integration Points
Stiffness, Mass, and Damping Matrices	None	None

13.27.1. Assumptions and Restrictions

All MATRIX27 matrices should normally be positive definite or positive semidefinite (see Positive Definite Matrices (p. 390) for definition) in order to be valid structural matrices. The only exception to this occurs when other (positive definite) matrices dominate the involved DOFs and/or sufficient DOFs are removed by way of imposed constraints, so that the total (structure) matrix is positive definite.

13.28. SHELL28 - Shear/Twist Panel



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (see reference)	None
Mass Matrix	None (one-sixth of the mass of each of the IJK, JKL, KLI, and LIJ subtriangles is put at the nodes)	None
Stress Stiffness Matrix	No shape functions are used. Rather, the stress stiffness matrix is developed from the two diagonal forces used as spars	None

Reference: Garvey([116] (p. 927))

13.28.1. Assumptions and Restrictions

This element is based directly on the reference by Garvey([116] (p. 927)). It uses the idea that shear effects can be represented by a uniform shear flow and nodal forces in the direction of the diagonals. The element only resists shear stress; direct stresses will not be resisted.

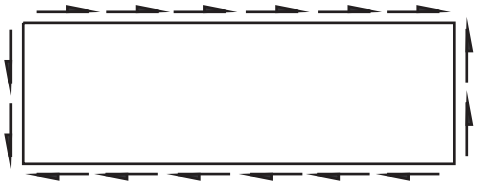
The shear panel assumes that only shearing stresses are present along the element edges. Similarly, the twist panel assumes only twisting moment, and no direct moment.

This element does not generate a consistent mass matrix; only the lumped mass matrix is available.

13.28.2. Commentary

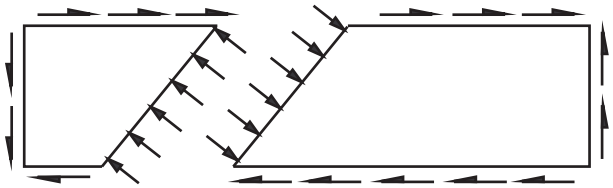
The element loses validity when used in shapes other than rectangular. For non-rectangular cases, the resulting shear stress is nonuniform, so that the patch test cannot be satisfied. Consider a rectangular element under uniform shear:

Figure 13.2: Uniform Shear on Rectangular Element



Then, add a fictional cut at 45° to break the rectangular element into two trapezoidal regions (elements):

Figure 13.3: Uniform Shear on Separated Rectangular Element



As can be seen, shear forces as well as normal forces are required to hold each part of the rectangle in equilibrium for the case of "uniform shear". The above discussion for trapezoids can be extended to parallelograms. If the presumption of uniform shear stress is dropped, it is possible to hold the parts in equilibrium using only shear stresses along all edges of the quadrilateral (the presumption used by Garvey) but a truly uniform shear state will not exist.

13.28.3. Output Terms

The stresses are also computed using the approach of Garvey([116] (p. 927)).

When all four nodes lie in a flat plane, the shear flows are related to the nodal forces by:

$$S_{IJ}^{fl} = \frac{F_{JI} - F_{IJ}}{\ell_{IJ}} \tag{13.27}$$

where:

S_{IJ}^{kl} = shear flow along edge IJ (output as SFLIJ)

F_{JI} = force at node I from node J (output as FJI)

F_{IJ} = force at node J from node I (output as FIJ)

ℓ_{IJ} = length of edge I-J

The forces in the element z direction (output quantities FZI, FZJ, FZK, FZL) are zero for the flat case. When the flat element is also rectangular, all shear flows are the same. The stresses are:

$$\sigma_{xy} = \frac{S_{IJ}^{fl}}{t} \tag{13.28}$$

where:

σ_{xy} = shear stress (output as SXY)

t = thickness (input as THCK on **R** command)

The logic to compute the results for the cases where all four nodes do not lie in a flat plane or the element is non-rectangular is more complicated and is not developed here.

The margin of safety calculation is:

$$M_s = \begin{cases} \dfrac{\sigma_{xy}^u}{\sigma_{xy}^m} - 1.0 \text{ if both } \sigma_{xy}^m \text{ and } \sigma_{xy}^u \neq 0 \\ 0.0 \text{ if either } \sigma_{xy}^m \text{ or } \sigma_{xy}^u = 0 \end{cases} \quad (13.29)$$

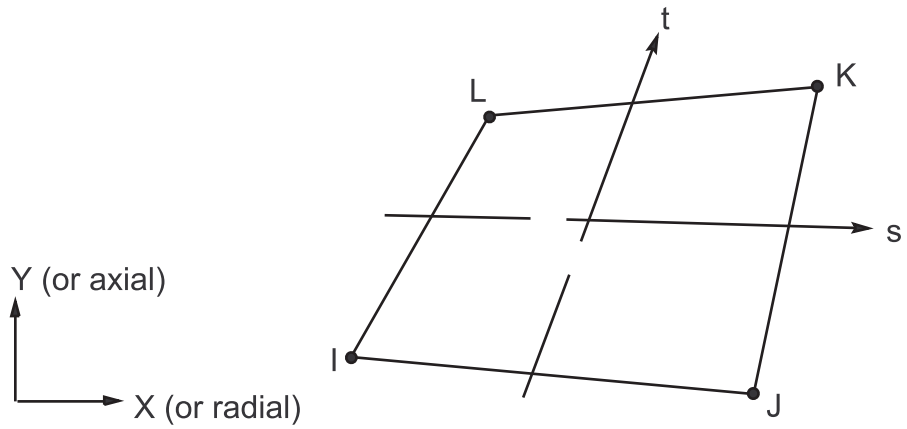
where:

M_s = margin of safety (output as SMARGN)

σ_{xy}^m = maximum nodal shear stress (output as SXY(MAX))

σ_{xy}^u = maximum allowable shear stress (input as SULT on **R** command)

13.29. FLUID29 - 2-D Acoustic Fluid

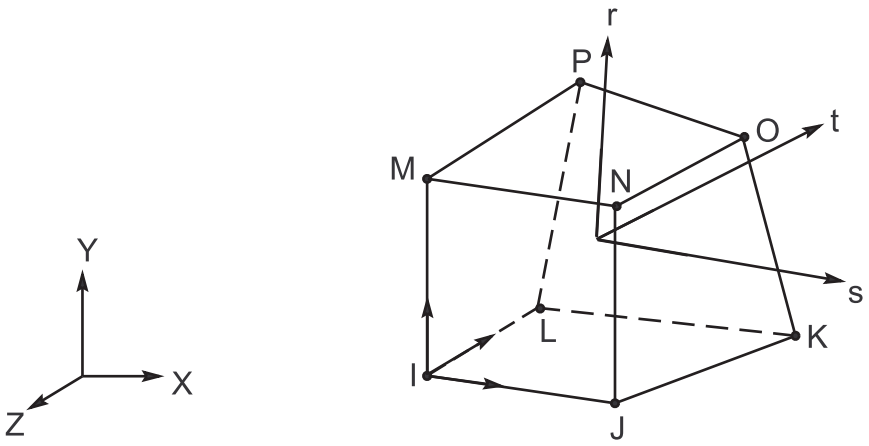


Matrix or Vector	Shape Functions	Integration Points
Fluid Stiffness and Mass Matrices	Equation 11.127 (p. 341)	2 x 2
Coupling Stiffness, Mass, and Damping Matrices (fluid-structure interface)	Equation 11.120 (p. 341), Equation 11.121 (p. 341), and Equation 11.127 (p. 341) specialized to the interface	2

13.29.1. Other Applicable Sections

Acoustics (p. 253) describes the derivation of acoustic element matrices and load vectors.

13.30. FLUID30 - 3-D Acoustic Fluid

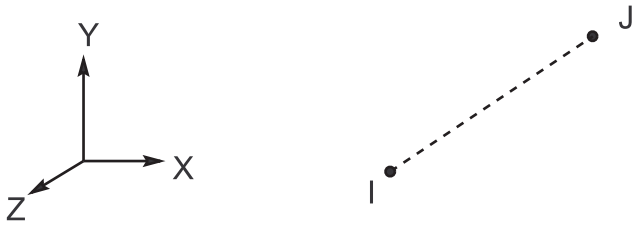


Matrix or Vector	Shape Functions	Integration Points
Fluid Stiffness and Mass Matrices	Equation 11.221 (p. 353)	2 x 2 x 2
Coupling Stiffness and Mass Matrices (fluid-structure interface)	Equation 11.212 (p. 353), Equation 11.213 (p. 353), Equation 11.214 (p. 353), and Equation 11.221 (p. 353) specialized to the interface	2 x 2
Fluid Damping Matrix (fluid at fluid-structure interface)	No shape functions are used. Instead, the area associated with each node at the interface is computed for the damping to act upon.	None

13.30.1. Other Applicable Sections

Acoustics (p. 253) describes the derivation of acoustic element matrices and load vectors.

13.31. LINK31 - Radiation Link



Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix	None (nodes may be coincident)	None

13.31.1. Standard Radiation (KEYOPT(3) = 0)

The two-surface radiation equation (from Equation 6.14 (p. 231)) that is solved (iteratively) is:

$$Q = \sigma\varepsilon FA(T_I^4 - T_J^4) \tag{13.30}$$

where:

Q = heat flow rate between nodes I and J (output as HEAT RATE)
σ = Stefan-Boltzmann constant (input as SBC on **R** command)
ε = emissivity (input as EMISSIVITY on **R** or EMIS on **MP** command)
F = geometric form factor (input as FORM FACTOR on **R** command)
A = area of element (input as AREA on **R** command)
T_I, T_J = absolute temperatures at nodes I and J

The program uses a linear equation solver. Therefore, Equation 13.30 (p. 431) is expanded as:

$$Q = \sigma\varepsilon FA(T_I^2 + T_J^2)(T_I + T_J)(T_I - T_J) \tag{13.31}$$

and then rewritten as:

$$Q = \sigma\varepsilon FA(T_{I,n-1}^2 + T_{J,n-1}^2)(T_{I,n-1} + T_{J,n-1})(T_{I,n} - T_{J,n}) \tag{13.32}$$

where the subscripts n and n-1 refer to the current and previous iterations, respectively. It is then recast into finite element form:

$$\begin{Bmatrix} Q_I \\ Q_J \end{Bmatrix} = C_o \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} T_{I,n} \\ T_{J,n} \end{Bmatrix} \tag{13.33}$$

with

$$C_o = \sigma\varepsilon FA(T_{I,n-1}^2 + T_{J,n-1}^2)(T_{I,n-1} + T_{J,n-1}) \tag{13.34}$$

13.31.2. Empirical Radiation (KEYOPT(3) = 1)

The basic equation is:

$$Q = \sigma\varepsilon(FT_I^4 - AT) \tag{13.35}$$

instead of Equation 13.30 (p. 431). This form leads to

$$C_o = \sigma\varepsilon\left(F^{\frac{1}{2}}T_{I,n-1}^2 + A^{\frac{1}{2}}T_{J,n-1}^2\right)\left(F^{\frac{1}{4}}T_{I,n-1} + A^{\frac{1}{4}}T_{J,n-1}\right) \tag{13.36}$$

instead of Equation 13.34 (p. 431). And, hence the matrix Equation 13.33 (p. 431) becomes:

$$\begin{Bmatrix} Q_I \\ Q_J \end{Bmatrix} = C_o \begin{bmatrix} F^{\frac{1}{4}} & -A^{\frac{1}{4}} \\ -F^{\frac{1}{4}} & A^{\frac{1}{4}} \end{bmatrix} \begin{Bmatrix} T_{I,n} \\ T_{J,n} \end{Bmatrix} \tag{13.37}$$

13.31.3. Solution

If the emissivity is input on a temperature dependent basis, Equation 13.34 (p. 431) is rewritten to be:

$$C_o = \sigma FA(\beta_{I,n-1}^2 + \beta_{J,n-1}^2)(\beta_{I,n-1} + \beta_{J,n-1}) \tag{13.38}$$

where:

$$\beta_i = T_i(\varepsilon_i)^{\frac{1}{3}} \qquad (i = 1 \text{ or } J)$$

ε_i = emissivity at node i evaluated at temperature = T_i^f

$T_i^f = T_i - T_{off}$

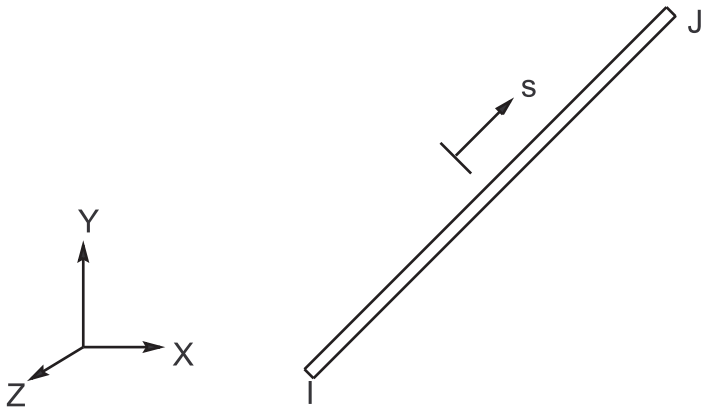
T_{off} = offset temperature (input on **TOFFST** command)

Equation 13.36 (p. 431) is handled analogously.

13.32. Reserved for Future Use

This section is reserved for future use.

13.33. LINK33 - 3-D Conduction Bar



Matrix or Vector	Shape Functions	Integration Points
Conductivity and Specific Heat Matrices; and Heat Generation Load Vector	Equation 11.13 (p. 330)	None

13.33.1. Other Applicable Sections

Heat Flow (p. 227) describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations.

13.33.2. Matrices and Load Vectors

The conductivity matrix is:

$$[K_e^t] = \frac{AK_x}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.39}$$

where:

A = area (input as AREA on **R** command)
K_x = conductivity (input as KXX on **MP** command)
L = distance between nodes

The specific heat matrix is:

$$[C_e^t] = \frac{\rho C_p AL}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{13.40}$$

where:

ρ = density (input as DENS on **MP** command)
C_p = specific heat (input as C on **MP** command)

This specific heat matrix is a diagonal matrix with each diagonal being the sum of the corresponding row of a consistent specific heat matrix. The heat generation load vector is:

$$\{Q_e\} = \frac{\ddot{q}AL}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{13.41}$$

where:

$\ddot{q}$ = heat generation rate (input on **BF** or **BFE** command)

13.33.3. Output

The output is computed as:

$$q = K_x \frac{(T_I - T_J)}{L} \tag{13.42}$$

and

$$Q = qA \tag{13.43}$$

where:

q = thermal flux (output as THERMAL FLUX)
T_I = temperature at node I
T_J = temperature at node J
Q = heat rate (output as HEAT RATE)

13.34. LINK34 - Convection Link

Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	None (nodes may be coincident)	None

13.34.1. Conductivity Matrix

The element conductivity (convection) matrix is

$$[K_e^t] = Ah_f^{eff}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.44}$$

where:

A = area over which element acts (input as AREA on **R** command)

h_f^{eff} = effective film coefficient, defined by equation below

The effective film coefficient is:

$$h_f^{eff} = \begin{cases} \text{maximum of } (h_f', C_c) & \text{if KEYOPT(3) = 3} \\ h_f' + C_c & \text{if KEYOPT(3)} \neq 3 \end{cases} \tag{13.45}$$

where:

h_f' = partial film coefficient term defined by equation below

C_c = user input constant (input as CC on **R** command)

The partial film coefficient term is:

$$h_f' = \begin{cases} Fh_f & \text{if } n = 0.0 \\ Fh_f \left|\Delta T_p\right|^n & \text{if } n \neq 0.0 \text{ and } \Delta T_p \neq 0 \\ 0.0 & \text{if } n \neq 0.0 \text{ and } \Delta T_p = 0 \end{cases} \tag{13.46}$$

where:

$$F = \begin{cases} T_B & \text{if } T_B > 0 \text{ and KEYOPT(3) = 2} \\ 1.0 & \text{if } T_B \leq 0 \quad \text{or KEYOPT(3)} \neq 2 \end{cases}$$

T_B = bulk temperature (input as TBULK on **SFE** command)

$$h_f = \begin{cases} & \text{if KEYOPT(3)} \neq 2 \\ H(me) & \text{or} \\ & \text{if KEYOPT(3) = 2 and } h_f^{in} = 0.0 \\ h_f^{in} & \text{if KEYOPT(3) = 2 and } h_f^{in} > 0.0 \end{cases}$$

H(x) = alternate film coefficient (input on **MP**,HF command for material x)

m_e = material number for this element (input on **MAT** command)

h_f^{in} = primary film coefficient (input on **SFE**,,,CONV,1 command)

$\Delta T_p = T_{p,J} - T_{p,I}$
$T_{p,J}$ = temperature from previous iteration at node J
n = exponent on temperature change (input as EN on **R** command)

ΔT_p must be thought of as unitless, even though it is obviously derived from temperatures.

The heat generation load vector is:

$$\{Q_e\} = \frac{\ddot{q}AL}{2}\begin{Bmatrix}1\\1\end{Bmatrix} \tag{13.47}$$

where:

$\ddot{q}$ = heat generation rate (input on **BF** or **BFE** command)
L = distance between nodes

13.34.2. Output

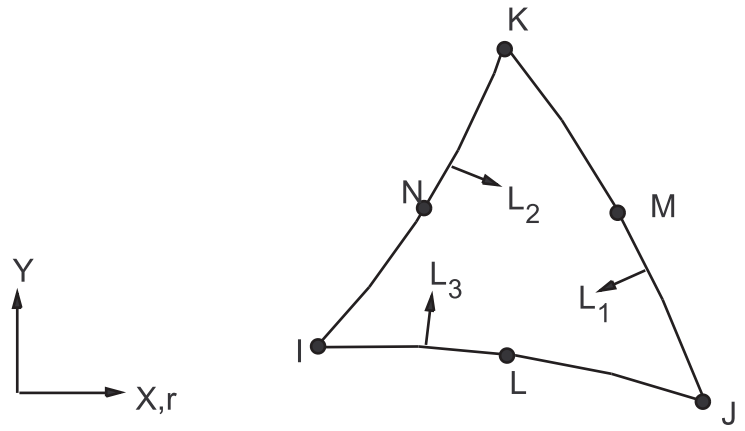
The output is computed as:

$$Q = Ah_f^{eff}(T_I - T_J) \tag{13.48}$$

where:

Q = heat rate (output as HEAT RATE)
T_I = temperature at node I
T_J = temperature at node J

13.35. PLANE35 - 2-D 6-Node Triangular Thermal Solid



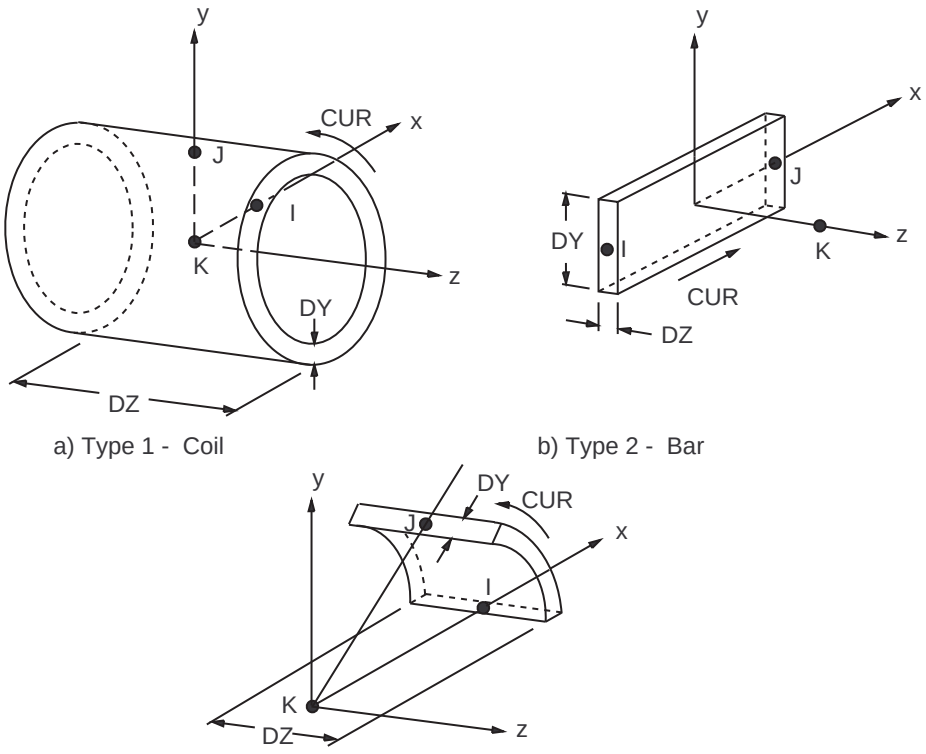
Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Equation 11.117 (p. 341)	6
Specific Heat Matrix	Equation 11.117 (p. 341). If KEYOPT(1) = 1, matrix is diagonalized as described in Lumped Matrices (p. 391)	6

Matrix or Vector	Shape Functions	Integration Points
Convection Surface Matrix and Load Vector	Equation 11.117 (p. 341), specialized to the face	2

13.35.1. Other Applicable Sections

Heat Flow (p. 227) describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations.

13.36. SOURC36 - Current Source



a) Type 1 - Coil

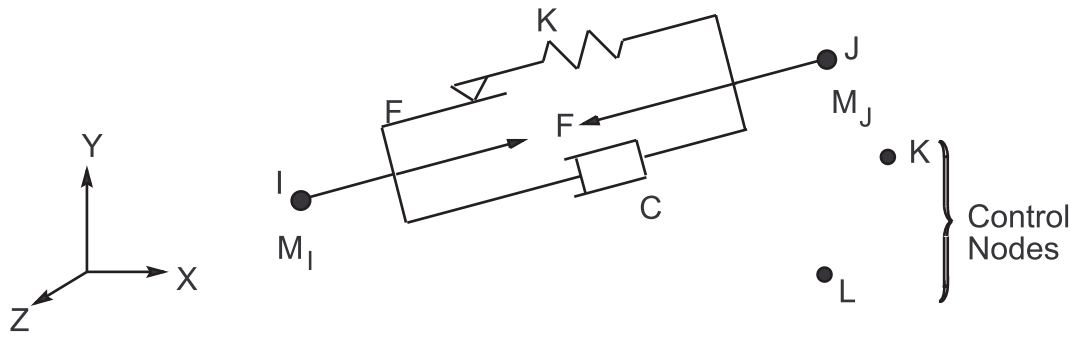
b) Type 2 - Bar

c) Type 3 - Arc

13.36.1. Description

The functionality of SOURC36 is basically one of user convenience. It provides a means of specifying the necessary data to evaluate the Biot-Savart integral (Equation 5.18 (p. 181)) for the simple current source configurations, coil, bar and arc. The magnetic field $\{H_s\}$ that results from this evaluation in turn becomes a load for the magnetic scalar potential elements (SOLID5, SOLID96 and SOLID98) as discussed in Electromagnetics (p. 177).

13.37. COMBIN37 - Control



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (nodes may be coincident)	None
Mass Matrix	None (lumped mass formulation)	None
Damping Matrix	None	None

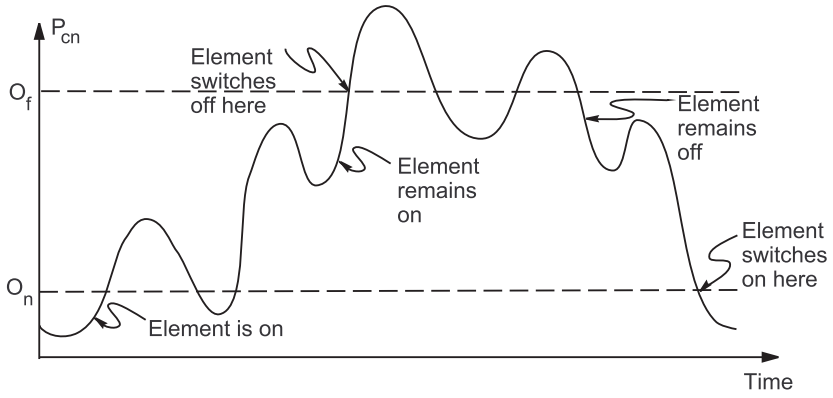
13.37.1. Element Characteristics

COMBIN37 is a nonlinear, 1-D element with two active nodes and one or two control nodes. The element has spring-damper-sliding capability similar to COMBIN40. The degree of freedom (DOF) for the active nodes is selected using KEYOPT(3) and the DOF for the control nodes is selected using KEYOPT(2).

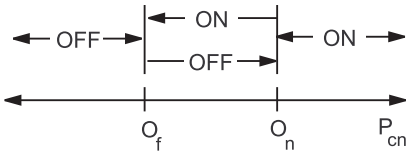
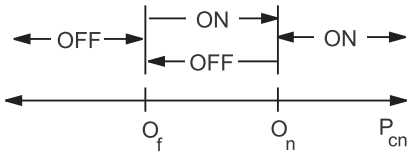
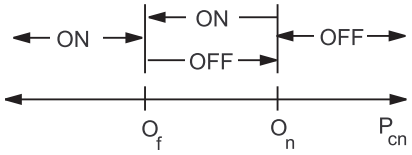
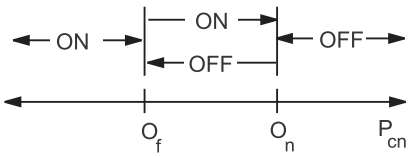
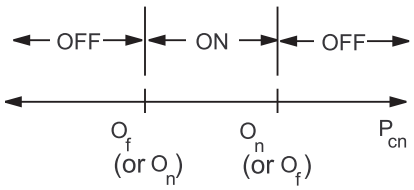
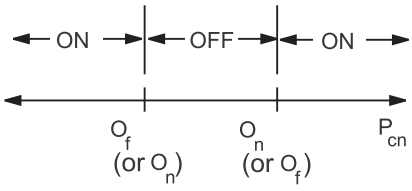
The action of the element in the structure is based upon the value of the control parameter (P_{cn}) (explained later), O_n and O_f (input as ONVAL and OFFVAL on **R** command), and the behavior switches KEYOPT(4) and (5). Figure 13.4: Element Behavior (p. 437) illustrates the behavior of one of the more common modes of operation of the element. It is analogous to the normal home thermostat during the winter.

The behavior of all possible combinations of KEYOPT(4) and (5) values is summarized in the following table. P_{cn} represents the control parameter (output as CONTROL PARAM). The element is active where the figure indicates on, and inactive where it indicates off. For some options, the element may be either on or off for P_{cn} between O_n and O_f, depending upon the last status change.

Figure 13.4: Element Behavior



KEYOPT(4) = 0, KEYOPT(5) = 1, and $O_f > O_n$

KEYOPT(4) = 0, KEYOPT(5) = 0, $O_f \leq O_n$: 	KEYOPT(4) = 0, KEYOPT(5) = 0, $O_f > O_n$: 
KEYOPT(4) = 0, KEYOPT(5) = 1, $O_f \leq O_n$: 	KEYOPT(4) = 0, KEYOPT(5) = 1, $O_f > O_n$:  [1]
KEYOPT(4) = 1, KEYOPT(5) = 0: 	KEYOPT(4) = 1, KEYOPT(5) = 1: 

1. Analogous to Figure 13.4: Element Behavior (p. 437)

13.37.2. Element Matrices

When the element status is ON, the element matrices are:

$$[K_e] = k_o \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.49}$$

$$[M_e] = \begin{bmatrix} M_I & 0 \\ 0 & M_J \end{bmatrix} \tag{13.50}$$

$$[C_e] = C_o \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.51}$$

where:

k_o = stiffness (input as STIF on **R** command)
M_I = mass at node I (input as MASI on **R** command)
M_J = mass at node J (input as MASJ on **R** command)
C_o = damping constant (input as DAMP on **R** command)

When the element status is OFF, all element matrices are set to zero.

13.37.3. Adjustment of Real Constants

If KEYOPT(6) > 0, a real constant is to be adjusted as a function of the control parameter as well as other real constants. Specifically,

$$\text{if KEYOPT(6)} = 0 \text{ or } 1,\ k_o' = k_o + D \tag{13.52}$$

$$\text{if KEYOPT(6)} = 2,\ C_o' = C_o + D \tag{13.53}$$

$$\text{if KEYOPT(6)} = 3,\ M_J' = M_J + D \tag{13.54}$$

$$\text{if KEYOPT(6)} = 4,\ O_n' = O_n + D \tag{13.55}$$

$$\text{if KEYOPT(6)} = 5,\ O_f' = O_f + D \tag{13.56}$$

$$\text{if KEYOPT(6)} = 6,\ F_A' = F_A + D \tag{13.57}$$

$$\text{if KEYOPT(6)} = 7,\ M_I' = M_I + D \tag{13.58}$$

$$\text{if KEYOPT(6)} = 8,\ F_S' = F_S + D \tag{13.59}$$

where:

$$D = \begin{cases} C_1 \left| P_{cn} \right|^{C_2} + C_3 \left| P_{cn} \right|^{C_4} & \text{if KEYOPT(9) = 0} \\ f_1(C_1, C_2, C_3, C_4, P_{cn}) & \text{if KEYOPT(9) = 1} \end{cases}$$

F_A = element load (input as AFORCE ON **R** command)
F_S = slider force (input as FSLIDE on **RMORE** command)
C_1, C_2, C_3, C_4 = input constants (input as C1, C2, C3, and C4 on **RMORE** command)
P_{cn} = control parameter (defined below)
f_1 = function defined by subroutine USERRC

If F_S' (or F_S, if KEYOPT(6) $\neq$ 8) is less than zero, it is reset to zero.

13.37.4. Evaluation of Control Parameter

The control parameter is defined as:

$$P_{cn} = \begin{cases} V & \text{if KEYOPT(1) = 0 or 1} \\ \dfrac{dV}{dt} & \text{if KEYOPT(1) = 2} \\ \dfrac{d^2V}{dt^2} & \text{if KEYOPT(1) = 3} \\ \displaystyle\int_o^t V dt & \text{if KEYOPT(1) = 4} \\ t & \text{if KEYOPT(1) = 5} \end{cases} \tag{13.60}$$

where:

$$V = \begin{cases} u(K) - u(L) & \text{if node L is defined} \\ u(K) & \text{if node L is not defined} \end{cases}$$

t = time (input on **TIME** command)
u = degree of freedom as selected by KEYOPT(2)

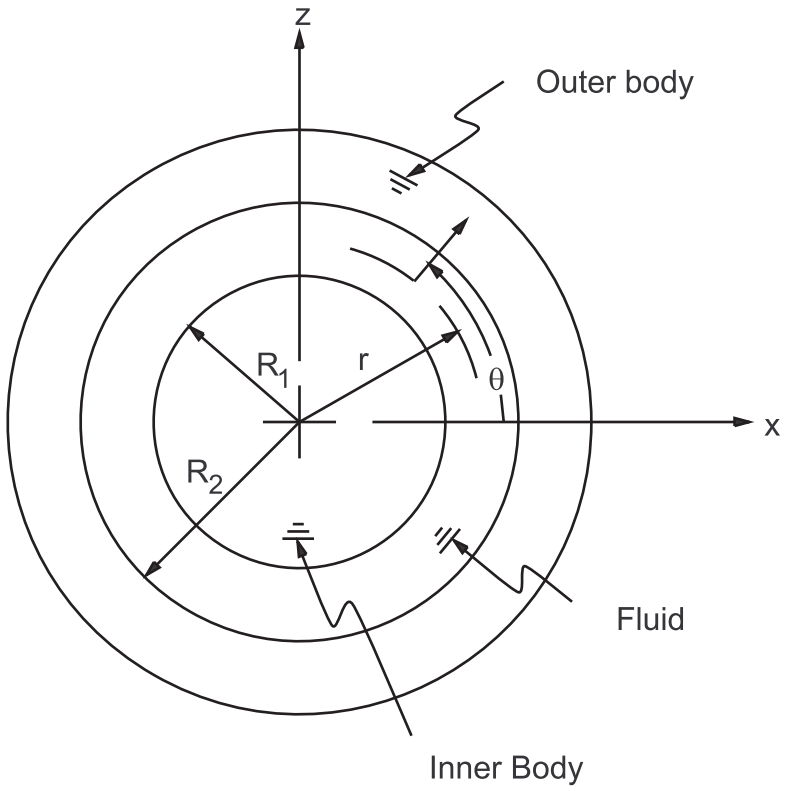
The assumed value of the control parameter for the first iteration (P_{cn}^1) is defined as:

$$P_{cn}^1 = \begin{cases} \dfrac{O_n + O_f}{2} & \text{if } S^t = 1 \text{ or } -1 \\ \text{or} & \\ T_{UNIF} & \text{if } S^t = 0 \text{ and KEYOPT}(2) = 8 \\ \text{or} & \\ 0 & \text{all other cases} \end{cases} \tag{13.61}$$

where:

S^t = constant defining starting status where: 1 means ON, -1 means OFF (input as START on **R** command)
T_{UNIF} = uniform temperature (input on **BFUNIF** command)

13.38. FLUID38 - Dynamic Fluid Coupling



Matrix or Vector	Shape Functions	Integration Points
Mass Matrix	$u = \left(\frac{C_1}{r^2} - C_2\right)\cos\theta$	None

Matrix or Vector	Shape Functions	Integration Points
	$w = \left(\frac{C_1}{r^2} - C_2\right)\sin\theta$	
Damping Matrix	Not defined	None

Reference: Fritz([12] (p. 921))

13.38.1. Description

This element is used to represent a dynamic coupling between two points of a structure. The coupling is based on the dynamic response of two points connected by a constrained mass of fluid. The points represent the centerlines of concentric cylinders. The fluid is contained in the annular space between the cylinders. The cylinders may be circular or have an arbitrary cross-section. The element has two DOFs per node: translations in the nodal x and z directions. The axes of the cylinders are assumed to be in the nodal y directions. These orientations may be changed with KEYOPT(6).

13.38.2. Assumptions and Restrictions

1. The motions are assumed to be small with respect to the fluid channel thickness.
2. The fluid is assumed to be incompressible.
3. Fluid velocities should be less than 10% of the speed of sound in the fluid.
4. The flow channel length should be small compared to the wave length for propagating vibratory disturbances (less than about 10%), in order to avoid the possibility of standing wave effects.

13.38.3. Mass Matrix Formulation

The mass matrix formulation used in the element is of the following form:

$$[M_e] = \begin{bmatrix} m_{11} & 0 & m_{13} & 0 \\ 0 & m_{22} & 0 & m_{24} \\ m_{31} & 0 & m_{33} & 0 \\ 0 & m_{42} & 0 & m_{44} \end{bmatrix} \tag{13.62}$$

The m values are dependent upon the KEYOPT(3) value selected. For KEYOPT(3) = 0 (concentric cylinder case):

$$m_{11} = m_{22} = M(R_1^4 + R_1^2 R_2^2) \tag{13.63}$$

$$m_{13} = m_{31} = m_{24} = m_{42} = -M(2R_1^2 R_2^2) \tag{13.64}$$

$$m_{33} = m_{44} = M(R_1^2 R_2^2 + R_2^4) \tag{13.65}$$

where:

$$M = \frac{\pi L \rho}{R_2^2 - R_1^2} \left(\text{Mass}/\text{Length}^4\right)$$

ρ = fluid mass density (input as DENS on **MP** command)

R_1 = radius of inner cylinder (input as R1 on **R** command)
R_2 = radius of outer cylinder (input as R2 on **R** command)
L = length of cylinders (input as L on **R** command)

Note that the shape functions are similar to that for PLANE25 with MODE = 1. The element mass used in the evaluation of the total structure mass is $\pi L\rho(R_2^2 - R_1^2)$.

For KEYOPT(3) = 2, which is a generalization of the above cylindrical values but for different geometries, the m values are as follows:

$$m_{11} = M_{hx} \tag{13.66}$$

$$m_{13} = m_{31} = -(M_1 + M_{hx}) \tag{13.67}$$

$$m_{33} = (M_1 + M_2 + M_{hx}) \tag{13.68}$$

$$m_{22} = M_{hz} \tag{13.69}$$

$$m_{24} = m_{42} = -(M_1 + M_{hz}) \tag{13.70}$$

$$m_{44} = M_1 + M_2 + M_{hz} \tag{13.71}$$

where:

M_1 = mass of fluid displaced by the inner boundary (Boundary 1) (input as M1 on **R** command)
M_2 = mass of fluid that could be contained within the outer boundary (Boundary 2) in absence of the inner boundary (input as M2 on **R** command)
M_{hx}, M_{hz} = hydrodynamic mass for motion in the x and z directions, respectively (input as MHX and MHZ on **R** command)

The element mass used in the evaluation of the total structure mass is M_2 - M_1.

The lumped mass option (**LUMPM**,ON) is not available.

13.38.4. Damping Matrix Formulation

The damping matrix formulation used in the element is of the following form:

$$[C_e] = \begin{bmatrix} c_{11} & 0 & c_{13} & 0 \\ 0 & c_{22} & 0 & c_{24} \\ c_{31} & 0 & c_{33} & 0 \\ 0 & c_{42} & 0 & c_{44} \end{bmatrix} \tag{13.72}$$

The c values are dependent upon the KEYOPT(3) value selected. For KEYOPT(3) = 0:

$$c_{11} = c_{33} = C\Delta x W_x \tag{13.73}$$

$$c_{13} = c_{31} = -C\Delta x W_x \tag{13.74}$$

$$c_{22} = c_{44} = C\Delta z W_z \tag{13.75}$$

$$c_{24} = c_{42} = -C\Delta z W_z \tag{13.76}$$

where:

$$C = \frac{f\rho L R_1^2 (R_1^2 + R_2^2)}{3(R_2 - R_1)^3} (\text{Mass/Length})$$

W_x, W_z = estimate of resonant frequencies in the x and z response directions, respectively (input as WX, WZ on **RMORE** command)
f = Darcy friction factor for turbulent flow (input as F on **R** command)
Δx, Δz = estimate of peak relative amplitudes between inner and outer boundaries for the x and z motions, respectively (input as DX, DZ on **R** command)

For KEYOPT(3) = 2, the c values are as follows:

$$c_{11} = c_{33} = C_x \Delta x W_x \quad (13.77)$$

$$c_{13} = c_{31} = -C_x \Delta x W_x \quad (13.78)$$

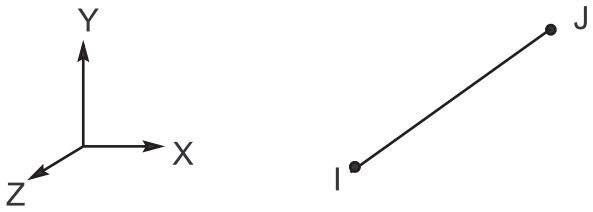
$$c_{22} = c_{44} = C_z \Delta z W_z \quad (13.79)$$

$$c_{24} = c_{42} = -C_z \Delta z W_z \quad (13.80)$$

where:

C_x, C_z = flow and geometry constants for the x and z motions, respectively (input as CX, CZ on **RMORE** command)

13.39. COMBIN39 - Nonlinear Spring



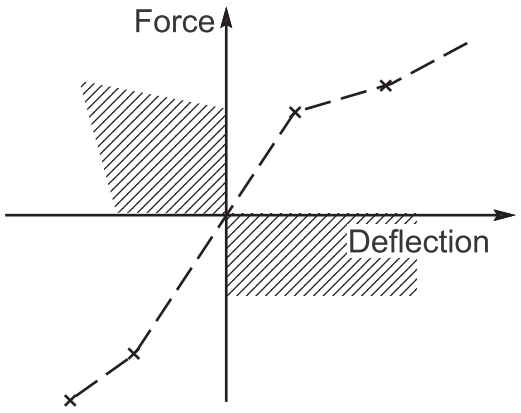
Matrix or Vector	Option	Shape Functions[1]	Integration Points
Stiffness Matrix	Longitudinal	Equation 11.15 (p. 330)	None
	Torsional	Equation 11.18 (p. 330)	None
Stress Stiffening Matrix	Longitudinal	Equation 11.7 (p. 330) and Equation 11.8 (p. 330)	None

1. There are no shape functions used if the element is input as a one DOF per node basis (KEYOPT(4) = 0) as the nodes are coincident.

13.39.1. Input

The user explicitly defines the force-deflection curve for COMBIN39 by the input of discrete points of force versus deflection. Up to 20 points on the curve may be defined, and are entered as real constants. The input curve must pass through the origin and must lie within the unshaded regions, if KEYOPT(1) = 1.

Figure 13.5: Input Force-Deflection Curve



The input deflections must be given in ascending order, with the minimum change of deflection of:

$$u_{i+1} - u_i > \Delta u_{min}, \qquad i=1,19 \tag{13.81}$$

where:

u_i = input deflections (input as D1, D2, ... D20 on **R** or **RMORE** commands)

$$\Delta u_{min} = \frac{u_{max} - u_{min}}{10^7}$$

u_{max} = most positive input deflection

u_{min} = most negative input deflection

13.39.2. Element Stiffness Matrix and Load Vector

During the stiffness pass of a given iteration, COMBIN39 will use the results of the previous iteration to determine which segment of the input force-deflection curve is active. The stiffness matrix and load vector of the element are then:

$$[K_e] = K^{tg}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.82}$$

$$\{F_e^{nr}\} = F_1\begin{Bmatrix} 1 \\ -1 \end{Bmatrix} \tag{13.83}$$

where:

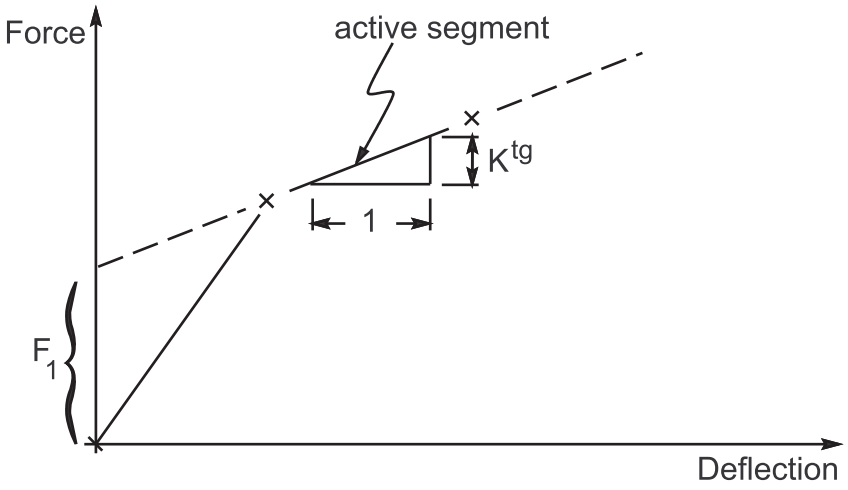
K^{tg} = slope of active segment from previous iteration (output as SLOPE)

F_1 = force in element from previous iteration (output as FORCE)

If KEYOPT(4) > 0, Equation 13.82 (p. 444) and Equation 13.83 (p. 444) are expanded to 2 or 3 dimensions.

During the stress pass, the deflections of the current equilibrium iteration will be examined to see whether a different segment of the force-deflection curve should be used in the next equilibrium iteration.

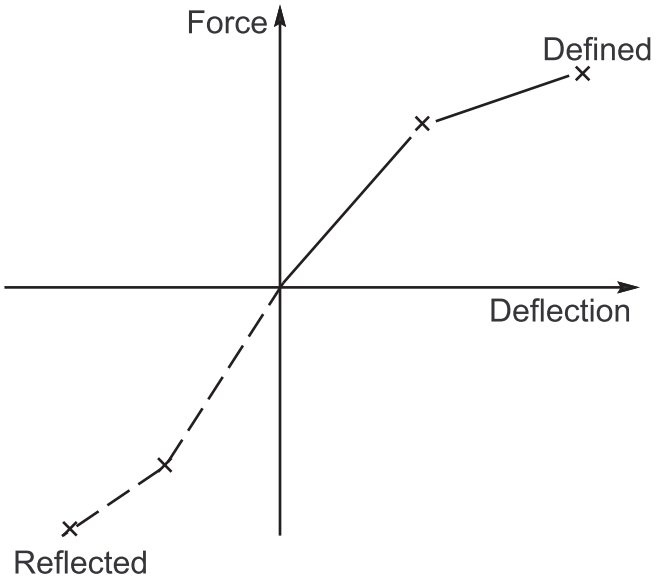
Figure 13.6: Stiffness Computation



13.39.3. Choices for Element Behavior

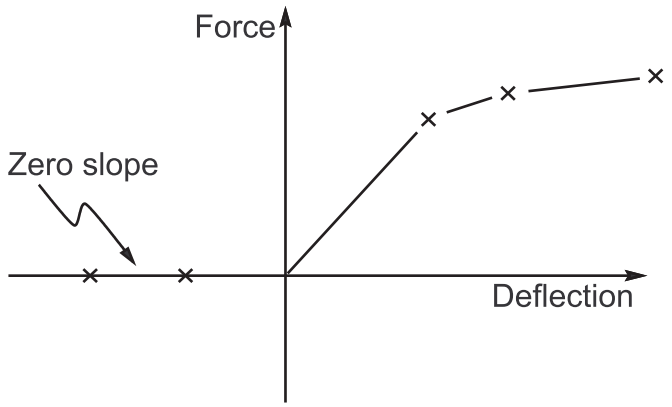
If KEYOPT(2) = 0 and if no force-deflection points are input for deflection less than zero, the points in the first quadrant are reflected through the origin (Figure 13.7: Input Force-Deflection Curve Reflected Through Origin (p. 445)).

Figure 13.7: Input Force-Deflection Curve Reflected Through Origin



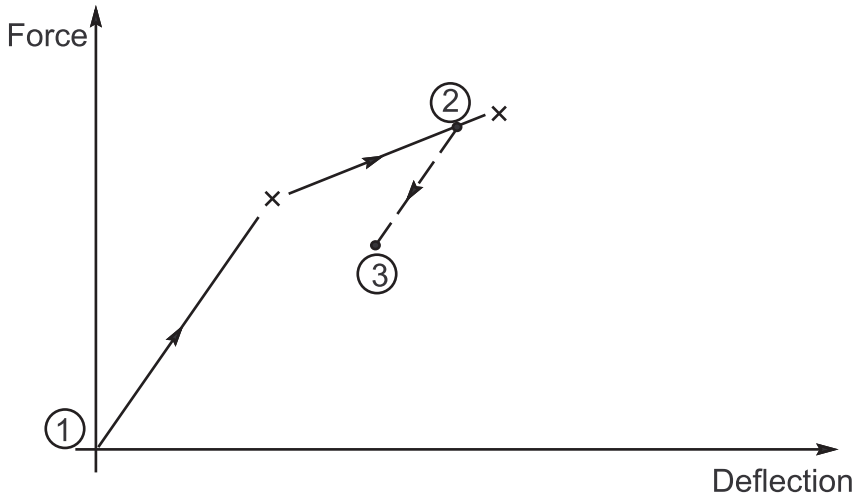
If KEYOPT(2) = 1, there will be no stiffness for the deflection less than zero (Figure 13.8: Force-Deflection Curve with KEYOPT(2) = 1 (p. 446)).

Figure 13.8: Force-Deflection Curve with KEYOPT(2) = 1



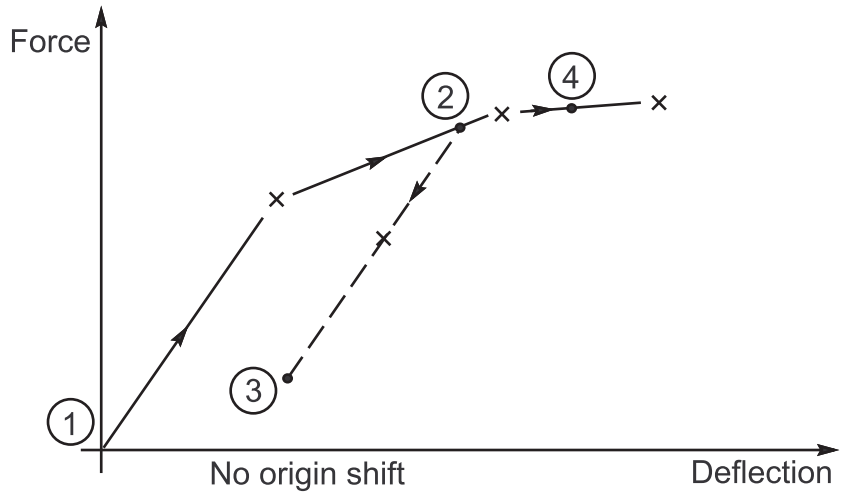
If KEYOPT(1) = 0, COMBIN39 is conservative. This means that regardless of the number of loading reversals, the element will remain on the originally defined force-deflection curve, and no energy loss will occur in the element. This also means that the solution is not path-dependent. If, however, KEYOPT(1) = 1, the element is nonconservative. With this option, energy losses can occur in the element, so that the solution is path-dependent. The resulting behavior is illustrated in Figure 13.9: Nonconservative Unloading (KEYOPT(1) = 1) (p. 446).

Figure 13.9: Nonconservative Unloading (KEYOPT(1) = 1)



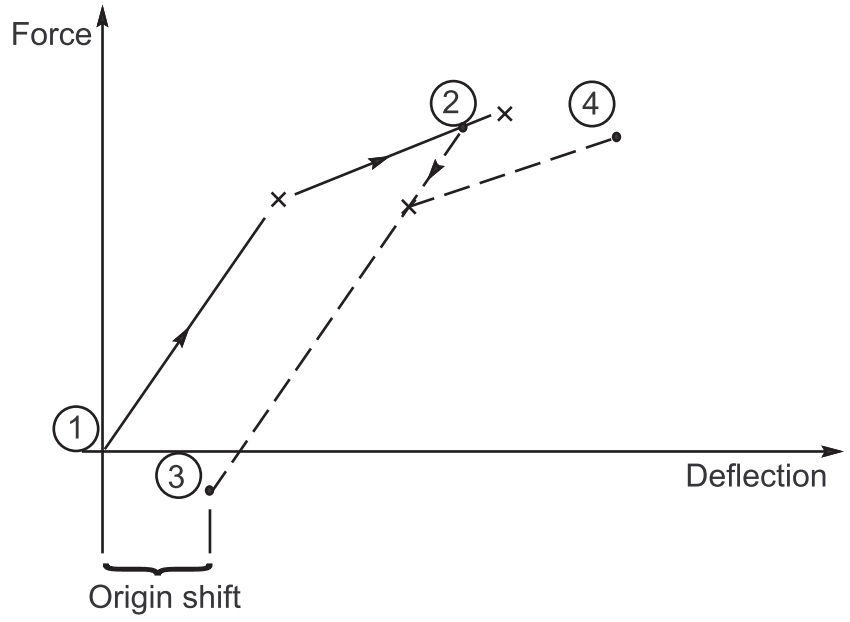
When a load reversal occurs, the element will follow a new force-deflection line passing through the point of reversal and with slope equal to the slope of the original curve on that side of the origin (0+ or 0-). If the reversal does not continue past the force = 0 line, reloading will follow the straight line back to the original curve (Figure 13.10: No Origin Shift on Reversed Loading (KEYOPT(1) = 1) (p. 447)).

Figure 13.10: No Origin Shift on Reversed Loading (KEYOPT(1) = 1)



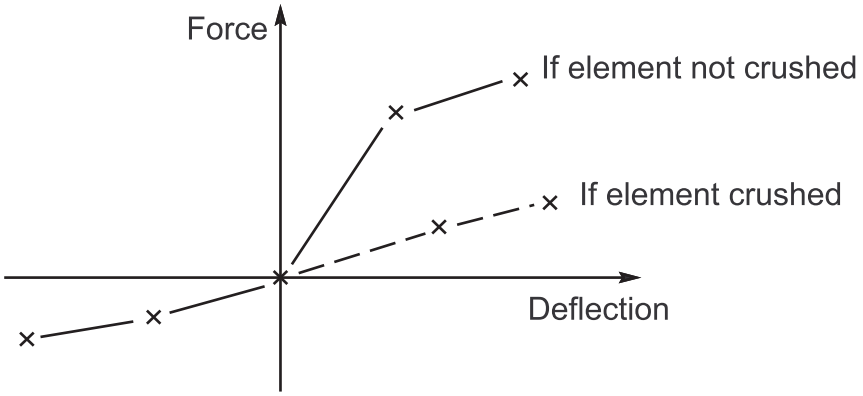
If the reversal continues past the force = 0 line, a type of origin shift occurs, and reloading will follow a curve that has been shifted a distance uorig (output as UORIG) (Figure 13.11: Origin Shift on Reversed Loading (KEYOPT(1) = 1) (p. 447)).

Figure 13.11: Origin Shift on Reversed Loading (KEYOPT(1) = 1)

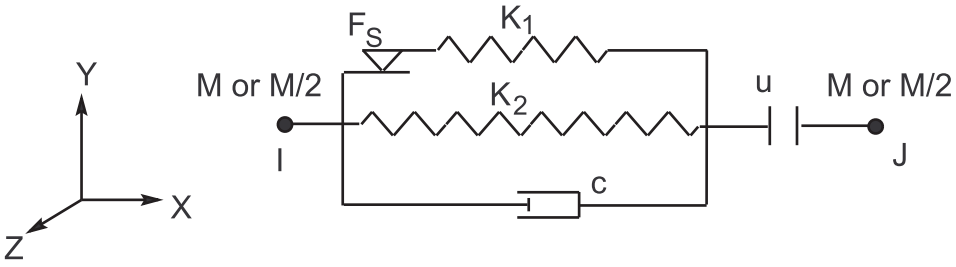


A special option (KEYOPT(2) = 2) is included to model crushing behavior. With this option, the element will follow the defined tensile curve if it has never been loaded in compression. Otherwise, it will follow a reflection through the origin of the defined compressive curve (Figure 13.12: Crush Option (KEYOPT(2) = 2) (p. 448)).

Figure 13.12: Crush Option (KEYOPT(2) = 2)



13.40. COMBIN40 - Combination

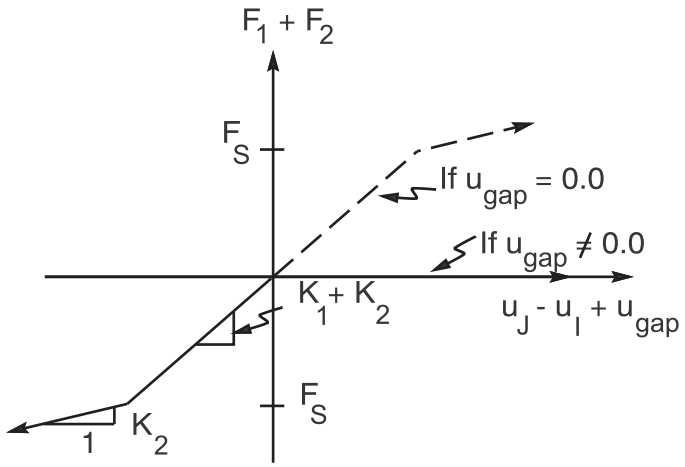


Matrix or Vector	Shape Functions	Integration Points
Stiffness, Mass, and Damping Matrices	None (nodes may be coincident)	None

13.40.1. Characteristics of the Element

The force-deflection relationship for the combination element under initial loading is as shown below (for no damping).

Figure 13.13: Force-Deflection Relationship



where:

F_1 = force in spring 1 (output as F1)
F_2 = force in spring 2 (output as F2)
K_1 = stiffness of spring 1 (input as K1 on **R** command)
K_2 = stiffness of spring 2 (input as K2 on **R** command)
u_{gap} = initial gap size (input as GAP on **R** command) (if zero, gap capability removed)
u_I = displacement at node I
u_J = displacement at node J
F_S = force required in spring 1 to cause sliding (input as FSLIDE on **R** command)

13.40.2. Element Matrices for Structural Applications

The element mass matrix is:

$$[M_e] = M\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \text{ if KEYOPT(6) = 0} \tag{13.84}$$

$$[M_e] = \frac{M}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ if KEYOPT(6) = 1} \tag{13.85}$$

$$[M_e] = M\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \text{ if KEYOPT(6) = 2} \tag{13.86}$$

where:

M = element mass (input as M on **R** command)

If the gap is open during the previous iteration, all other matrices and load vectors are null vectors. Otherwise, the element damping matrix is:

$$[C_e] = c\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.87}$$

where:

c = damping constant (input as C on **R** command)

The element stiffness matrix is:

$$[K_e] = k\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.88}$$

where:

$$k = \begin{cases} K_1 + K_2 & \text{if slider was not sliding in previous iteration} \\ K_2 & \text{if slider was sliding in previous iteration} \end{cases}$$

and the element Newton-Raphson load vector is:

$$\{F_e^{nr}\} = (F_1 + F_2)\begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \tag{13.89}$$

F_1 and F_2 are the current forces in the element.

13.40.3. Determination of F1 and F2 for Structural Applications

1. If the gap is open,

   $$F_1 + F_2 = 0.0 \tag{13.90}$$

   If no sliding has taken place, $F_1 = F_2 = 0.0$. However, if sliding has taken place during unidirectional motion,

   $$F_1 = \frac{u_s K_1 K_2}{K_1 + K_2} \tag{13.91}$$

   and thus

   $$F_2 = -F_1 \tag{13.92}$$

   where:

   u_s = amount of sliding (output as SLIDE)

2. If the gap is closed and the slider is sliding,

   $$F_1 = \pm F_S \tag{13.93}$$

   and

   $$F_2 = K_2 u_2 \tag{13.94}$$

   where:

   $u_2 = u_J - u_I + u_{gap}$ = output as STR2

3. If the gap is closed and the slider is not sliding, but had slid before,

   $$F_1 = K_1 u_1 \tag{13.95}$$

where:

$u_1 = u_2 - u_s$ = output as STR1

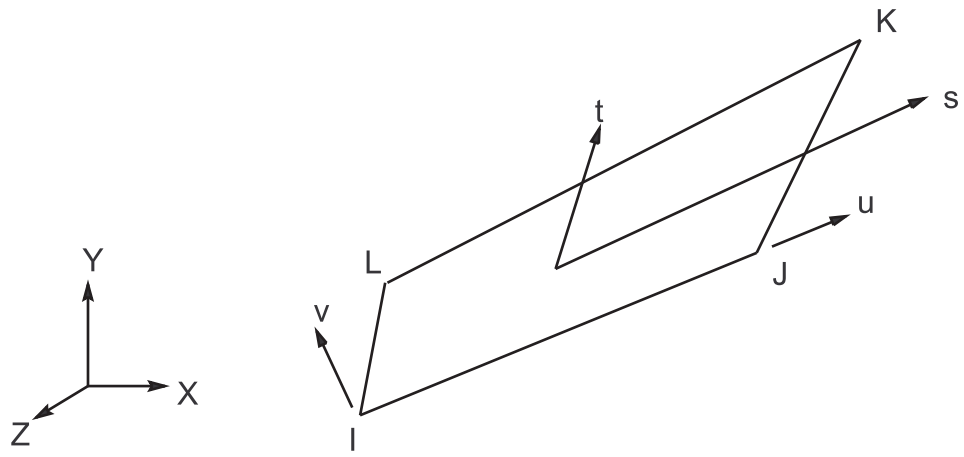
and

$$F_2 = K_2 u_2 \tag{13.96}$$

13.40.4. Thermal Analysis

The above description refers to structural analysis only. When this element is used in a thermal analysis, the conductivity matrix is $[K_e]$, the specific heat matrix is $[C_e]$ and the Newton-Raphson load vector is $\{f_e^{nr}\}$, where F_1 and F_2 represent heat flow. The mass matrix [M] is not used. The gap size u_{gap} is the temperature difference. Sliding, F_{slide}, is the element heat flow limit for conductor K_1.

13.41. SHELL41 - Membrane Shell



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix; and Thermal and Normal Pressure Load Vector	Quad	Equation 11.69 (p. 337) and Equation 11.70 (p. 337) and, if modified extra shape functions are included (KEYOPT(2) = 0) and element has 4 unique nodes Equation 11.82 (p. 338) and Equation 11.83 (p. 338)	2 x 2
	Triangle	Equation 11.49 (p. 336) and Equation 11.50 (p. 336)	1
Foundation Stiffness Matrix	Quad	Equation 11.71 (p. 337)	2 x 2
	Triangle	Equation 11.51 (p. 336)	1
Mass and Stress Stiffness Matrices	Quad	Equation 11.69 (p. 337), Equation 11.70 (p. 337) and Equation 11.71 (p. 337)	2 x 2

Matrix or Vector	Geometry	Shape Functions	Integration Points
	Triangle	Equation 11.49 (p. 336), Equation 11.50 (p. 336), and Equation 11.51 (p. 336)	1
Edge Pressure Load Vector	Same as mass matrix, specialized to the edge		2

Load Type	Distribution
Element Temperature	Bilinear in plane of element, constant thru thickness
Nodal Temperature	Bilinear in plane of element, constant thru thickness
Pressure	Bilinear in plane of element and linear along each edge

References: Wilson([38] (p. 922)), Taylor([49] (p. 923))

13.41.1. Assumptions and Restrictions

There is no out-of-plane bending stiffness.

When the 4-node option of this element is used, it is possible to input these four nodes so they do not lie in an exact flat plane. This is called a warped element, and such a nodal pattern should be avoided because equilibrium is lost. The element assumes that the resisting stiffness is at one location (in the plane defined by the cross product of the diagonals) and the structure assumes that the resisting stiffnesses are at other locations (the nodes). This causes an imbalance of the moments. The warping factor is computed as:

$$\phi = \frac{D}{\sqrt{A}} \tag{13.97}$$

where:

D = component of the vector from the first node to the fourth node parallel to the element normal
A = element area

A warning message will print out if the warping factor exceeds 0.00004 and a fatal message occurs if it exceeds 0.04. Rigid offsets are not used.

13.41.2. Wrinkle Option

When the wrinkle option is requested (KEYOPT(1) = 2), the stiffness is removed when the previous iteration is in compression, which is similar to the logic of the gap elements. This is referred to as the wrinkle option or cloth option. The following logic is used. First, the membrane stresses at each integration point are resolved into their principal directions so that shear is not directly considered. Then, three possibilities exist:

1. Both principal stresses are in tension. In this case, the program proceeds with the full stiffness at this integration point in the usual manner.

2. Both principal stresses are in compression. In this case, the contribution of this integration point to the stiffness is ignored.

3. One of the principal stresses is in tension and one is in compression. In this case, the integration point is treated as an orthotropic material with no stiffness in the compression direction and full stiffness in the tension direction. Then a tensor transformation is done to convert these material properties to the element coordinate system. The rest of the development of the element is done in the same manner is if the option were not used.

13.42. Reserved for Future Use

This section is reserved for future use.

13.43. Reserved for Future Use

This section is reserved for future use.

13.44. Reserved for Future Use

This section is reserved for future use.

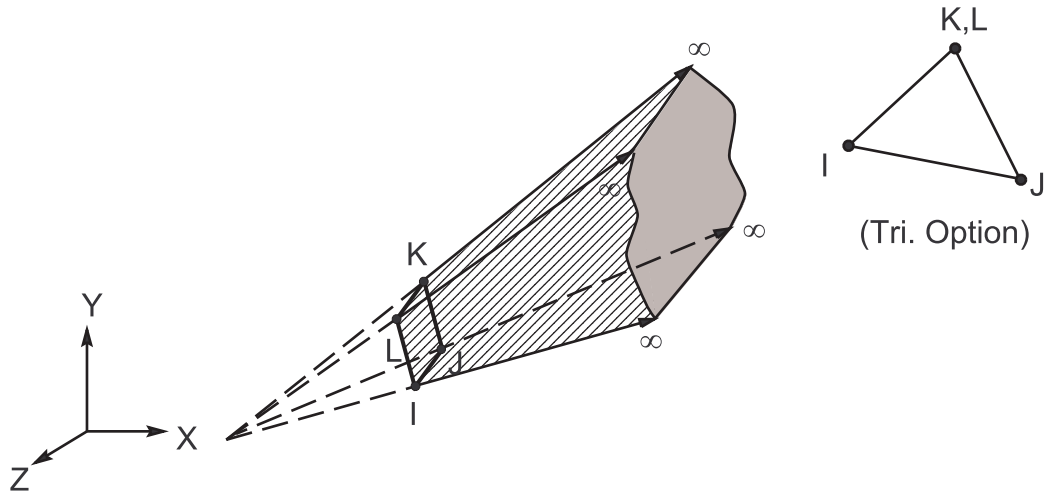
13.45. Reserved for Future Use

This section is reserved for future use.

13.46. Reserved for Future Use

This section is reserved for future use.

13.47. INFIN47 - 3-D Infinite Boundary



Matrix or Vector	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix or Thermal Conductivity Matrix	$\phi = N_I\phi_I + N_J\phi_J + N_K\phi_K$, $N_I = \frac{1}{2A_o}[(x_J y_K - x_K y_J) - (y_K - y_J)x + (x_K - x_J)y]$ $N_J = \frac{1}{2A_o}[(x_K y_I - x_I y_K) - (y_I - y_K)x + (x_I - x_K)y]$ $N_K = \frac{1}{2A_o}[(x_I y_J - x_J y_I) - (y_J - y_I)x + (x_J - x_I)y]$ A_o = area of triangle IJK	None on the boundary element IJK itself, however, 16-point 1-D Gaussian quadrature is applied for some of the integration on each of the edges IJ, JK, and KI of the infinite elements IJML, JKNM, and KILN (see Figure 13.14: A Semi-infinite Boundary Element Zone and the Corresponding Boundary Element IJK (p. 454))

Reference: Kaljevic', et al.([130] (p. 928))

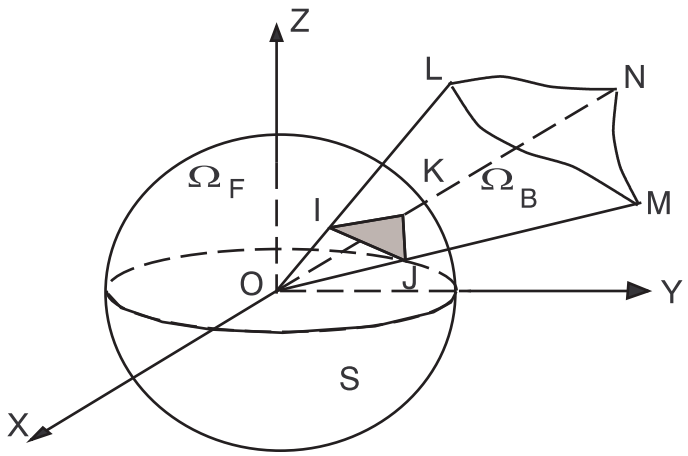
13.47.1. Introduction

This boundary element (BE) models the exterior infinite domain of the far-field magnetic and thermal problems. This element is to be used in combination with 3-D scalar potential solid elements, and can have magnetic scalar potential (MAG), or temperature (TEMP) as the DOF.

13.47.2. Theory

The formulation of this element is based on a first order triangular infinite boundary element (IBE), but the element can be used as a 4-node quadrilateral as well. For unbounded field problems, the model domain is set up to consist of an interior volumetric finite element domain, Ω_F, and a series of exterior volumetric BE subdomains, Ω_B, as shown in Figure 13.14: A Semi-infinite Boundary Element Zone and the Corresponding Boundary Element IJK (p. 454). Each subdomain, Ω_B, is treated as an ordinary BE domain consisting of five segments: the boundary element IJK, infinite elements IJML, JKNM and KILN, and element LMN; element LMN is assumed to be located at infinity.

Figure 13.14: A Semi-infinite Boundary Element Zone and the Corresponding Boundary Element IJK



The approach used here is to write BE equations for Ω_B, and then convert them into equivalent load vectors for the nodes I, J and K. The procedure consists of four steps that are summarized below (see (Kaljevic', et al.[130] (p. 928)) for details).

First, a set of boundary integral equations is written for Ω_B. To achieve this, the potential (or temperature) and its normal derivatives (fluxes) are interpolated on the triangle IJK (Figure 13.14: A Semi-infinite Boundary Element Zone and the Corresponding Boundary Element IJK (p. 454)) by linear shape functions:

$$\phi(x,y) = N_I\phi_I + N_J\phi_J N_K\phi_K \tag{13.98}$$

$$q_n(x,y) = N_I q_{nI} + N_J q_{nJ} + N_K q_{nK} \tag{13.99}$$

where:

ϕ = potential (or temperature)

$q_n = \frac{\partial\phi}{\partial n}$ = normal derivative or flux

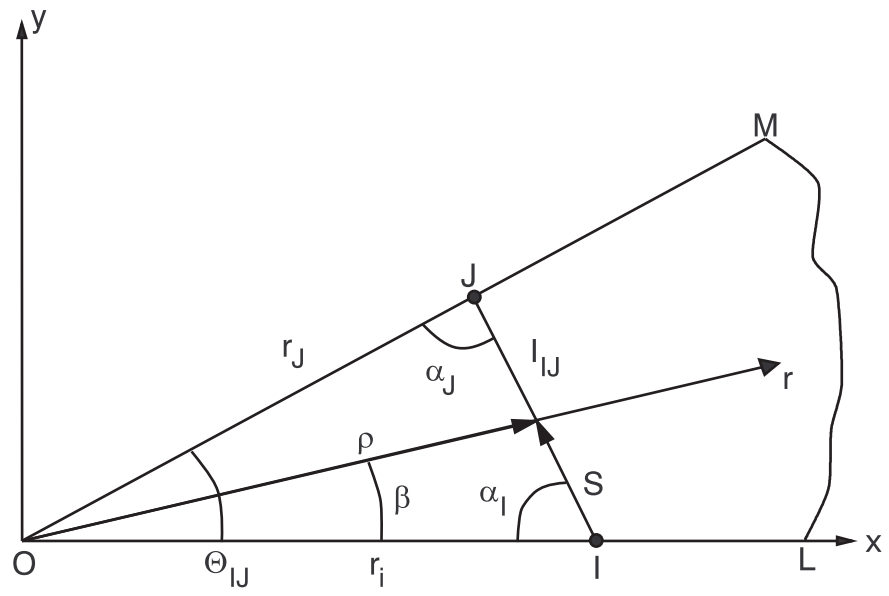
N_I, N_J, N_K = linear shape functions defined earlier

ϕ_I, ϕ_J, ϕ_K = nodal potentials (or temperatures)

q_{nI}, q_{nJ}, q_{nK} = nodal normal derivatives (or fluxes)

n = normal to the surface IJK

Figure 13.15: Infinite Element IJML and the Local Coordinate System



Over an infinite element, such as IJML (Figure 13.15: Infinite Element IJML and the Local Coordinate System (p. 455)), the dependent variables, i.e., potentials (or temperatures) and their normal derivatives (fluxes) are respectively assumed to be (Figure 13.15: Infinite Element IJML and the Local Coordinate System (p. 455)):

$$\phi(r,\beta) = \left\{\left(1-\frac{s}{L_{IJ}}\right)\phi_I + \left(\frac{s}{L_{IJ}}\right)\phi_J\right\}\left(\frac{\rho}{r}\right)^2 \tag{13.100}$$

$$q_\tau(r,\beta) = \left\{\left(1-\frac{s}{L_{IJ}}\right)q_{\tau J} + \left(\frac{s}{L_{IJ}}\right)q_{\tau J}\right\}\left(\frac{\rho}{r}\right)^3 \tag{13.101}$$

where:

$q_\tau = \frac{\partial\phi}{\partial\tau}$ = normal derivative (or flux) to infinite elements; e.g., IJML (see figure above)

$q_{\tau I}$, $q_{\tau J}$ = nodal (nodes I and J) normal derivatives for infinite element IJML

s = a variable length from node I towards node J

L_{IJ} = length of edge IJ

ρ = radial distance from the origin of the local coordinate system O to the edge IJ

r = radial distance from the edge IJ towards infinity

β = variable angle from x-axis for local polar coordinate system

τ = normal to infinite elements IJML

The boundary integral equations for Ω_B are now written as:

$$c(\xi)\phi(\xi) = \int_{\Gamma_B}[G(x,\xi)q(x) - F(x,\xi)\phi(x)]d\Gamma(x) \tag{13.102}$$

where:

$c(\xi)$ = jump term in boundary element method

$G(x,\xi) = \frac{1}{4\pi kr}$ = Green's function or fundamental solution for Laplace's equation

$F(x,\xi) = \frac{\partial}{\partial n}[G(x,\xi)]$

(x,ξ) = field and source points, respectively

r = distance between field and source points

$K = \begin{cases}\text{Magnetic reluctivity (inverse of free space permeability)}\\ \text{(input on } \textbf{EMUNIT} \text{ command) for AZ DOF (KEYOPT(1) = 0)}\\ \text{or}\\ \text{isotropic thermal conductivity (input as KXX on } \textbf{MP} \text{ command)}\\ \text{for TEMP DOF (KEYOPT(1) = 1)}\end{cases}$

The integrations in Equation 13.102 (p. 456) are performed in closed form on the boundary element IJK. The integrations on the infinite elements IJML, JKNM and KILN in the 'r' direction (Figure 13.15: Infinite Element IJML and the Local Coordinate System (p. 455)) are also performed in closed form. However, a 16-point Gaussian quadrature rule is used for the integrations on each of the edges IJ, JK and KI on the infinite elements.

Second, in the absence of a source or sink in Ω_B, the flux q(r) is integrated over the boundary Γ_B of Ω_B and set to zero:

$$\int_{\Gamma_B} q\,dr = 0 \tag{13.103}$$

Third, geometric constraint conditions that exist between the potential ϕ (or temperature) and its derivatives $\frac{\partial \phi}{\partial n} = q_n$ and $\frac{\partial \phi}{\partial \tau} = q_\tau$ at the nodes I, J and K are written. These conditions would express the fact that the normal derivative q_n at the node I, say, can be decomposed into components along the normals to the two infinite elements IJML and KILN meeting at I and along OI.

Fourth, the energy flow quantity from Ω_B is written as:

$$w = \int_{\Gamma_B} q\phi dr \tag{13.104}$$

This energy flow is equated to that due to an equivalent nodal force vector {F} defined below.

The four steps mentioned above are combined together to yield, after eliminating q_n and $q\tau$,

$$[K]\{\phi\} \equiv \{F\}_{eqv} \tag{13.105}$$

where:

[K] = 3 x 3 equivalent unsymmetric element coefficient matrix
$\{\phi\}$ = 3 x 1 nodal degrees of freedom, MAG or TEMP
$\{F\}_{eqv}$ = 3 x 1 equivalent nodal force vector

The coefficient matrix [K] multiplied by the nodal DOF's $\{\phi\}$ represents the equivalent nodal load vector which brings the effects of the semi-infinite domain Ω_B onto nodes I, J and K.

As mentioned in the beginning, the INFIN47 can be used with magnetic scalar potential elements to solve 3-D magnetic scalar potential problems (MAG degree of freedom). Magnetic scalar potential elements incorporate three different scalar potential formulations (see Electromagnetic Field Fundamentals (p. 177)) selected with the **MAGOPT** command:

1. Reduced Scalar Potential (accessed with **MAGOPT**,0)
2. Difference Scalar Potential (accessed with **MAGOPT**,2 and MAGOPT,3)
3. Generalized Scalar Potential (accessed with **MAGOPT**,1, **MAGOPT**,2, and then **MAGOPT**,3)

13.47.3. Reduced Scalar Potential

If there is no "iron" in the problem domain, the reduced scalar potential formulation can be used both in the FE and the BE regimes. In this case, the potential is continuous across FE-BE interface. If there is "iron" in the FE domain, the reduced potential formulation is likely to produce "cancellation errors".

13.47.4. Difference Scalar Potential

If there is "iron" and current in the FE region and the problem domain is singly-connected, we can use the difference potential formulation in order to avoid cancellation error. The formulation consists of two-step solution procedures:

1. Preliminary solution in the air domain (**MAGOPT**, 2)

   Here the first step consists of computing a magnetic field $\{H_o\}$ under the assumption that the magnetic permeability of iron is infinity, thereby neglecting any saturation. The reduced scalar potential

ϕ is used in FE region and the total scalar potential ψ is used in BE region. In this case, the potential will be discontinuous across the FE-BE interface. The continuity condition of the magnetic field at the interface can be written as:

$$-\nabla\psi \cdot \{\tau\} = -\nabla\phi \cdot \{\tau\} + \{H_s\}^T \{\tau\} \tag{13.106}$$

where:

$\{\tau\}$ = tangent vector at the interface along element edge
$\{H_s\}$ = magnetic field due to current sources

Integrating the above equation along the interface, we obtain

$$\psi_p = \phi_p - \int_{p_o}^{p} \{H_s\}^T \{\tau\} dt \tag{13.107}$$

If we take $\psi = \phi$ at a convenient point p_o on the interface, then the above equation defines the potential jump at any point p on the interface. Now, the total potential ψ can be eliminated from the problem using this equation, leading to the computation of the additional load vector,

$$\{f_g\} = [K]\{g\} \tag{13.108}$$

where:

$$g_i = \int_{p_o}^{p_i} \{H_s\}^T \{\tau\} dt$$

[K] = coefficient matrix defined with Equation 13.105 (p. 457)

2. Total solution (air and iron) (**MAGOPT**, 3)

In this step the total field, $\{H\} = \{H_o\} - \nabla\psi$, is computed where $\{H\}$ is the actual field and $\{H_o\}$ is the field computed in step 1 above. Note that the same relation given in Equation 5.39 (p. 184) uses ϕ_g in place of ψ. The total potential ψ is used in both FE and BE regimes. As a result, no potential discontinuity exists at the interface, but an additional load vector due to the field $\{H_o\}$ must be computed. Since the magnetic flux continuity condition at the interface of air and iron is:

$$\mu_I \frac{\partial\psi_I}{\partial n} - \mu_o \frac{\partial\psi_A}{\partial n} = -\mu_o \{H_o\}^T \{n\} \tag{13.109}$$

where:

μ_o = magnetic permeability of free space (air)
μ_I = magnetic permeability of iron

The additional load vector may be computed as

$$\{f_f\} = -\int_s \mu_o \{N\}\{H_o\}^T \{n\} ds \tag{13.110}$$

where:

{N} = weighting functions

13.47.5. Generalized Scalar Potential

If there is iron and current in the FE domain and the domain is multiply-connected, the generalized potential formulation can be used. It consists of three different steps.

1. Preliminary solution in the iron domain (**MAGOPT**, 1). This step computes a preliminary solution in the iron only. The boundary elements are not used for this step.
2. Preliminary solution in the air domain (**MAGOPT**, 2). This step is exactly the same as the step 1 of the difference potential formulation.
3. Total solution (air and iron) (**MAGOPT**, 3) . This step is exactly the same as the step 2 of the difference potential formulation.

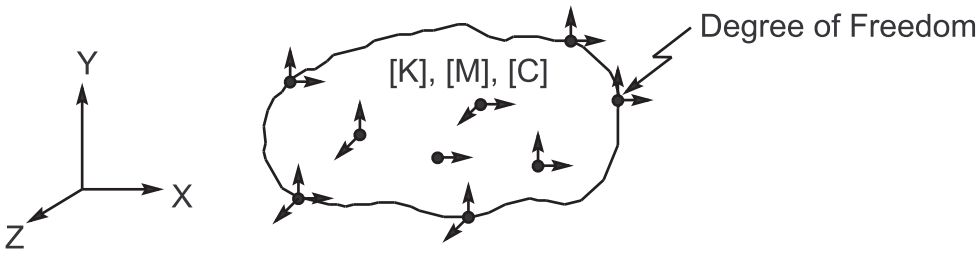
13.48. Not Documented

No detail or element available at this time.

13.49. Not Documented

No detail or element available at this time.

13.50. MATRIX50 - Superelement (or Substructure)



Matrix or Vector	Shape Functions	Integration Points
Stiffness, Conductivity, Stress Stiffness (used only when added to the Stiffness Matrix), Convection Surface Matrices; and Gravity, Thermal and Pressure/Heat Generation and Convection Surface Load Vectors	Same as the constituent elements	Same as the constituent elements
Mass/Specific Heat and Damping Matrices	Same as the constituent elements reduced down to the master degrees of freedom	Same as the constituent elements

Load Type	Distribution
Element Temperature and Heat Generation Rate	As input during generation run
Pressure/Convection Surface Distribution	As input during generation run

13.50.1. Other Applicable Sections

Superelements are discussed in Substructuring Analysis (p. 793).

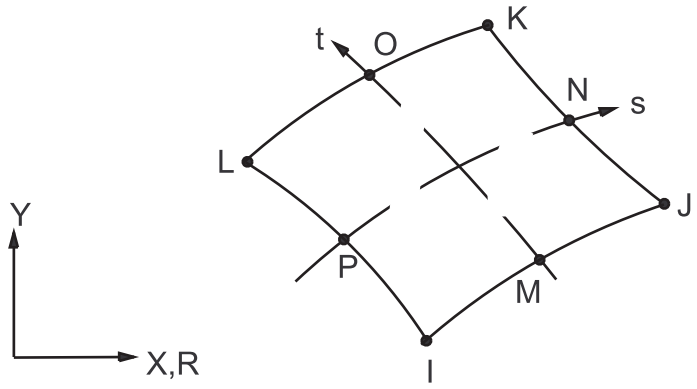
13.51. Not Documented

No detail or element available at this time.

13.52. Reserved for Future Use

This section is reserved for future use.

13.53. PLANE53 - 2-D 8-Node Magnetic Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix; and Permanent Magnet and Applied Current Load Vectors	Quad	Equation 11.137 (p. 342)	2 x 2
	Triangle	Equation 11.115 (p. 341)	3
Damping (Eddy Current) Matrix	Quad	Equation 11.137 (p. 342) and Equation 11.139 (p. 342)	Same as coefficient matrix
	Triangle	Equation 11.115 (p. 341) and Equation 11.118 (p. 341)	Same as coefficient matrix

Load Type	Distribution
Current Density, Voltage Load and Phase Angle Distribution	Bilinear across element

References: Silvester et al.([72] (p. 924)), Weiss et al.([94] (p. 925)), Garg et al.([95] (p. 925))

13.53.1. Other Applicable Sections

Derivation of Electromagnetic Matrices (p. 193) has a complete derivation of the matrices and load vectors of a general magnetic analysis element. Coupled Effects (p. 293) contains a discussion of coupled field analyses.

13.53.2. Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the solution varies linearly along that edge.

13.53.3. VOLT DOF in 2-D and Axisymmetric Skin Effect Analysis

KEYOPT(1) = 1 can be used to model skin effect problems. The corresponding DOFs are AZ and VOLT. Here, AZ represents the z- or θ-component of the magnetic vector potential for 2-D or axisymmetric geometry, respectively. VOLT has different meanings for 2-D and axisymmetric geometry. The difference is explained below for a transient case.

A skin effect analysis is used to find the eddy current distribution in a massive conductor when a source current is applied to it. In a general 3-D case, the (total) current density {J} is given by

$$\{J\} = -\sigma\frac{\partial\{A\}}{\partial t} - \sigma\frac{\partial\{\nabla\nu\}}{\partial t} \tag{13.111}$$

where:

ν = (time-integrated) electric scalar potential

Refer to Magnetic Vector Potential Results (p. 201) for definitions of other variables. For a 2-D massive conductor, the z-component of {J} may be rewritten as:

$$J_z = -\sigma\frac{\partial A_z}{\partial t} + \sigma\frac{\partial\{\nabla\tilde{V}\}}{\partial t} \tag{13.112}$$

where $\Delta\tilde{V}$ may be termed as the (time-integrated) source voltage drop per unit length and is defined by:

$$\Delta\tilde{V} = -\hat{z}\cdot\nabla\nu \tag{13.113}$$

For an axisymmetric massive conductor, the θ-component of {J} may be rewritten as

$$J_\theta = -\sigma\frac{\partial A_\theta}{\partial t} + \frac{\sigma}{2\pi r}\frac{\partial\{\nabla\tilde{V}\}}{\partial t} \tag{13.114}$$

where the (time-integrated) source voltage drop in a full 2π radius is defined by

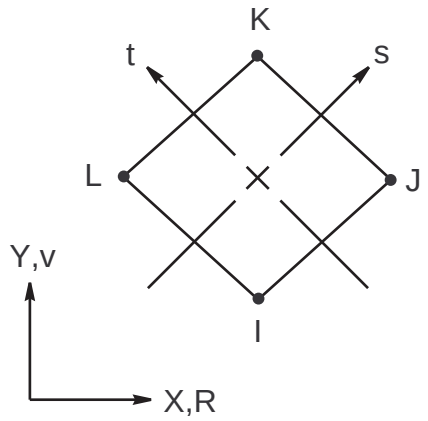
$$\Delta\tilde{V} = -2\pi r\,\hat{\theta}\cdot\nabla\nu \tag{13.115}$$

When KEYOPT(1) = 1, the VOLT DOF represents the definition given by Equation 13.113 (p. 461) and Equation 13.115 (p. 461) for a 2-D and axisymmetric conductor, respectively. Also, all VOLT DOFs in a massive conductor region must be coupled together so that $\Delta\tilde{V}$ has a single value.

13.54. Reserved for Future Use

This section is reserved for future use.

13.55. PLANE55 - 2-D Thermal Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Quad	Equation 11.128 (p. 341)	2 x 2
	Triangle	Equation 11.108 (p. 340)	1 if planar 3 if axisymmetric
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Lumped Matrices (p. 391).		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Same as conductivity matrix evaluated at the face		2

13.55.1. Other Applicable Sections

Heat Flow (p. 227) describes the derivation of the element matrices and load vectors as well as heat flux evaluations. SOLID70 - 3-D Thermal Solid (p. 477) describes fluid flow in a porous medium, accessed in PLANE55 with KEYOPT(9) = 1.

13.55.2. Mass Transport Option

If KEYOPT(8) > 0, the mass transport option is included as described in Heat Flow Fundamentals (p. 227) with Equation 6.1 (p. 227) and by K_e^{tm} of Equation 6.22 (p. 235). The solution accuracy is dependent on the element size. The accuracy is measured in terms of the non-dimensional criteria called the element Peclet number (Gresho([58] (p. 923))):

$$P_e = \frac{VL\rho C_p}{2K} \tag{13.116}$$

where:

V = magnitude of the velocity vector
L = element length dimension along the velocity vector direction
ρ = density of the fluid (input as DENS on **MP** command)
C_p = specific heat of the fluid (input as C on **MP** command)

K = equivalent thermal conductivity along the velocity vector direction

The terms V, L, and K are explained more thoroughly below:

$$V = (V_x^2 + V_y^2)^{1/2} \tag{13.117}$$

where:

V_x = fluid velocity (mass transport) in x direction (input as VX on **R** command)
V_y = fluid velocity (mass transport) in y direction (input as VY on **R** command)

Length L is calculated by finding the intersection points of the velocity vector which passes through the element origin and intersects at the element boundaries.

For orthotropic materials, the equivalent thermal conductivity K is given by:

$$K = K_x K_y \left[\frac{(1+m^2)}{K_y^2 + m^2 K_x^2} \right]^{1/2} \tag{13.118}$$

where:

K_x, K_y = thermal conductivities in the x and y directions (input as KXX and KYY on **MP** command)

$$m = \text{slope of velocity vector in element coordinate system} = \frac{V_y}{V_x} \text{ (if KEYOPT(4) = 0)}$$

For the solution to be physically valid, the following condition has to be satisfied (Gresho([58] (p. 923))):

$$P_e < 1 \tag{13.119}$$

This check is carried out during the element formulation and an error message is printed out if equation (14.431) is not satisfied. When this error occurs, the problem should be rerun after reducing the element size in the direction of the velocity vector.

13.56. Not Documented

No detail or element available at this time.

13.57. Reserved for Future Use

This section is reserved for future use.

13.58. Not Documented

No detail or element available at this time.

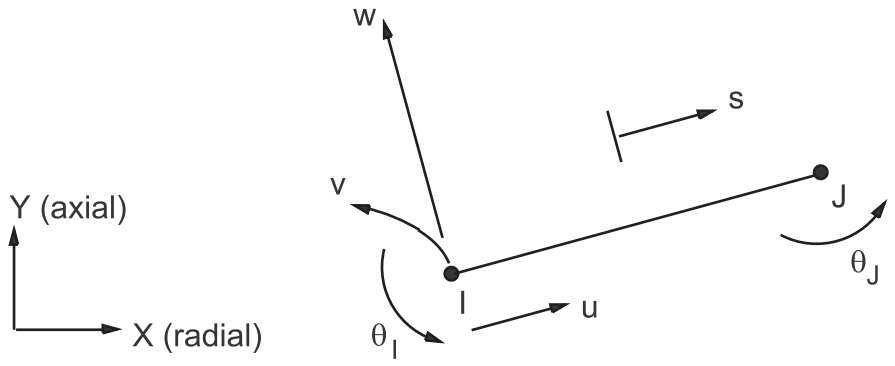
13.59. Reserved for Future Use

This section is reserved for future use.

13.60. Reserved for Future Use

This section is reserved for future use.

13.61. SHELL61 - Axisymmetric-Harmonic Structural Shell



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix; and Thermal and Pressure Load Vectors	Equation 11.38 (p. 333), Equation 11.39 (p. 333), and Equation 11.40 (p. 333). If extra shape functions are not included (KEYOPT(3) = 1): Equation 11.35 (p. 332), Equation 11.36 (p. 333), and Equation 11.37 (p. 333)	3 along length
Mass and Stress Stiffness Matrices	Equation 11.32 (p. 331), Equation 11.33 (p. 332), and Equation 11.34 (p. 332)	Same as stiffness matrix

Load Type	Distribution
Element Temperature	Linear through thickness and along length, harmonic around circumference
Nodal Temperature	Constant through thickness, linear along length, harmonic around circumference
Pressure	Linear along length, harmonic around circumference

Reference: Zienkiewicz([39] (p. 922))

13.61.1. Other Applicable Sections

Structures (p. 5) discusses fundamentals of linear elements. PLANE25 - Axisymmetric-Harmonic 4-Node Structural Solid (p. 425) has a discussion on temperature, applicable to this element.

13.61.2. Assumptions and Restrictions

The material properties are assumed to be constant around the entire circumference, regardless of temperature dependent material properties or loading.

13.61.3. Stress, Force, and Moment Calculations

Element output comes in two forms:

1. Stresses as well as forces and moments per unit length: This printout is controlled by the KEYOPT(6). The thru-the-thickness stress locations are shown in Figure 13.16: Stress Locations (p. 465). The stresses are computed using standard procedures as given in Structural Strain and Stress Evaluations (p. 15). The stresses may then be integrated thru the thickness to give forces per unit length and moments per unit length at requested points along the length:

$$T_x = \sigma_x|_c \, t \tag{13.120}$$

$$T_z = \sigma_z|_c \, t \tag{13.121}$$

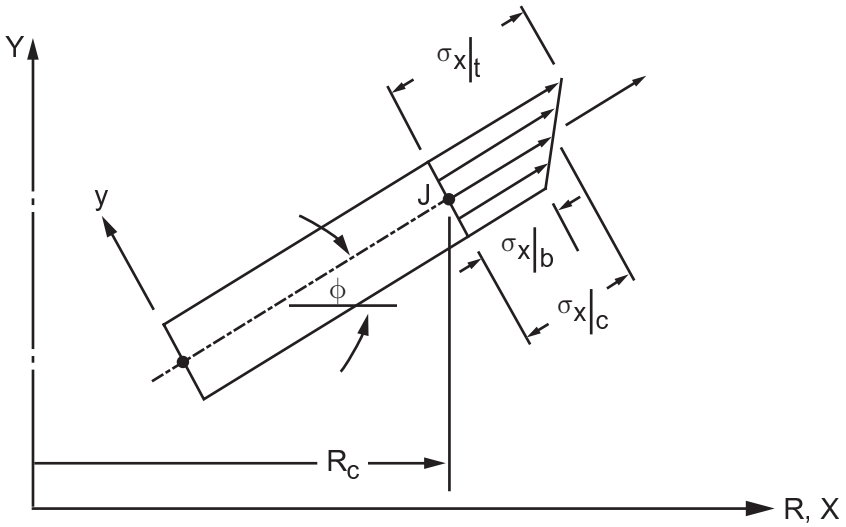
$$T_{xz} = \sigma_{xz}|_c \, t \tag{13.122}$$

$$M_x = (\sigma_x|_c - \sigma_x|_c)\frac{t^2}{12} \tag{13.123}$$

$$M_z = (\sigma_z|_c - \sigma_z|_c)\frac{t^2}{12} \tag{13.124}$$

$$M_{xz} = (\sigma_{xz}|_c - \sigma_{xz}|_c)\frac{t^2}{12} \tag{13.125}$$

Figure 13.16: Stress Locations



where:

T_x, T_z, T_{xz}, M_x, M_z, M_{xz} = resultant forces and moments (output as TX, TZ, TXZ, MX, MZ, MXZ, respectively)

t = thickness (input as TK(I), TK(J) on **R** command)

σ_x, σ_y, σ_z, σ_{xz} = stresses (output as SX, SY, SZ, and SXZ, respectively)

$\sigma_x|_c = (\sigma_x|_t + \sigma_x|_b)/2$ = x stress at centerplane (also nodal locations)

$\sigma_x|_t$ = x stress at top

$\sigma_x|_b$ = x stress at bottom

2. Forces and moments on a circumference basis: This printout is controlled by KEYOPT(4). The values are computed using:

$$\{F_\ell\} = [T_R]^T ([K_e]\{u_e\} - \{F_e^{th}\} - \{F_e^{pr}\}) \quad (13.126)$$

where:

$F_\ell = \left\lfloor F_{x,1}^r \quad F_{y,1}^r \quad F_{z,1}^r \quad M_{z,1}^r \quad F_{x,2}^r \quad F_{y,2}^r \quad F_{z,2}^r \quad M_{z,2}^r \right\rfloor^T$ (output as MFOR and MMOM)

$[T_R]$ = local to global transformation matrix

$[K_e]$ = element stiffness matrix

$\{u_e\}$ = nodal displacements

$\{F_e^{th}\}$ = element thermal load vector

$\{F_e^{pr}\}$ = element pressure load vector

Another difference between the two types of output are the nomenclature conventions. Since the first group of output uses a shell nomenclature convention and the second group of output uses a nodal nomenclature convention, M_z and M_z^r represent moments in different directions.

The rest of this subsection will describe some of the expected relationships between these two methods of output at the ends of the element. This is done to give a better understanding of the terms, and possibly detect poor internal consistency, suggesting that a finer mesh is in order. It is advised to concentrate on the primary load carrying mechanisms. In order to relate these two types of output in the printout, they have to be requested with both KEYOPT(6) > 1 and KEYOPT(4) = 1. Further, care must be taken to ensure that the same end of the element is being considered.

The axial reaction force based on the stress over an angle $\Delta\beta$ is:

$$F_x^r = \int_{-t/2}^{t/2} \left(\frac{(\sigma_x|_t + \sigma_x|_b)}{2} + \frac{(\sigma_x|_t - \sigma_x|_b)y}{t} \right) \Delta\beta(R_c - y\sin\phi)dy \quad (13.127)$$

or

$$F_x^r = \Delta\beta \left(\frac{(\sigma_x|_t + \sigma_x|_b)}{2} R_c t - (\sigma_x|_t - \sigma_x|_b)\sin\phi \frac{t^2}{12} \right) \quad (13.128)$$

where:

R_c = radius at midplane

t = thickness

The reaction moment based on the stress over an angle $\Delta\beta$ is:

$$M_x^r = \int_{-t/2}^{t/2} \left(\frac{(\sigma_x|_t + \sigma_x|_b)}{2} + \frac{(\sigma_x|_t - \sigma_x|_b)y}{t} \right) y\Delta\beta(R_c - y\sin\phi)dy \quad (13.129)$$

or

$$M_x^r = \Delta\beta \left(-\frac{(\sigma_x|_t + \sigma_x|_b)}{2} \frac{t^3\sin\phi}{12} + (\sigma_x|_t - \sigma_x|_b)R_c \frac{t^2}{12} \right) \quad (13.130)$$

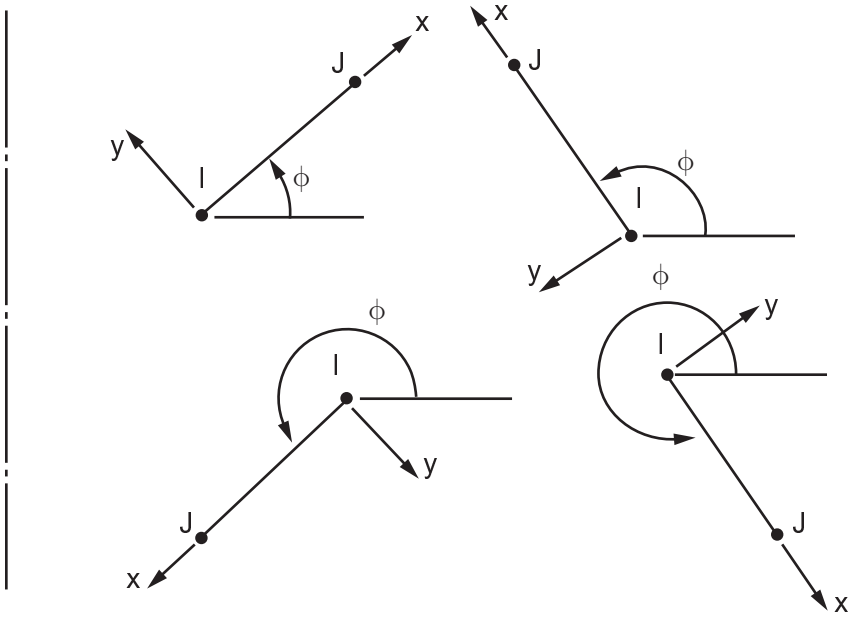
Since SHELL61 computes stiffness matrices and load vectors using the entire circumference for axisymmetric structures, $\Delta\beta = 2\pi$. Using this fact, the definition of $\sigma_x|_c$, and Equation 13.120 (p. 465) and Equation 13.123 (p. 465), Equation 13.128 (p. 466) and Equation 13.130 (p. 466) become:

$$F_x^r = 2\pi(R_c T_x - \sin\phi M_x) \tag{13.131}$$

$$M_z^r = 2\pi\left(-\frac{t^2\sin\phi}{12}T_x + R_c M_x\right) \tag{13.132}$$

As the definition of ϕ is critical for these equations, Figure 13.17: Element Orientations (p. 467) is provided to show ϕ in all four quadrants.

Figure 13.17: Element Orientations



In a uniform stress (σ_x) environment, a reaction moment will be generated to account for the greater material on the outside side. This is equivalent to moving the reaction point outward a distance y_f. y_f is computed by:

$$y_f = \frac{M_z^r}{F_x^r} \tag{13.133}$$

Using Equation 13.131 (p. 467) and Equation 13.132 (p. 467) and setting M_x to zero gives:

$$y_f = -\frac{t^2\sin\phi}{12R_c} \tag{13.134}$$

13.62. Reserved for Future Use

This section is reserved for future use.

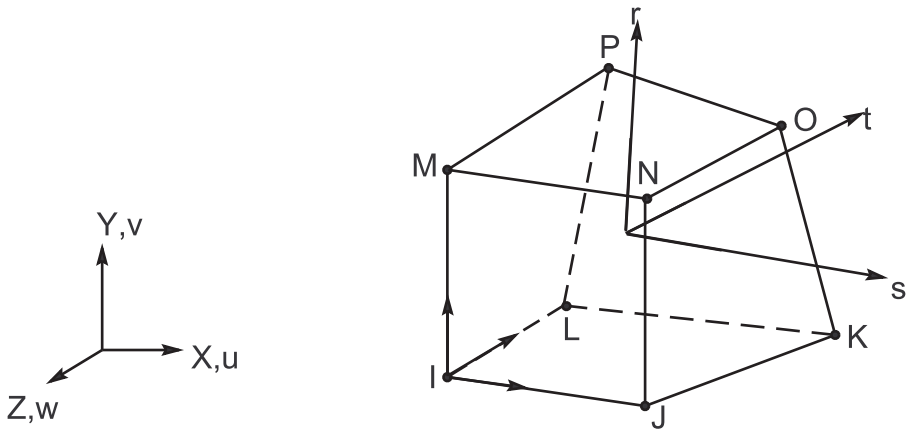
13.63. Reserved for Future Use

This section is reserved for future use.

13.64. Not Documented

No detail or element available at this time.

13.65. SOLID65 - 3-D Reinforced Concrete Solid



Matrix or Vector	Shape Functions		Integration Points
Stiffness Matrix and Thermal Load Vector	Equation 11.212 (p. 353), Equation 11.213 (p. 353), and Equation 11.214 (p. 353), or if modified extra shape functions are included (KEYOPT(1) = 0) and element has 8 unique nodes Equation 11.227 (p. 354), Equation 11.228 (p. 354), and Equation 11.229 (p. 354)		2 x 2 x 2
Mass Matrix	Equation 11.212 (p. 353), Equation 11.213 (p. 353), and Equation 11.214 (p. 353)		2 x 2 x 2
Pressure Load Vector	Quad	Equation 11.69 (p. 337) and Equation 11.70 (p. 337)	2 x 2
	Triangle	Equation 11.49 (p. 336) and Equation 11.50 (p. 336)	3

Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

References: Willam and Warnke([37] (p. 922)), Wilson([38] (p. 922)), Taylor([49] (p. 923))

13.65.1. Assumptions and Restrictions

1. Cracking is permitted in three orthogonal directions at each integration point.

2. If cracking occurs at an integration point, the cracking is modeled through an adjustment of material properties which effectively treats the cracking as a "smeared band" of cracks, rather than discrete cracks.

3. The concrete material is assumed to be initially isotropic.

4. Whenever the reinforcement capability of the element is used, the reinforcement is assumed to be "smeared" throughout the element.

5. In addition to cracking and crushing, the concrete may also undergo plasticity, with the Drucker-Prager failure surface being most commonly used. In this case, the plasticity is done before the cracking and crushing checks.

13.65.2. Description

SOLID65 allows the presence of four different materials within each element; one matrix material (e.g. concrete) and a maximum of three independent reinforcing materials. The concrete material is capable of directional integration point cracking and crushing besides incorporating plastic and creep behavior. The reinforcement (which also incorporates creep and plasticity) has uniaxial stiffness only and is assumed to be smeared throughout the element. Directional orientation is accomplished through user specified angles.

13.65.3. Linear Behavior - General

The stress-strain matrix [D] used for this element is defined as:

$$[D] = \left(1 - \sum_{i=1}^{N_r} V_i^R\right)[D^c] + \sum_{i=1}^{N_r} V_i^R [D^r]_i \tag{13.135}$$

where:

N_r = number of reinforcing materials (maximum of three, all reinforcement is ignored if M_1 is zero. Also, if M_1, M_2, or M_3 equals the concrete material number, the reinforcement with that material number is ignored)

V_i^R = ratio of volume of reinforcing material i to total volume of element (input as VRi on **R** command)

$[D^c]$ = stress-strain matrix for concrete, defined by Equation 13.136 (p. 470)

$[D^r]_i$ = stress-strain matrix for reinforcement i, defined by Equation 13.137 (p. 470)

M_1, M_2, M_3 = material numbers associated of reinforcement (input as MAT1, MAT2, and MAT3 on **R** command)

13.65.4. Linear Behavior - Concrete

The matrix $[D^c]$ is derived by specializing and inverting the orthotropic stress-strain relations defined by Equation 2.4 (p. 6) to the case of an isotropic material or

$$[D^c] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \quad (13.136)$$

where:

E = Young's modulus for concrete (input as EX on **MP** command)
ν = Poisson's ratio for concrete (input as PRXY or NUXY on **MP** command)

13.65.5. Linear Behavior - Reinforcement

The orientation of the reinforcement i within an element is depicted in Figure 13.18: Reinforcement Orientation (p. 471). The element coordinate system is denoted by (X, Y, Z) and (x_i^r, y_i^r, z_i^r) describes the coordinate system for reinforcement type i. The stress-strain matrix with respect to each coordinate system (x_i^r, y_i^r, z_i^r) has the form

$$\begin{Bmatrix} \sigma_{xx}^r \\ \sigma_{yy}^r \\ \sigma_{zz}^r \\ \sigma_{xy}^r \\ \sigma_{yz}^r \\ \sigma_{xz}^r \end{Bmatrix} = \begin{bmatrix} E_i^r & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx}^r \\ \varepsilon_{yy}^r \\ \varepsilon_{zz}^r \\ \varepsilon_{xy}^r \\ \varepsilon_{yz}^r \\ \varepsilon_{xz}^r \end{Bmatrix} = [D^r]_i \begin{Bmatrix} \varepsilon_{xx}^r \\ \varepsilon_{yy}^r \\ \varepsilon_{zz}^r \\ \varepsilon_{xy}^r \\ \varepsilon_{yz}^r \\ \varepsilon_{xz}^r \end{Bmatrix} \quad (13.137)$$

where:

E_i^r = Young's modulus of reinforcement type i (input as EX on **MP** command)

It may be seen that the only nonzero stress component is σ_{xx}^r, the axial stress in the x_i^r direction of reinforcement type i. The reinforcement direction x_i^r is related to element coordinates X, Y, Z through

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{Bmatrix} \cos\theta_i \cos\phi_i \\ \sin\theta_i \cos\phi_i \\ \sin\theta_i \end{Bmatrix} x_i^r = \begin{Bmatrix} \ell_1^r \\ \ell_2^r \\ \ell_3^r \end{Bmatrix} x_i^r \quad (13.138)$$

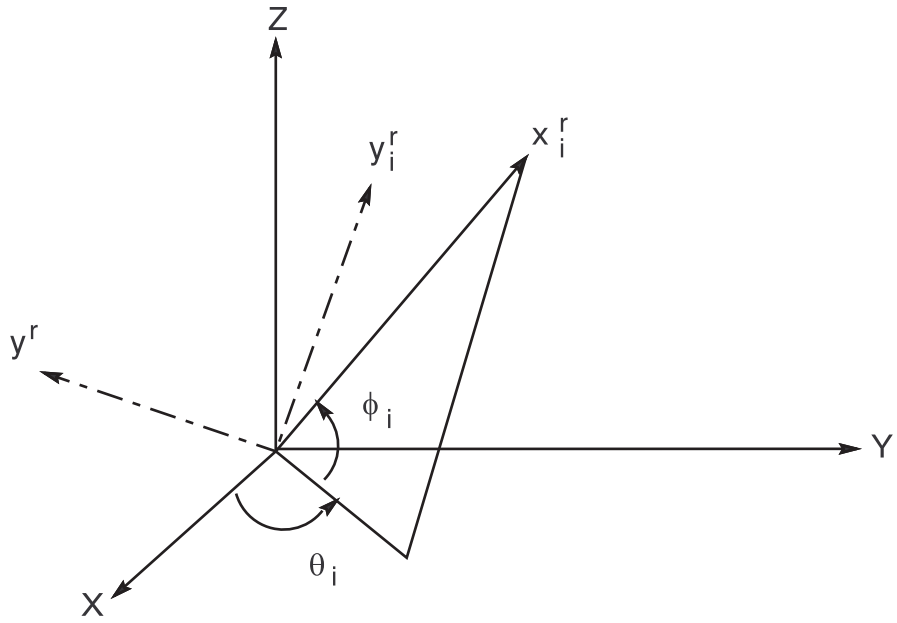
where:

θ_i = angle between the projection of the x_i^r axis on XY plane and the X axis (input as THETA1, THETA2, and THETA3 on **R** command)

ϕ_i = angle between the x_i^r axis and the XY plane (input as PHI1, PHI2, and PHI3 on **R** command)

ℓ_i^r = direction cosines between x_i^r axis and element X, Y, Z axes

Figure 13.18: Reinforcement Orientation



Since the reinforcement material matrix is defined in coordinates aligned in the direction of reinforcement orientation, it is necessary to construct a transformation of the form

$$[D^R]_i = [T^r]^T [D^r]_i [T^r] \tag{13.139}$$

in order to express the material behavior of the reinforcement in global coordinates. The form of this transformation by Schnobrich([29] (p. 922)) is

$$[T^r] = \begin{bmatrix} a_{11}^2 & a_{12}^2 & a_{13}^2 & a_{11}a_{12} & a_{12}a_{13} & a_{11}a_{13} \\ a_{21}^2 & a_{22}^2 & a_{23}^2 & a_{21}a_{22} & a_{22}a_{23} & a_{21}a_{23} \\ a_{31}^2 & a_{32}^2 & a_{33}^2 & a_{31}a_{32} & a_{32}a_{33} & a_{31}a_{33} \\ 2a_{11}a_{21} & 2a_{12}a_{22} & 2a_{13}a_{23} & a_{11}a_{22} + a_{12}a_{21} & a_{12}a_{23} + a_{13}a_{32} & a_{11}a_{23} + a_{13}a_{21} \\ 2a_{21}a_{31} & 2a_{22}a_{32} & 2a_{23}a_{33} & a_{21}a_{32} + a_{22}a_{31} & a_{22}a_{33} + a_{23}a_{32} & a_{21}a_{33} + a_{13}a_{21} \\ 2a_{11}a_{31} & 2a_{12}a_{32} & 2a_{13}a_{33} & a_{11}a_{32} + a_{12}a_{31} & a_{12}a_{33} + a_{13}a_{32} & a_{11}a_{33} + a_{13}a_{31} \end{bmatrix} \tag{13.140}$$

where the coefficients a_{ij} are defined as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} \ell_1^r & \ell_2^r & \ell_3^r \\ m_1^r & m_2^r & m_3^r \\ n_1^r & n_2^r & n_3^r \end{bmatrix} \tag{13.141}$$

The vector $\begin{bmatrix} \ell_1^r & \ell_2^r & \ell_3^r \end{bmatrix}^T$ is defined by Equation 13.138 (p. 470) while $\begin{bmatrix} m_1^r & m_2^r & m_3^r \end{bmatrix}^T$ and $\begin{bmatrix} n_1^r & n_2^r & n_3^r \end{bmatrix}^T$ are unit vectors mutually orthogonal to $\begin{bmatrix} \ell_1^r & \ell_2^r & \ell_3^r \end{bmatrix}^T$ thus defining a Cartesian coordinate referring to reinforcement directions. If the operations presented by Equation 13.139 (p. 471) are performed substituting Equation 13.137 (p. 470) and Equation 13.140 (p. 471), the resulting reinforcement material matrix in element coordinates takes the form

$$[D^r]_i = E_i^r \{A_d\}\{A_d\}^T \tag{13.142}$$

where:

$$\{A_d\} = \begin{bmatrix} a_{11}^2 & a_{21}^2 & \cdots & a_{11}^2 & a_{13}^2 \end{bmatrix}^T$$

Therefore, the only direction cosines used in $[D^R]_i$ involve the uniquely defined unit vector $\begin{bmatrix} \ell_1^r & \ell_2^r & \ell_3^r \end{bmatrix}^T$.

13.65.6. Nonlinear Behavior - Concrete

The matrix material (concrete) is capable of plasticity, creep, cracking and crushing. The concrete material model with its cracking and crushing capabilities is discussed in Concrete (p. 155). This material model predicts either elastic behavior, cracking behavior or crushing behavior. If elastic behavior is predicted, the concrete is treated as a linear elastic material (discussed above). If cracking or crushing behavior is predicted, the elastic, stress-strain matrix is adjusted as discussed below for each failure mode.

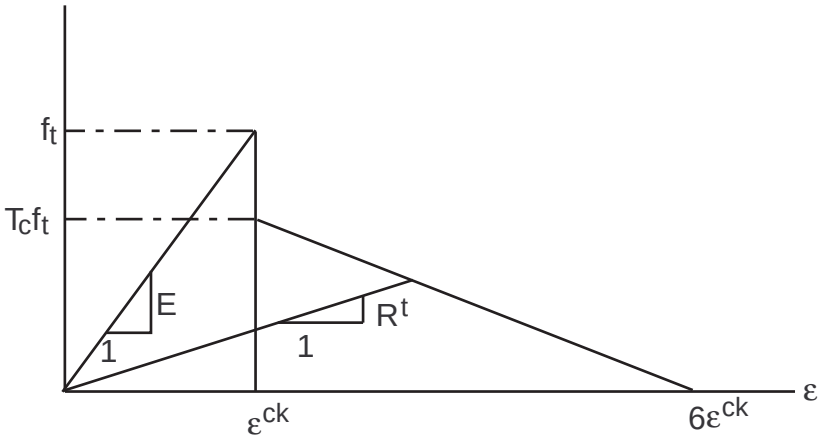
13.65.7. Modeling of a Crack

The presence of a crack at an integration point is represented through modification of the stress-strain relations by introducing a plane of weakness in a direction normal to the crack face. Also, a shear transfer coefficient β_t (constant C_1 with **TB**,CONCR) is introduced which represents a shear strength reduction factor for those subsequent loads which induce sliding (shear) across the crack face. The stress-strain relations for a material that has cracked in one direction only become:

$$[D_c^{ck}] = \frac{E}{(1+\nu)} \begin{bmatrix} \frac{R^t(1+\nu)}{E} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ 0 & \frac{\nu}{1-\nu} & \frac{1}{1-\nu} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\beta_t}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\beta_t}{2} \end{bmatrix} \quad (13.143)$$

where the superscript ck signifies that the stress strain relations refer to a coordinate system parallel to principal stress directions with the x^{ck} axis perpendicular to the crack face. If KEYOPT(7) = 0, R^t = 0.0. If KEYOPT(7) = 1, R^t is the slope (secant modulus) as defined in the figure below. R^t works with adaptive descent and diminishes to 0.0 as the solution converges.

Figure 13.19: Strength of Cracked Condition



where:

f_t = uniaxial tensile cracking stress (input as C_3 with **TB**,CONCR)
T_c = multiplier for amount of tensile stress relaxation (input as C_9 with **TB**,CONCR, defaults to 0.6)

If the crack closes, then all compressive stresses normal to the crack plane are transmitted across the crack and only a shear transfer coefficient β_c (constant C_2 with **TB**,CONCR) for a closed crack is introduced.

Then $[D_c^{ck}]$ can be expressed as

$$[D_c^{ck}] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_c \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_c \frac{(1-2\nu)}{2} \end{bmatrix} \tag{13.144}$$

The stress-strain relations for concrete that has cracked in two directions are:

$$[D_c^{ck}] = E \begin{bmatrix} \frac{R^t}{E} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{R^t}{E} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} \end{bmatrix} \tag{13.145}$$

If both directions reclose,

$$[D_c^{ck}] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_c \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_c \frac{(1-2\nu)}{2} \end{bmatrix} \tag{13.146}$$

The stress-strain relations for concrete that has cracked in all three directions are:

$$[D_c^{ck}] = E \begin{bmatrix} \frac{R^t}{E} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{R^t}{E} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} \end{bmatrix} \tag{13.147}$$

If all three cracks reclose, Equation 13.146 (p. 474) is followed. In total there are 16 possible combinations of crack arrangement and appropriate changes in stress-strain relationships incorporated in SOLID65. A note is output if $1 > \beta_c > \beta_t > 0$ are not true.

The transformation of $[D_c^{ck}]$ to element coordinates has the form

$$[D_c] = [T^{ck}]^T [D_c^{ck}][T^{ck}] \tag{13.148}$$

where $[T^{ck}]$ has a form identical to Equation 13.140 (p. 471) and the three columns of [A] in Equation 13.141 (p. 472) are now the principal direction vectors.

The open or closed status of integration point cracking is based on a strain value ε_{ck}^{ck} called the crack strain. For the case of a possible crack in the x direction, this strain is evaluated as

$$\varepsilon_{ck}^{ck} = \begin{cases} \varepsilon_x^{ck} + \dfrac{\nu}{1-\nu}\varepsilon_y^{ck} + \varepsilon_z^{ck} & \text{if no cracking has occurred} \\ \varepsilon_x^{ck} + \nu\varepsilon_z^{ck} & \text{if y direction has cracked} \\ \varepsilon_x^{ck} & \text{if y and z direction have cracked} \end{cases} \tag{13.149}$$

where:

$\varepsilon_x^{ck}, \varepsilon_y^{ck}$ and ε_z^{ck} = three normal component strains in crack orientation

The vector $\{\varepsilon^{ck}\}$ is computed by:

$$\{\varepsilon^{ck}\} = [T^{ck}]\{\varepsilon'\} \tag{13.150}$$

where:

$\{\varepsilon'\}$ = modified total strain (in element coordinates)

$\{\varepsilon'\}$, in turn, is defined as:

$$\{\varepsilon_n'\} = \{\varepsilon_{n-1}^{el}\} + \{\Delta\varepsilon_n\} - \{\Delta\varepsilon_n^{th}\} - \{\Delta\varepsilon_n^{pl}\} \tag{13.151}$$

where:

n = substep number

$\{\varepsilon_{n-1}^{el}\}$ = elastic strain from previous substep

$\{\Delta\varepsilon_n\}$ = total strain increment (based on $\{\Delta u_n\}$, the displacement increment over the substep)

$\{\Delta\varepsilon_n^{th}\}$ = thermal strain increment

$\{\Delta\varepsilon_n^{pl}\}$ = plastic strain increment

If ε_{ck}^{ck} is less than zero, the associated crack is assumed to be closed.

If ε_{ck}^{ck} is greater than or equal to zero, the associated crack is assumed to be open. When cracking first occurs at an integration point, the crack is assumed to be open for the next iteration.

13.65.8. Modeling of Crushing

If the material at an integration point fails in uniaxial, biaxial, or triaxial compression, the material is assumed to crush at that point. In SOLID65, crushing is defined as the complete deterioration of the structural integrity of the material (e.g. material spalling). Under conditions where crushing has occurred, material strength is assumed to have degraded to an extent such that the contribution to the stiffness of an element at the integration point in question can be ignored.

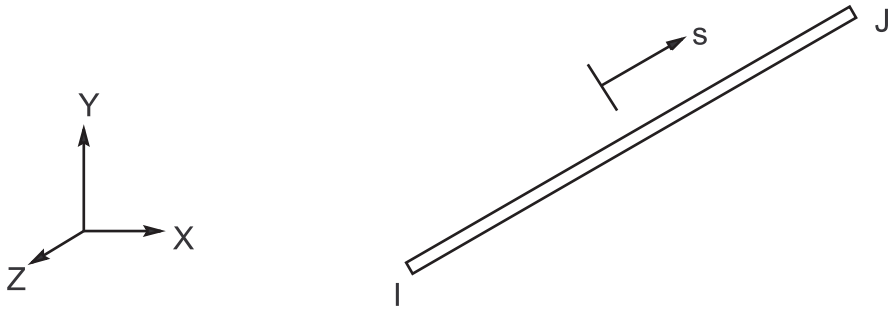
13.66. Reserved for Future Use

This section is reserved for future use.

13.67. Reserved for Future Use

This section is reserved for future use.

13.68. LINK68 - Coupled Thermal-Electric Line



Matrix or Vector	Shape Functions	Integration Points
Electrical Conductivity Matrices	Equation 11.14 (p. 330)	None
Thermal Conductivity and Specific Heat Matrices; and Heat Generation Load Vector	Equation 11.13 (p. 330)	None

Reference: Kohnke and Swanson([19] (p. 921))

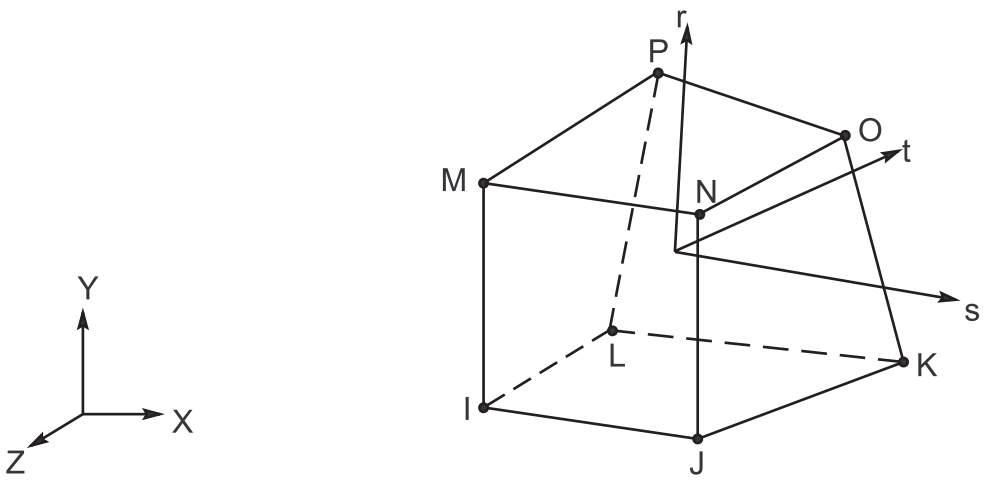
13.68.1. Other Applicable Sections

Coupling (p. 293) discusses coupled effects.

13.69. Reserved for Future Use

This section is reserved for future use.

13.70. SOLID70 - 3-D Thermal Solid



Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Equation 11.222 (p. 353)	2 x 2 x 2
Specific Heat Matrix	Equation 11.222 (p. 353). Matrix is diagonalized as described in Lumped Matrices (p. 391)	Same as conductivity matrix
Convection Surface Matrix and Load Vector	Equation 11.222 (p. 353) specialized to the face	2 x 2

13.70.1. Other Applicable Sections

Derivation of Heat Flow Matrices (p. 235) has a complete derivation of the matrices and load vectors of a general thermal analysis element. Mass transport is discussed in PLANE55 - 2-D Thermal Solid (p. 462).

13.70.2. Fluid Flow in a Porous Medium

An option (KEYOPT(7) = 1) is available to convert SOLID70 to a nonlinear steady-state fluid flow element. Pressure is the variable rather than temperature. From Equation 6.22 (p. 235), the element conductivity matrix is:

$$[K_e^{tb}] = \int_{vol} [B]^T [D][B] d(vol) \tag{13.152}$$

[B] is defined by Equation 6.22 (p. 235) and for this option, [D] is defined as:

$$[D]=\begin{bmatrix} \frac{K_x^{\infty}\rho}{\mu+K_x^{\infty}E} & 0 & 0 \\ 0 & \frac{K_y^{\infty}\rho}{\mu+K_y^{\infty}E} & 0 \\ 0 & 0 & \frac{K_z^{\infty}\rho}{\mu+K_z^{\infty}E} \end{bmatrix} \tag{13.153}$$

where:

K_x^{∞} = absolute permeability of the porous medium in the x direction (input as KXX on **MP** command)
ρ = mass density of the fluid (input as DENS on **MP** command)
μ = viscosity of the fluid (input as VISC on **MP** command)

$$E=\rho\beta S^{\alpha}$$

β = visco-inertial parameter of the fluid (input as C on **MP** command)
S = seepage velocity (at centroid from previous iteration, defined below)
α = empirical exponent on S (input as MU on **MP** command)

For this option, no "specific heat" matrix or "heat generation" load vector is computed.

The pressure gradient components are computed by:

$$\begin{Bmatrix} g_x^p \\ g_y^p \\ g_z^p \end{Bmatrix}=[B]\{T_e\} \tag{13.154}$$

where:

g_x^p = pressure gradient in the x-direction (output as PRESSURE GRADIENT (X))
$\{T_e\}$ = vector of element temperatures (pressures)

The pressure gradient is computed from:

$$g^p=\sqrt{(g_x^p)^2+(g_y^p)^2+(g_z^p)^2} \tag{13.155}$$

where:

g^p = total pressure gradient (output as PRESSURE GRADIENT (TOTAL))

The mass flux components are:

$$\begin{Bmatrix} f_x \\ f_y \\ f_z \end{Bmatrix}=-[D]\begin{Bmatrix} g_x^p \\ g_y^p \\ g_z^p \end{Bmatrix} \tag{13.156}$$

The vector sum of the mass flux components is:

$$f = \sqrt{f_x^2 + f_y^2 + f_z^2} \tag{13.157}$$

where:

f = mass flux (output as MASS FLUX)

The fluid velocity components are:

$$\begin{Bmatrix} S_x \\ S_y \\ S_z \end{Bmatrix} = \frac{1}{\rho} \begin{Bmatrix} f_x \\ f_y \\ f_z \end{Bmatrix} \tag{13.158}$$

where:

S_x = fluid velocity in the x-direction (output as FLUID VELOCITY (X))

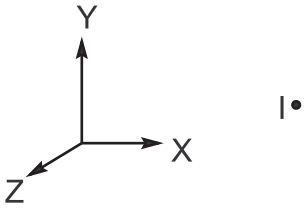
and the maximum fluid velocity is:

$$S = \frac{f}{\rho} \tag{13.159}$$

where:

S = total fluid velocity (output as FLUID VELOCITY (TOTAL))

13.71. MASS71 - Thermal Mass



Matrix or Vector	Shape Functions	Integration Points
Specific Heat Matrix and Heat Generation Load Vector	None	None

13.71.1. Specific Heat Matrix

The specific heat matrix for this element is simply:

$$[C_e^t] = [C^o] \tag{13.160}$$

C^o is defined as:

$$C^o = \begin{cases} \rho C_p(vol) & \text{if KEYOPT(3) = 0} \\ C_a & \text{if KEYOPT(3) = 1} \end{cases} \tag{13.161}$$

where:

ρ = density (input as DENS on **MP** command)
C_p = specific heat (input as C on **MP** command)
vol = volume (input as CON1 on **R** command)
C_a = capacitance (input as CON1 on **R** command)

13.71.2. Heat Generation Load Vector

The heat generation load vector is:

$$\{Q_e^g\} = \{A_q\} \tag{13.162}$$

where:

$$A_q = \begin{cases} Q_R & \text{if } A_1 \text{ thru } A_6 \text{ are not provided} \\ A_1 + A_2 T + A_3 T^{A_4} + A_5 T^{A_6} & \text{if } A_1 \text{ thru } A_6 \text{ are provided} \end{cases}$$

Q_R = heat rate (input as QRATE on **MP** command)
A_1, A_2, etc. = constants (input as A1, A2, etc. on **R** command)

$T = T_\ell + T_o$ = absolute temperature

$$T_\ell = \begin{cases} T_{unif} & \text{for first iteration} \\ T'_\ell & \text{for second and subsequent iterations} \end{cases}$$

T_{unif} = uniform temperature (input on **BFUNIF** command)

T'_ℓ = temperature from previous iteration
T_o = offset temperature (input on **TOFFST** command)

13.72. Reserved for Future Use

This section is reserved for future use.

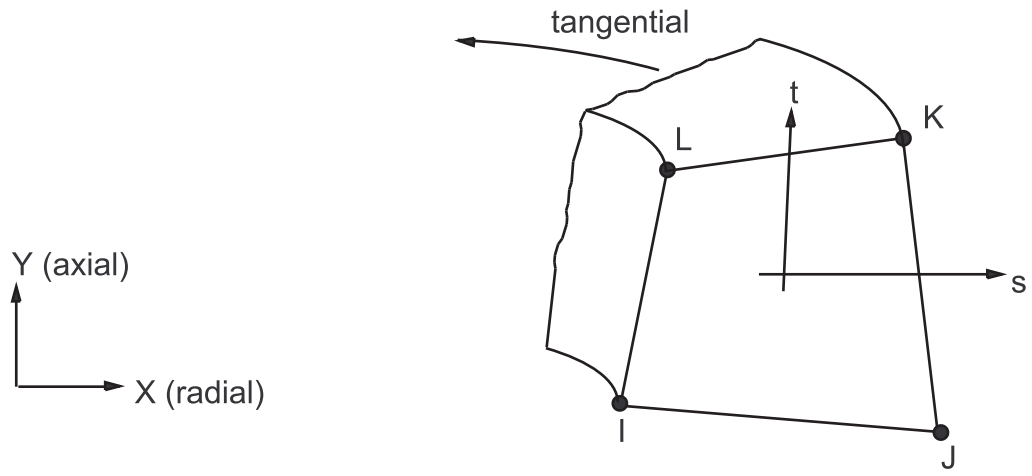
13.73. Reserved for Future Use

This section is reserved for future use.

13.74. Not Documented

No detail or element available at this time.

13.75. PLANE75 - Axisymmetric-Harmonic 4-Node Thermal Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Quad	Equation 11.162 (p. 346)	2 x 2
	Triangle	Equation 11.154 (p. 345)	3
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Lumped Matrices (p. 391)		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Same as conductivity matrix specialized to the face		2

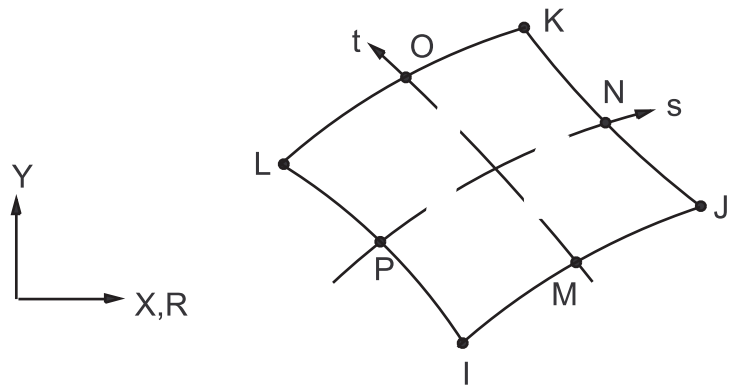
13.75.1. Other Applicable Sections

Heat Flow (p. 227) describes the derivation of the element matrices and load vectors as well as heat flux evaluations.

13.76. Reserved for Future Use

This section is reserved for future use.

13.77. PLANE77 - 2-D 8-Node Thermal Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Quad	Equation 11.138 (p. 342)	3 x 3
	Triangle	Equation 11.117 (p. 341)	6
Specific Heat Matrix	Same as conductivity matrix. If KEYOPT(1) = 1, matrix is diagonalized as described in Lumped Matrices (p. 391)		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Same as conductivity matrix, specialized to the face		2

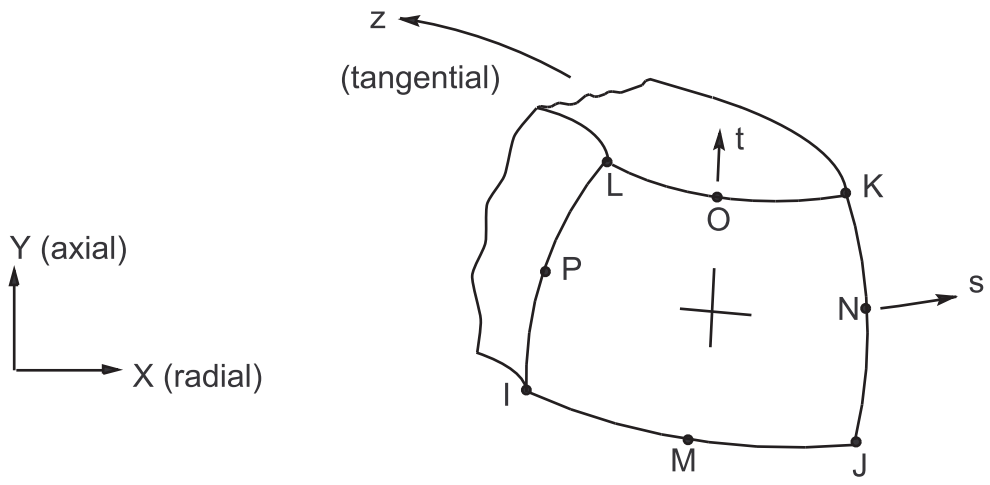
13.77.1. Other Applicable Sections

Heat Flow (p. 227) describes the derivation of the thermal element matrices and load vectors as well as heat flux evaluations. If KEYOPT(1) = 1, the specific heat matrix is diagonalized as described in Lumped Matrices (p. 391).

13.77.2. Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the temperature varies linearly along that edge.

13.78. PLANE78 - Axisymmetric-Harmonic 8-Node Thermal Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Quad	Equation 11.169 (p. 347)	3 x 3
	Triangle	Equation 11.158 (p. 346)	6
Specific Heat Matrix	Same as conductivity matrix. If KEYOPT(1) = 1, matrix is diagonalized as described in Lumped Matrices (p. 391)		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Same as stiffness matrix, specialized to the face		2

13.78.1. Other Applicable Sections

Heat Flow (p. 227) describes the derivation of the thermal element matrices and load vectors as well as heat flux evaluations.

13.78.2. Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the temperature varies linearly along that edge.

13.79. Reserved for Future Use

This section is reserved for future use.

13.80. Reserved for Future Use

This section is reserved for future use.

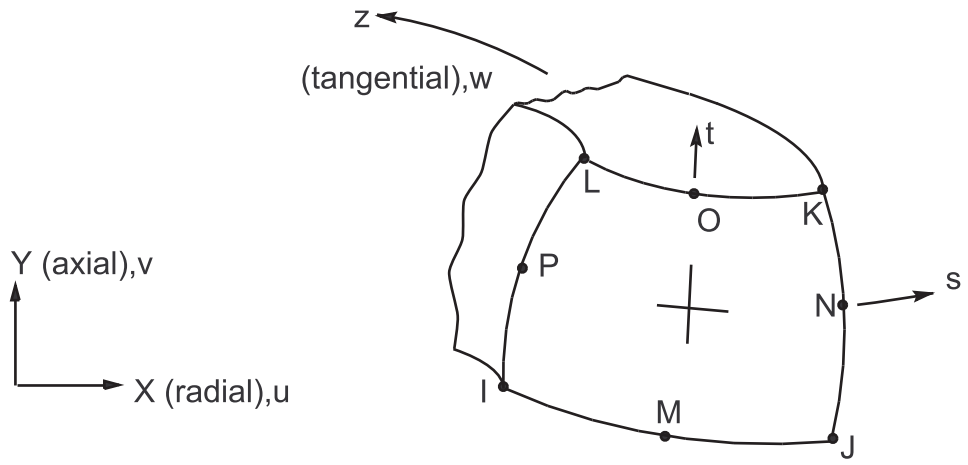
13.81. Reserved for Future Use

This section is reserved for future use.

13.82. Reserved for Future Use

This section is reserved for future use.

13.83. PLANE83 - Axisymmetric-Harmonic 8-Node Structural Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness, Mass, and Stress Stiffness Matrices; and Thermal Load Vector	Quad	Equation 11.166 (p. 346), Equation 11.167 (p. 346), and Equation 11.168 (p. 347)	2 x 2
	Triangle	Equation 11.155 (p. 345), Equation 11.156 (p. 345), and Equation 11.157 (p. 345)	3
Pressure Load Vector	Same as stiffness matrix, specialized to the face		2

Load Type	Distribution
Element Temperature	Same as shape functions across element, harmonic around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face, harmonic around circumference

Reference: Zienkiewicz([39] (p. 922))

13.83.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. PLANE25 - Axisymmetric-Harmonic 4-Node Structural Solid (p. 425) has a discussion of temperature applicable to this element.

13.83.2. Assumptions and Restrictions

A dropped midside node implies that the edge is and remains straight.

The material properties are assumed to be constant around the entire circumference, regardless of temperature-dependent material properties or loading. For ℓ (input as MODE on **MODE** command) > 0, extreme values for combined stresses are obtained by computing these stresses at every 10/ℓ degrees and selecting the extreme values.

13.84. Not Documented

No detail or element available at this time.

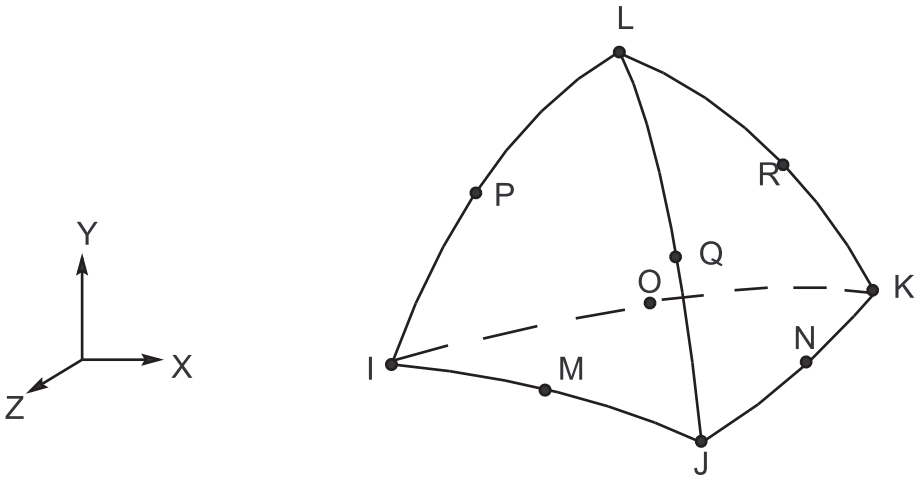
13.85. Reserved for Future Use

This section is reserved for future use.

13.86. Not Documented

No detail or element available at this time.

13.87. SOLID87 - 3-D 10-Node Tetrahedral Thermal Solid



Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Equation 11.185 (p. 348)	4
Specific Heat Matrix	Same as conductivity matrix. If KEYOPT(1) = 1, the matrix is diagonalized as described in Lumped Matrices (p. 391)	11
Convection Surface Matrix and Load Vector	Equation 11.185 (p. 348) specialized to the face. Diagonalized surface matrix if KEYOPT(5) = 0, consistent surface matrix if KEYOPT(5) = 1	6

13.87.1. Other Applicable Sections

Heat Flow (p. 227) describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. If KEYOPT(1) = 1, the specific heat matrix is diagonalized as described in Lumped Matrices (p. 391).

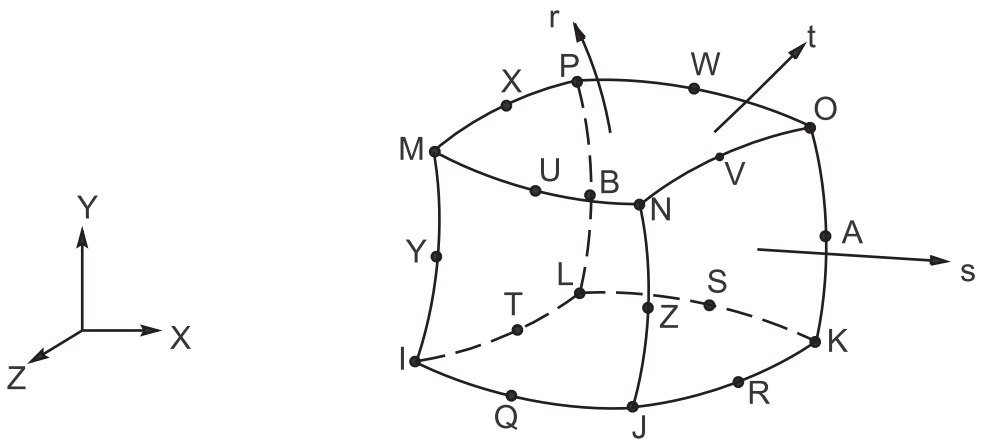
13.88. Reserved for Future Use

This section is reserved for future use.

13.89. Reserved for Future Use

This section is reserved for future use.

13.90. SOLID90 - 3-D 20-Node Thermal Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Brick	Equation 11.233 (p. 355)	14
	Wedge	Equation 11.209 (p. 352)	3 x 3
	Pyramid	Equation 11.193 (p. 350)	2 x 2 x 2
	Tet	Equation 11.185 (p. 348)	4
Specific Heat Matrix	Same as conductivity matrix. If KEYOPT(1) = 1, the matrix is diagonalized as described in Lumped Matrices (p. 391).		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Quad	Equation 11.91 (p. 338)	3 x 3
	Triangle	Equation 11.63 (p. 337)	6

13.90.1. Other Applicable Sections

Heat Flow (p. 227) describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations.

13.91. Reserved for Future Use

This section is reserved for future use.

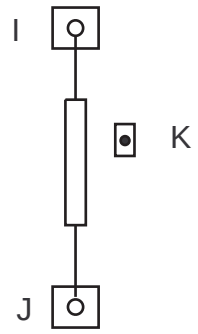
13.92. Reserved for Future Use

This section is reserved for future use.

13.93. Reserved for Future Use

This section is reserved for future use.

13.94. CIRCU94 - Piezoelectric Circuit



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (lumped)	None
Damping Matrix	None (lumped, harmonic analysis only)	None
Load Vector	None (lumped)	None

The piezoelectric circuit element, CIRCU94, simulates basic linear electric circuit components that can be directly connected to the piezoelectric FEA domain. For details about the underlying theory, see Wang and Ostergaard([323] (p. 939)). It is suitable for the simulation of circuit-fed piezoelectric transducers, piezoelectric dampers for vibration control, crystal filters and oscillators etc.

13.94.1. Electric Circuit Elements

CIRCU94 contains 5 linear electric circuit element options:

a. Resistor	(KEYOPT(1) = 0)
b. Inductor	(KEYOPT(1) = 1)
c. Capacitor	(KEYOPT(1) = 2)
d. Current Source	(KEYOPT(1) = 3)
e. Voltage Source	(KEYOPT(1) = 4)

Options a, b, c, d are defined by two nodes I and J (see figure above), each node having a VOLT DOF. The voltage source is also characterized by a third node K with CURR DOF to represent an auxiliary charge variable.

13.94.2. Piezoelectric Circuit Element Matrices and Load Vectors

The finite element equations for the resistor, inductor, capacitor and current source of CIRCU94 are derived using the nodal analysis method (McCalla([188] (p. 931))) that enforces Kirchhoff's Current Law (KCL) at each circuit node. To be compatible with the system of piezoelectric finite element equations (see Piezoelectrics (p. 313)), the nodal analysis method has been adapted to maintain the charge balance at each node:

$$[K]\{V\} = \{Q\} \tag{13.163}$$

where:

[K] = stiffness (capacitance) matrix
{V} = vector of nodal voltages (to be determined)
{Q} = load vector of nodal charges

The voltage source is modeled using the modified nodal analysis method (McCalla([188] (p. 931))) in which the set of unknowns is extended to include electric charge at the auxiliary node K, while the corresponding entry of the load vector is augmented by the voltage source amplitude. In a transient analysis, different integration schemes are employed to determine the vector of nodal voltages.

For a resistor, the generalized trapezoidal rule is used to approximate the charge at time step n+1 thus yielding:

$$[K] = \frac{\theta\Delta t}{R}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \text{stiffness matrix} \tag{13.164}$$

$$\{V\} = \begin{Bmatrix} V_I^{n+1} \\ V_J^{n+1} \end{Bmatrix} = \text{nodal voltages} \tag{13.165}$$

$$\{Q\} = \begin{Bmatrix} -Q_R^{n+1} \\ Q_R^{n+1} \end{Bmatrix} = \text{element vector charge} \tag{13.166}$$

where:

θ = first order time integration parameter (input on **TINTP** command)
Δt = time increment (input on **DELTIM** command)
R = resistance

$$Q_R^{n+1} = (1-\theta) i_R^n \Delta t + q_R^n$$

$$q_R^{n+1} = \theta i_R^{n+1} \Delta t + (1-\theta) i_R^n \Delta t + q_R^n$$

$$i_R^{n+1} = \frac{V_I^{n+1} - V_J^{n+1}}{R}$$

The constitutive equation for an inductor is of second order with respect to the charge time-derivative, and therefore the Newmark integration scheme is used to derive its finite element equation:

$$[K] = \frac{\alpha\Delta t^2}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \text{stiffness matrix} \tag{13.167}$$

$$\{Q\} = \begin{Bmatrix} -Q_L^{n+1} \\ Q_L^{n+1} \end{Bmatrix} = \text{vector charge} \tag{13.168}$$

where:

L = inductance

$$Q_L^{n+1} = \left(\frac{1}{2} - \alpha\right)\frac{\Delta t^2}{L}(V_I^n - V_J^n) + i_L^n \Delta t + q_L^n$$

$$q_L^{n+1} = \alpha\frac{\Delta t^2}{L}(V_I^{n+1} - V_J^{n+1}) + \left(\frac{1}{2} - \alpha\right)\frac{\Delta t^2}{L}(V_I^n - V_J^n) + i_L^n \Delta t + q_L^n$$

$$i_L^{n+1} = \delta\frac{\Delta t}{L}(V_I^{n+1} - V_J^{n+1}) + (1-\delta)\frac{\Delta t}{L}(V_I^n - V_J^n) + i_L^n$$

α, δ = Newmark integration parameters (input on **TINTP** command

A capacitor with nodes I and J is represented by

$$[K] = C\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \text{stiffness matrix} \tag{13.169}$$

$$\{Q\} = \begin{Bmatrix} -Q_C^{n+1} \\ Q_C^{n+1} \end{Bmatrix} = \text{charge vector} \tag{13.170}$$

where:

C = capacitance

$$Q_C^{n+1} = -C(V_I^n - V_J^n) + q_C^n$$

$$q_C^{n+1} = C(V_I^{n+1} - V_J^{n+1}) - C(V_I^n - V_J^n) + q_C^n$$

For a current source, the [K] matrix is a null matrix, while the charge vector is updated at each time step as

$$\{Q\} = \begin{Bmatrix} -Q_S^{n+1} \\ Q_S^{n+1} \end{Bmatrix} \tag{13.171}$$

where:

$$Q_S^{n+1} = \theta\Delta t I_S^{n+1} + (1-\theta)\Delta t I_S^n + Q_S^n$$

I_S^{n+1} = source current at time t_{n+1}

Note that for the first substep of the first load step in a transient analysis, as well as on the transient analysis restart, all the integration parameters (θ, α, δ) are set to 1. For every subsequent substep/load step, ANSYS uses either the default integration parameters or their values input using the **TINTP** command.

In a harmonic analysis, the time-derivative is replaced by jω, which produces

$$[K] = j\omega\left(-\frac{1}{\omega^2 R}\right)\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.172}$$

for a resistor,

$$[K] = \left(-\frac{1}{\omega^2 L}\right)\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.173}$$

for an inductor, and

$$[K] = C\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \text{capacitor} \tag{13.174}$$

where:

j = imaginary unit
ω = angular frequency (input on **HARFRQ** command)

The element charge vector {Q} is a null vector for all of the above components.

For a current source, the [K] matrix is a null matrix and the charge vector is calculated as

$$\{Q\} = \begin{Bmatrix} -Q_S \\ Q_S \end{Bmatrix} \tag{13.175}$$

where:

$$Q_S = \frac{1}{j\omega} I_S e^{j\phi}$$

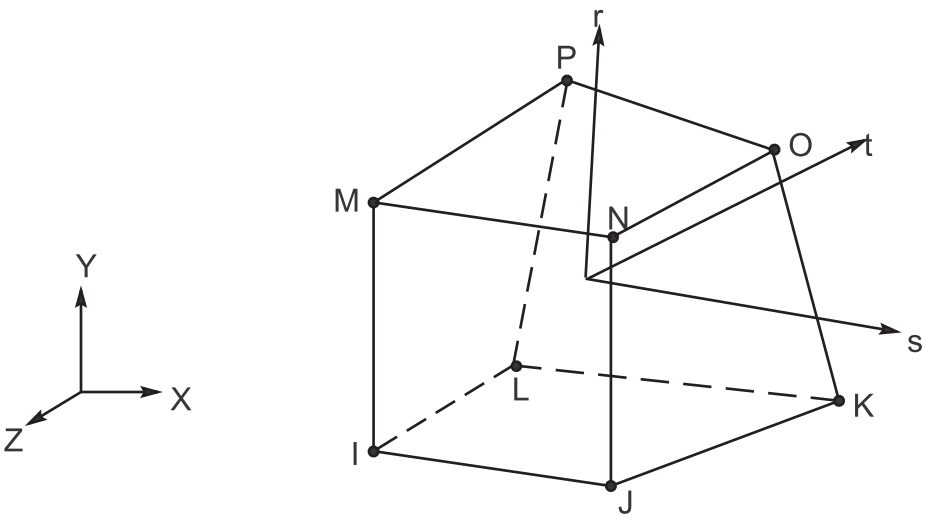
I_S = source current amplitude
ϕ = source current phase angle (in radians)

Note that all of the above matrices and load vectors are premultiplied by -1 before being assembled with the piezoelectric finite element equations that use negative electric charge as a through variable (reaction "force") for the VOLT degree of freedom.

13.95. Reserved for Future Use

This section is reserved for future use.

13.96. SOLID96 - 3-D Magnetic Scalar Solid



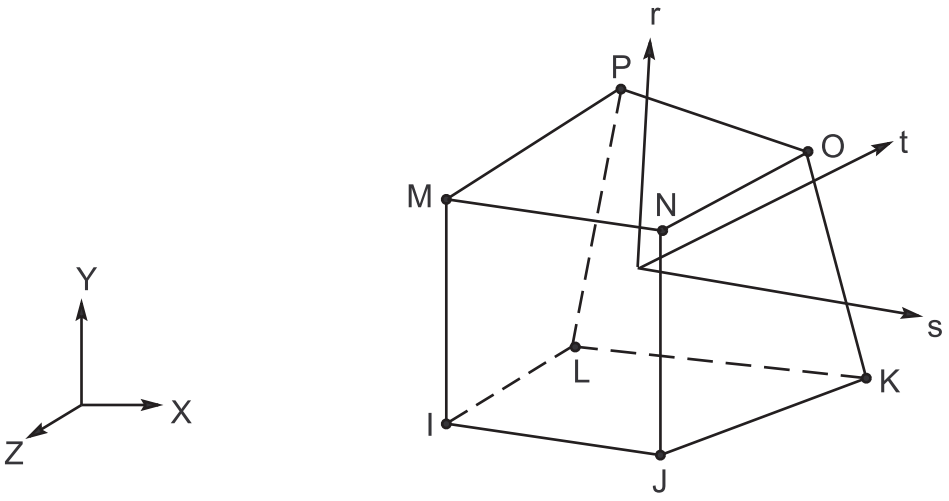
Matrix or Vector	Shape Functions	Integration Points
Magnetic Scalar Potential Coefficient Matrix; and Load Vector of Magnetism due to Permanent Magnets, and Source Currents	Equation 11.224 (p. 353)	2 x 2 x 2

References: Coulomb([76] (p. 924)), Mayergoyz([119] (p. 927)), Gyimesi([141] (p. 928),[149] (p. 929))

13.96.1. Other Applicable Sections

Derivation of Electromagnetic Matrices (p. 193) discusses the magnetic scalar potential method used by this element.

13.97. SOLID97 - 3-D Magnetic Solid



Matrix or Vector	Shape Functions	Integration Points
Magnetic Vector Potential Coefficient Matrix and Load Vector of Magnetism due to Source Currents, Permanent Magnets, and Applied Currents	Equation 11.215 (p. 353), Equation 11.216 (p. 353), and Equation 11.217 (p. 353)	2 x 2 x 2
Electric Potential Coefficient Matrix	Equation 11.223 (p. 353)	2 x 2 x 2

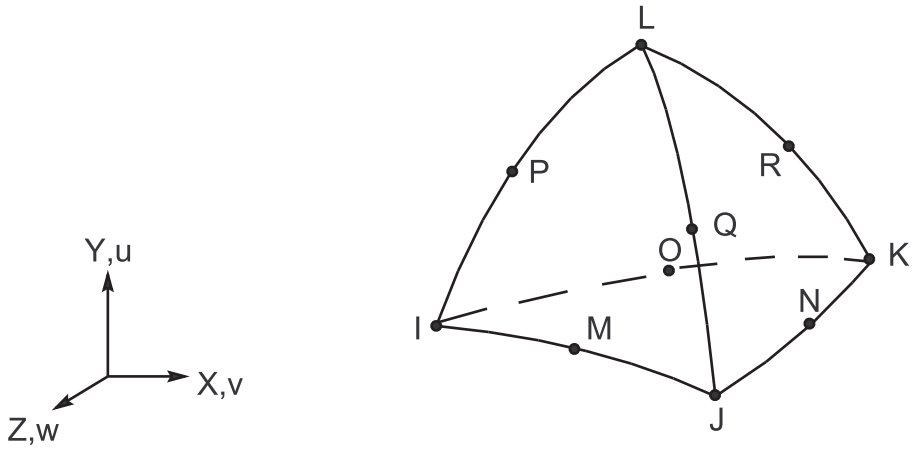
Load Type	Distribution
Current Density, Voltage Load and Phase Angle Distribution	Trilinearly thru element

References: Coulomb([76] (p. 924)), Mohammed([118] (p. 927)), Biro et al.([120] (p. 927))

13.97.1. Other Applicable Sections

Derivation of Electromagnetic Matrices (p. 193) and Electromagnetic Field Evaluations (p. 200) contain a discussion of the 2-D magnetic vector potential formulation which is similar to the 3-D formulation of this element.

13.98. SOLID98 - Tetrahedral Coupled-Field Solid



Matrix or Vector	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix	Equation 11.187 (p. 348)	4
Electric Conductivity Matrix	Equation 11.186 (p. 348)	4
Thermal Conductivity Matrix	Equation 11.185 (p. 348)	4
Stiffness and Mass Matrices; and Thermal Expansion Load Vector	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348)	4
Piezoelectric Coupling Matrix	Same as combination of stiffness matrix and conductivity matrix	4

Matrix or Vector	Shape Functions	Integration Points
Specific Heat Matrix	Same as conductivity matrix. If KEYOPT(3) = 1, matrix is diagonalized as described in Lumped Matrices (p. 391)	11
Load Vector due to Imposed Thermal and Electric Gradients, Heat Generation, Joule Heating, Magnetic Forces, Permanent Magnet and Magnetism due to Source Currents	Same as coefficient or conductivity matrix	4
Load Vector due to Convection and Pressures	Same as stiffness or conductivity matrix, specialized to the face	6

References: Zienkiewicz([39] (p. 922)), Coulomb([76] (p. 924)), Mayergoyz([119] (p. 927)), Gyimesi([141] (p. 928))

13.98.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. Heat Flow (p. 227) describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. Derivation of Electromagnetic Matrices (p. 193) describes the scalar potential method, which is used by this element. Piezoelectrics (p. 313) discusses the piezoelectric capability used by the element. If KEYOPT(3) = 1, the specific heat matrix is diagonalized as described in Lumped Matrices (p. 391). Also, Thermoelectrics (p. 319) discusses the thermoelectric capability.

13.99. Reserved for Future Use

This section is reserved for future use.

13.100. Reserved for Future Use

This section is reserved for future use.

13.101. Reserved for Future Use

This section is reserved for future use.

13.102. Reserved for Future Use

This section is reserved for future use.

13.103. Reserved for Future Use

This section is reserved for future use.

13.104. Reserved for Future Use

This section is reserved for future use.

13.105. Reserved for Future Use

This section is reserved for future use.

13.106. Reserved for Future Use

This section is reserved for future use.

13.107. Reserved for Future Use

This section is reserved for future use.

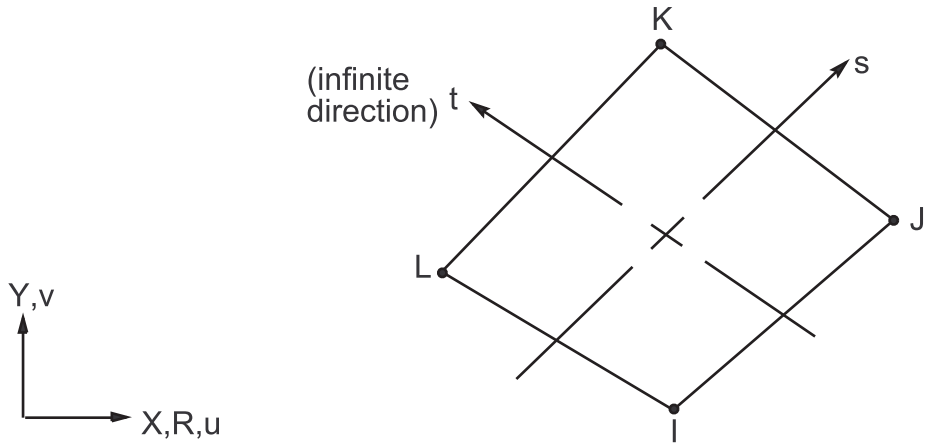
13.108. Reserved for Future Use

This section is reserved for future use.

13.109. Reserved for Future Use

This section is reserved for future use.

13.110. INFIN110 - 2-D Infinite Solid



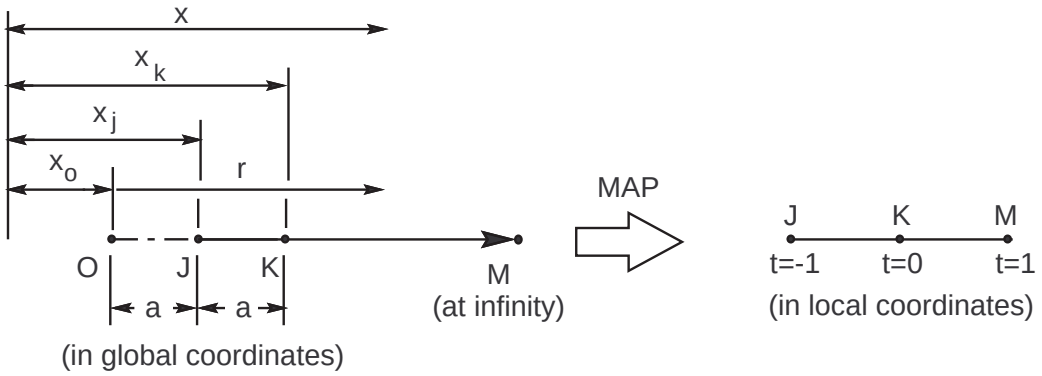
Matrix or Vector	Mapping and Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix	Equation 11.141 (p. 343), Equation 11.144 (p. 343), and Equation 11.145 (p. 343)	2 x 2
Thermal Conductivity and Specific Heat Matrices	Equation 11.142 (p. 343), Equation 11.144 (p. 343), and Equation 11.145 (p. 343)	2 x 2
Dielectric Permittivity and Electrical Conductivity Coefficient Matrices	Equation 11.143 (p. 343), Equation 11.144 (p. 343), and Equation 11.145 (p. 343)	2 x 2

References: Zienkiewicz et al.([169] (p. 930)), Damjanic' and Owen([170] (p. 930)), Marques and Owen([171] (p. 930)), Li et al.([172] (p. 930))

13.110.1. Mapping Functions

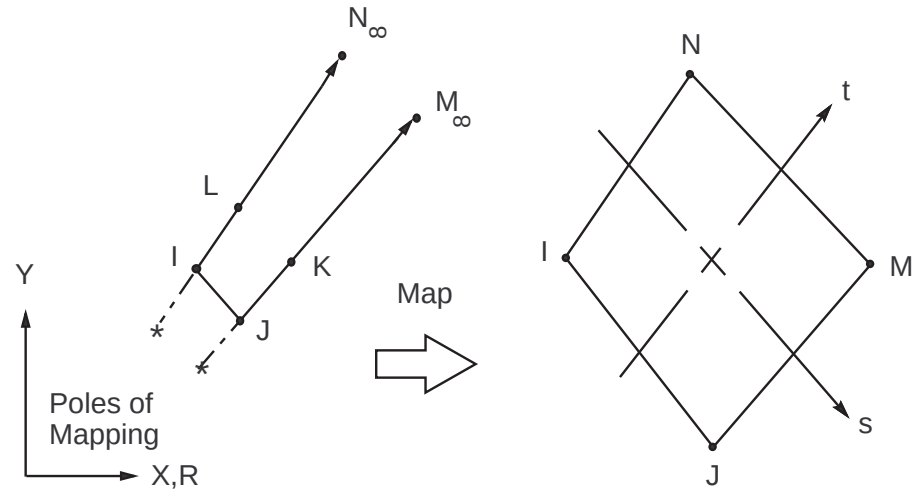
The theory for the infinite mapping functions is briefly summarized here. Consider the 1-D situation shown below:

Figure 13.20: Global to Local Mapping of a 1-D Infinite Element



The 1-D element may be thought of as one edge of the infinite element of Figure 13.21: Mapping of 2-D Solid Infinite Element (p. 495). It extends from node J, through node K to the point M at infinity and is mapped onto the parent element defined by the local coordinate system in the range -1 $\leq$ t $\leq$ 1.

Figure 13.21: Mapping of 2-D Solid Infinite Element



The position of the "pole", x_o, is arbitrary, and once chosen, the location of node K is defined by

$$x_K = 2x_J x_o \tag{13.176}$$

The interpolation from local to global positions is performed as

$$x(t) = M_J(t)x_J + M_K(t)x_K \tag{13.177}$$

where:

$M_J(t) = -2t/(1 - t)$

$M_K(t) = 1 - M_J(t)$

Examining the above mapping, it can be seen that t = -1, 0, 1 correspond respectively to the global positions x = x_J, x_K, ∞, respectively.

The basic field variable is:

$$A = \begin{cases} A_z & \text{Magnetic Vector Potential (accessed with KEYOPT(1) = 0)} \\ V & \text{Volt (accessed with KEYOPT(1) = 1 or 3)} \\ T & \text{Temperature (accessed with KEYOPT(1) = 2)} \end{cases}$$

and can be interpolated using standard shape functions, which when written in polynomial form becomes

$$A(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + - - - - \tag{13.178}$$

Solving Equation 13.177 (p. 495) for t yields

$$t = 1 - \frac{2a}{r} \tag{13.179}$$

where:

r = distance from the pole, O, to a general point within the element
a = x_K - x_J as shown in Figure 13.21: Mapping of 2-D Solid Infinite Element (p. 495)

Substituting Equation 13.179 (p. 496) into Equation 13.178 (p. 496) gives

$$A(t) = c_0 + \frac{c_1}{r} + \frac{c_2}{r^2} + \frac{c_3}{r^3} + - - - - - \tag{13.180}$$

Where c_0 = 0 is implied since the variable A is assumed to vanish at infinity.

Equation 13.180 (p. 496) is truncated at the quadratic (r^2) term in the present implementation. Equation 13.180 (p. 496) also shows the role of the pole position, O.

In 2-D (Figure 13.21: Mapping of 2-D Solid Infinite Element (p. 495)) mapping is achieved by the shape function products. The mapping functions and the Lagrangian isoparametric shape functions for 2-D and axisymmetric 4 node quadrilaterals are given in 2-D and Axisymmetric 4-Node Quadrilateral Infinite Solids (p. 343). The shape functions for the nodes M and N are not needed as the field variable, A, is assumed to vanish at infinity.

13.110.2. Matrices

The coefficient matrix can be written as:

$$[K_e] = \int_{vol} [B]^T [D][B] \, d(vol) \tag{13.181}$$

with the terms defined below:

1. Magnetic Vector Potential (accessed with KEYOPT(1) = 0)

   $[K_e]$ = magnetic potential coefficient matrix

$$[D] = \frac{1}{\mu_o}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

μ_o = magnetic permeability of free space (input on **EMUNIT** command)

The infinite elements can be used in magnetodynamic analysis even though these elements do not compute mass matrices. This is because air has negligible conductivity.

2. Electric Potential (Electric Charge) (accessed with KEYOPT(1) = 1)

$[K_e]$ = dielectric permittivity matrix

$$[D] = \begin{bmatrix} \varepsilon_x & 0 \\ 0 & \varepsilon_y \end{bmatrix}$$

ε_x, ε_y = dielectric permittivity (input as PERX and PERY on **MP** command)

$$[C_e] = \text{electrical conductivity matrix} = \int_{vol} [B]^T [C][B]\, d(vol)$$

$$[C] = \begin{bmatrix} \sigma_x^{eff} & 0 \\ 0 & \sigma_y^{eff} \end{bmatrix}$$

$\sigma_x^{eff}, \sigma_y^{eff}$ = effective electrical conductivity (defined by Equation 5.86 (p. 192))

3. Temperature (accessed with KEYOPT(1) = 2)

$[K_e]$ = thermal conductivity matrix

$$[D] = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix}$$

k_x, k_y = thermal conductivities in the x and y direction (input as KXX and KYY on **MP** command)

$$[C_e] = \text{specific heat matrix} = \int_{vol} C_c \{N\}\{N\}^T\, d(vol)$$

$C_c = \rho\, C_p$
ρ = density of the fluid (input as DENS on **MP** command)
C_p = specific heat of the fluid (input as C on **MP** command)
{N} = shape functions given in 2-D and Axisymmetric 4-Node Quadrilateral Infinite Solids (p. 343)

4. Electric Potential (Electric Current) (accessed with KEYOPT(1) = 3)

$[K_e]$ = electrical conductivity matrix

$$[D] = \begin{bmatrix} \sigma_x^{eff} & 0 \\ 0 & \sigma_y^{eff} \end{bmatrix}$$

$\sigma_x^{eff}, \sigma_y^{eff}$ = effective electrical conductivity (defined by Equation 5.86 (p. 192))

$$[C_e] = \text{dielectrical permittivity matrix} = \int_{vol} [B]^T [C][B]\, d(vol)$$

$$[C] = \begin{bmatrix} \varepsilon_x & 0 \\ 0 & \varepsilon_y \end{bmatrix}$$

ε_x, ε_y = dielectric permittivity (input as PERX and PERY on **MP** command)

Although it is assumed that the nodal DOFs are zero at infinity, it is possible to solve thermal problems in which the nodal temperatures tend to some constant value, T_o, rather than zero. In that case, the temperature differential, θ (= T - T_o), may be thought to be posed as the nodal DOF. The actual temperature can then be easily found from T = θ + T_o. For transient analysis, θ must be zero at infinity t > 0, where t is time. Neumann boundary condition is automatically satisfied at infinity.

The $\{B_i\}$ vectors of the [B] matrix in Equation 13.181 (p. 496) contain the derivatives of N_i with respect to the global coordinates which are evaluated according to

$$\{B_i\} = \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial s} \\ \\ \frac{\partial N_i}{\partial t} \end{Bmatrix} \tag{13.182}$$

where:

[J] = Jacobian matrix which defines the geometric mapping

[J] is given by

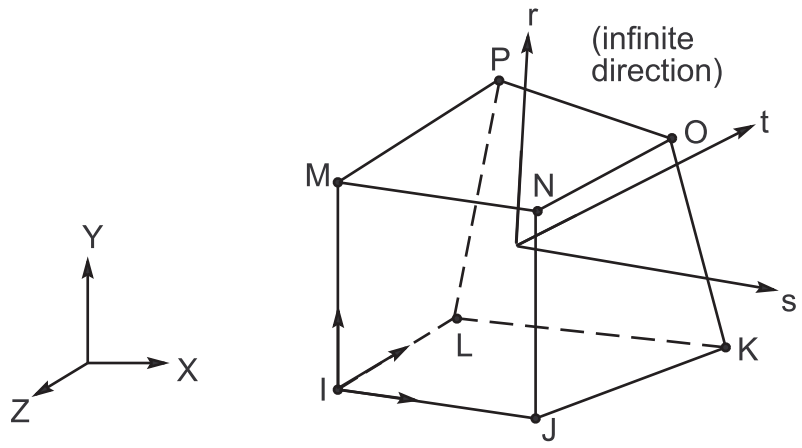
$$[J] = \sum_{i=1}^{4} \begin{bmatrix} \frac{\partial M_i}{\partial s} x_i & \frac{\partial M_i}{\partial s} y_i \\ \\ \frac{\partial M_i}{\partial t} x_i & \frac{\partial M_i}{\partial t} y_i \end{bmatrix} \tag{13.183}$$

The mapping functions [M] in terms of s and t are given in 2-D and Axisymmetric 4-Node Quadrilateral Infinite Solids (p. 343). The domain differential d(vol) must also be written in terms of the local coordinates, so that

$$d(vol) = dx\,dy = |J|\,ds\,dt \tag{13.184}$$

Subject to the evaluation of $\{B_i\}$ and d(vol), which involves the mapping functions, the element matrices $[K_e]$ and $[C_e]$ may now be computed in the standard manner using Gaussian quadrature.

13.111. INFIN111 - 3-D Infinite Solid



Matrix or Vector	Mapping and Shape Functions	Integration Points
Magnetic Scalar Potential Coefficient, Dielectric Permittivity, Electrical Conductivity Coefficient, and Thermal Conductivity Matrices	Equation 11.147 (p. 344), Equation 11.148 (p. 344), Equation 11.149 (p. 344), Equation 11.150 (p. 344), Equation 11.151 (p. 345), and Equation 11.152 (p. 345)	2 x 2 x 2
Specific Heat Matrix	Equation 11.147 (p. 344), Equation 11.150 (p. 344), Equation 11.151 (p. 345), and Equation 11.152 (p. 345)	2 x 2 x 2
Magnetic Vector Potential Coefficient Matrix	Equation 11.144 (p. 343), Equation 11.145 (p. 343), Equation 11.146 (p. 344), Equation 11.150 (p. 344), Equation 11.151 (p. 345), and Equation 11.152 (p. 345)	2 x 2 x 2

13.111.1. Other Applicable Sections

See INFIN110 - 2-D Infinite Solid (p. 494) for the theoretical development of infinite solid elements. The derivation presented in INFIN110 - 2-D Infinite Solid (p. 494) for 2-D can be extended to 3-D in a straightforward manner.

13.112. Reserved for Future Use

This section is reserved for future use.

13.113. Reserved for Future Use

This section is reserved for future use.

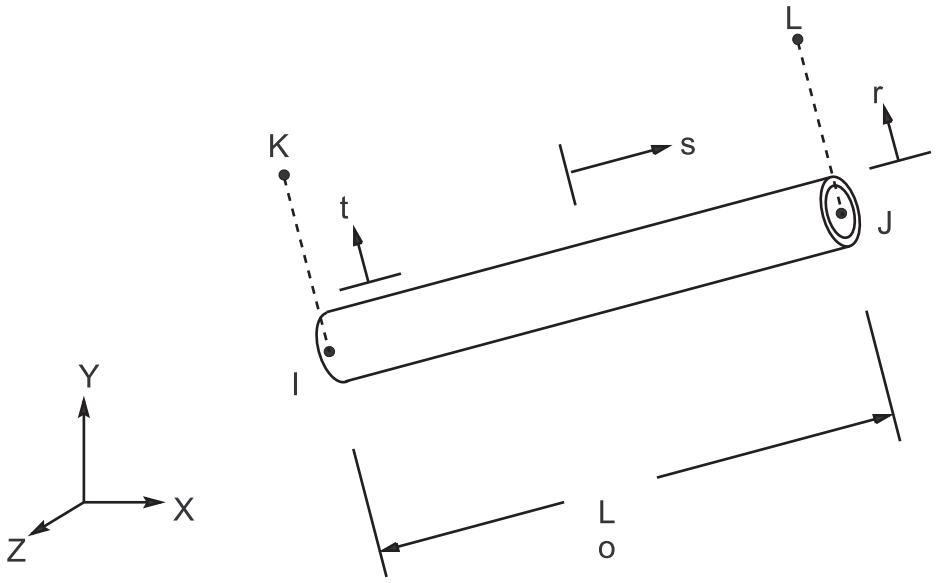
13.114. Reserved for Future Use

This section is reserved for future use.

13.115. Reserved for Future Use

This section is reserved for future use.

13.116. FLUID116 - Coupled Thermal-Fluid Pipe



Matrix or Vector	Geometry	Shape Functions	Integration Points
Thermal Conductivity Matrix	Between nodes I and J	Equation 11.13 (p. 330)	None
	Convection between nodes I and K and between nodes J and L (optional)	None	None
Pressure Conductivity Matrix	Between nodes I and J	Equation 11.12 (p. 330)	None
Specific Heat Matrix and Heat Generation Load Vector	Equation 11.13 (p. 330)		None

13.116.1. Assumptions and Restrictions

Transient pressure and compressibility effects are also not included.

13.116.2. Combined Equations

The thermal and pressure aspects of the problem have been combined into one element having two different types of working variables: temperatures and pressures. The equilibrium equations for one element have the form of:

$$N_c \begin{bmatrix} [C^t] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{T}\} \\ \{0\} \end{Bmatrix} + N_c \begin{bmatrix} [K^t] & [0] \\ [0] & [K^p] \end{bmatrix} \begin{Bmatrix} \{T\} \\ \{P\} \end{Bmatrix} = \begin{Bmatrix} \{Q\} \\ \{w\} \end{Bmatrix} + N_c \begin{Bmatrix} \{Q^g\} \\ \{H\} \end{Bmatrix} \quad (13.185)$$

where:

$[C^t]$ = specific heat matrix for one channel
$\{T\}$ = nodal temperature vector
$\{\dot{T}\}$ = vector of variations of nodal temperature with respect to time
$\{P\}$ = nodal pressure vector
$[K^t]$ = thermal conductivity matrix for one channel (includes effects of convection and mass transport)
$[K^p]$ = pressure conductivity matrix for one channel
$\{Q\}$ = nodal heat flow vector (input as HEAT on **F** command)
$\{w\}$ = nodal fluid flow vector (input as FLOW on **F** command)
$\{Q^g\}$ = internal heat generation vector for one channel
$\{H\}$ = gravity and pumping effects vector for one channel
N_c = number of parallel flow channels (input as N_c on **R** command)

13.116.3. Thermal Matrix Definitions

Specific Heat Matrix

The specific heat matrix is a diagonal matrix with each term being the sum of the corresponding row of a consistent specific heat matrix:

$$[C^t] = A_c \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{13.186}$$

where:

$$A_c = \frac{\rho_u C_p A L_o}{2}$$

$$\rho_u = \text{effective density} = \begin{cases} \rho & \text{if } R_{gas} = 0.0 \\ \text{or} & \\ \dfrac{P}{R_{gas} T_{abs}} \quad \text{(ideal gas law)} & \text{if } R_{gas} \neq 0.0 \end{cases}$$

ρ = mass density (input as DENS on **MP** command
P = pressure (average of first two nodes)
T_{abs} = T + TOFFST = absolute temperature
T = temperature (average of first two nodes)
TOFFST = offset temperature (input on **TOFFST** command)
C_p = specific heat (input as C on **MP** command)
A = flow cross-sectional area (input as A on **R** command)
L_o = length of member (distance between nodes I and J)
R_{gas} = gas constant (input as R_{gas} on **R** command)

Thermal Conductivity Matrix

The thermal conductivity matrix is given by:

$$[K^t] = \begin{bmatrix} B_1 + B_2 - B_4 & -B_1 + B_4 & -B_2 & 0 \\ -B_1 - B_5 & B_1 + B_3 + B_5 & 0 & -B_3 \\ -B_2 & 0 & B_2 & 0 \\ 0 & -B_3 & 0 & B_3 \end{bmatrix} \quad (13.187)$$

where:

$$B_1 = \frac{AK_s}{\ell}$$

K_s = thermal conductivity (input as KXX on **MP** command)

$B_2 = h\ A_I$

h = film coefficient (defined below)

A_I = lateral area of pipe associated with end I (input as $(A_n)_I$ on **R** command)

(defaults to $\frac{\pi DL}{2}$ if KEYOPT(2) = 2, defaults to πDL if KEYOPT(2) = 3)

$B_3 = h\ A_J$

A_J = lateral area of pipe associated with end I (input as $(A_n)_J$ on **R** command)

(defaults to $\frac{\pi DL}{2}$ if KEYOPT(2) = 2, defaults to πDL if KEYOPT(2) = 4)

D = hydraulic diameter (input as D on **R** command)

$$B_4 = \begin{cases} wC_p & \text{if flow is from node J to node I} \\ 0 & \text{if flow is from node I to node J} \end{cases}$$

$$B_5 = \begin{cases} wC_p & \text{if flow is from node I to node J} \\ 0 & \text{if flow is from node J to node I} \end{cases}$$

w = mass fluid flow rate in the element

w may be determined by the program or may be input by the user:

$$W = \begin{cases} \text{computed from previous iteration} & \text{if pressure is a degree of freedom} \\ \text{or} & \\ \text{input (VAL1 on SFE,,,**HFLUX** command)} & \text{if pressure is not a degree of freedom} \end{cases} \quad (13.188)$$

The above definitions of B_4 and B_5, as used by Equation 13.187 (p. 502), cause the energy change due to mass transport to be lumped at the outlet node.

The film coefficient h is defined as:

$$h = \begin{cases} \text{material property input (HF on MP command)} & \text{if KEYOPT(4) = 0} \\ \text{or} & \\ \dfrac{NuK_s}{D} & \text{if KEYOPT(4) = 1} \\ \text{or} & \\ \text{table input (TB, HFLM table)} & \text{if KEYOPT(4) = 2,3, or 4} \\ \text{or} & \\ \text{defined by user programmable feature User116Hf} & \text{if KEYOPT(4) = 5} \end{cases} \tag{13.189}$$

Nu, the Nusselt number, is defined for KEYOPT(4) = 1 as:

$$Nu = N_1 + N_2 Re^{N_3} Pr^{N_4} \tag{13.190}$$

where:

N1 to N4 = input constants (input on **R** commands)

$Re = \dfrac{wD}{\mu A}$ = Reynolds number

μ = viscosity (input as VISC on **MP** command)

$Pr = \dfrac{C_p \mu}{K_s}$ = Prandtl number

A common usage of Equation 13.190 (p. 503) is the Dittus-Boelter correlation for fully developed turbulent flow in smooth tubes (Holman([55] (p. 923))):

$$Nu = 0.023 Re^{0.8} Pr^{a} \tag{13.191}$$

where:

$$a = \begin{cases} 0.4 \text{ for heating} \\ 0.3 \text{ for cooling} \end{cases}$$

Heat Generation Load Vector

The internal heat generation load vector is due to both average heating effects and viscous damping:

$$\{Q^g\} = \begin{Bmatrix} Q_n \\ Q_n \\ 0 \\ 0 \end{Bmatrix} \tag{13.192}$$

where:

$$Q_n = \frac{L_o}{2}(A\ddot{q} + \pi V_{DF} C_{ver} F\mu v^2)$$

$\ddot{q}$ = internal heat generation rate per unit volume (input on **BF** or **BFE** command)

V_{DF} = viscous damping multiplier (input on **RMORE** command)
C_{ver} = units conversion factor (input on **RMORE** command)

$$F = \text{flow type factor} = \begin{cases} 8.0 & \text{if } Re \le 2500.0 \\ 0.21420 & \text{if } Re > 2500.0 \end{cases}$$

v = average velocity

The expression for the viscous damping part of Q_n is based on fully developed laminar flow.

13.116.4. Fluid Equations

Bernoulli's equation is:

$$Z_I + \frac{P_I}{\gamma} + \frac{v_I^2}{2g} + \frac{P_{PMP}}{\gamma} = Z_J + \frac{P_J}{\gamma} + \frac{v_J^2}{2g} + C_L \frac{v_a^2}{2g} \tag{13.193}$$

where:

Z = coordinate in the negative acceleration direction
P = pressure
$\gamma = \rho g$
g = acceleration of gravity
v = velocity
P_{PMP} = pump pressure (input as P_p on **R** command)
C_L = loss coefficient

The loss coefficient is defined as:

$$C_L = \frac{f\ell}{D} + \beta\ell \tag{13.194}$$

where:

$$\beta = \text{extra flow loss factor} = \begin{cases} \frac{f\ell_a}{D\ell} & \text{if KEYOPT(8) = 0} \\ \text{or} & \\ \frac{k}{\ell} & \text{if KEYOPT(8) = 1} \end{cases}$$

ℓ_a = additional length to account for extra flow losses (input as L_a on **R** command)
k = loss coefficient for typical fittings (input as K on **R** command)
f = Moody friction factor, defined below:

For the first iteration of the first load step,

$$f = \begin{cases} f_m & \text{if } f_m \ne 0.0 \\ 1.0 & \text{if } f_m = 0.0 \end{cases} \tag{13.195}$$

where:

f_m = input as MU on **MP** command

For all subsequent iterations

$$f = \begin{cases} f_x & \text{if KEYOPT(7) = 0} \\ f_m & \text{if KEYOPT(7) = 1} \\ \text{table input(defined by TB, FLOW)} & \text{if KEYOPT(7) = 2,3} \end{cases} \tag{13.196}$$

The smooth pipe empirical correlation is:

$$f_x = \begin{cases} \dfrac{64}{Re} & 0 < Re \le 2500 \\ \text{or} & \\ \dfrac{0.316}{(Re)^{1/4}} & 2500 < Re \end{cases} \tag{13.197}$$

Bernoulli's Equation 13.193 (p. 504) may be simplified for this element, since the cross-sectional area of the pipe does not change. Therefore, continuity requires all velocities not to vary along the length. Hence $v_1 = v_2 = v_a$, so that Bernoulli's Equation 13.193 (p. 504) reduces to:

$$Z_I - Z_J + \frac{P_I - P_J}{\gamma} + \frac{P_{PMP}}{\gamma} = C_L \frac{v^2}{2g} \tag{13.198}$$

Writing Equation 13.198 (p. 505) in terms of mass flow rate ($w = \rho A v$), and rearranging terms to match the second half of Equation 13.185 (p. 500),

$$\frac{2\rho A^2}{C_L}(P_I - P_J) = w^2 + \frac{2g\rho^2 A^2}{C_L}\left(-Z_I + Z_J - \frac{P_{PMP}}{\gamma}\right) \tag{13.199}$$

Since the pressure drop (P_I - P_J) is not linearly related to the flow (w), a nonlinear solution will be required. As the w term may not be squared in the solution, the square root of all terms is taken in a heuristic way:

$$A\sqrt{\frac{2\rho}{C_L}}\sqrt{P_I - P_J} = w + A\sqrt{\frac{2\rho}{C_L}}((-Z_I + Z_J)\rho g - P_{PMP}) \tag{13.200}$$

Defining:

$$B_c = A\sqrt{\frac{2\rho}{C_L}} \tag{13.201}$$

and

$$P_L = (-Z_I + Z_J)\rho g - P_{PMP} \tag{13.202}$$

Equation 13.200 (p. 505) reduces to:

$$B_c\sqrt{P_I - P_J} = w + B_c P_L \tag{13.203}$$

Hence, the pressure conductivity matrix is based on the term $\dfrac{B_c}{\sqrt{P_I - P_J}}$ and the pressure (gravity and pumping) load vector is based on the term $B_c\, P_L$.

Two further points:

1. B_c is generalized as:

$$B_c = \begin{cases} A\sqrt{\frac{2\rho}{C_L}} & \text{if KEYOPT(6) = 0} \\ \text{input constant (input as C on R command)} & \text{if KEYOPT(6) = 1} \\ \text{table input (defined by TB,FCON)} & \text{if KEYOPT(6) = 2 or 3} \\ \text{defined by user programmable feature, User116Cond} & \text{if KEYOPT(6) = 4} \end{cases} \tag{13.204}$$

1. $(-Z_I + Z_J)g$ is generalized as:

$$(-Z_I + Z_J)g = \{\Delta x\}^T \{a_t\} \tag{13.205}$$

where:

$\{\Delta x\}$ = vector from node I to node J
$\{a_t\}$ = translational acceleration vector which includes effects of angular velocities (see Acceleration Effect (p. 665))

13.117. Reserved for Future Use

This section is reserved for future use.

13.118. Reserved for Future Use

This section is reserved for future use.

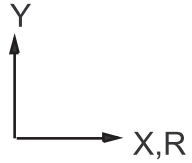
13.119. Reserved for Future Use

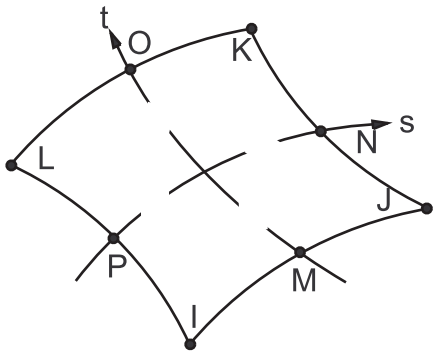
This section is reserved for future use.

13.120. Reserved for Future Use

This section is reserved for future use.

13.121. PLANE121 - 2-D 8-Node Electrostatic Solid





Matrix or Vector	Geometry	Shape Functions	Integration Points
Dielectric Permittivity and Electrical Conductivity Coefficient Matrices, Charge Density Load Vector	Quad	Equation 11.139 (p. 342)	3 x 3
	Triangle	Equation 11.118 (p. 341)	3
Surface Charge Density and Load Vector	Same as coefficient matrix, specialized to the face		2

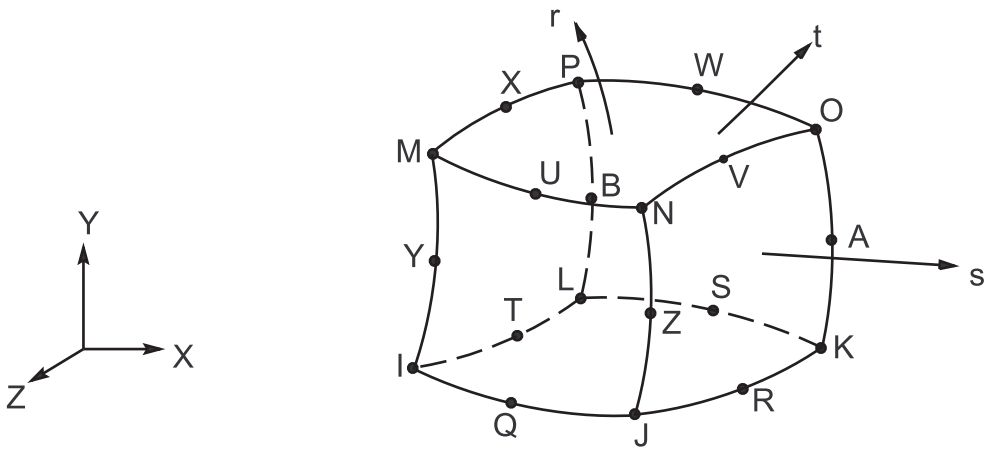
13.121.1. Other Applicable Sections

Electromagnetics (p. 177) describes the derivation of the electrostatic element matrices and load vectors as well as electric field evaluations.

13.121.2. Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the potential varies linearly along that edge.

13.122. SOLID122 - 3-D 20-Node Electrostatic Solid

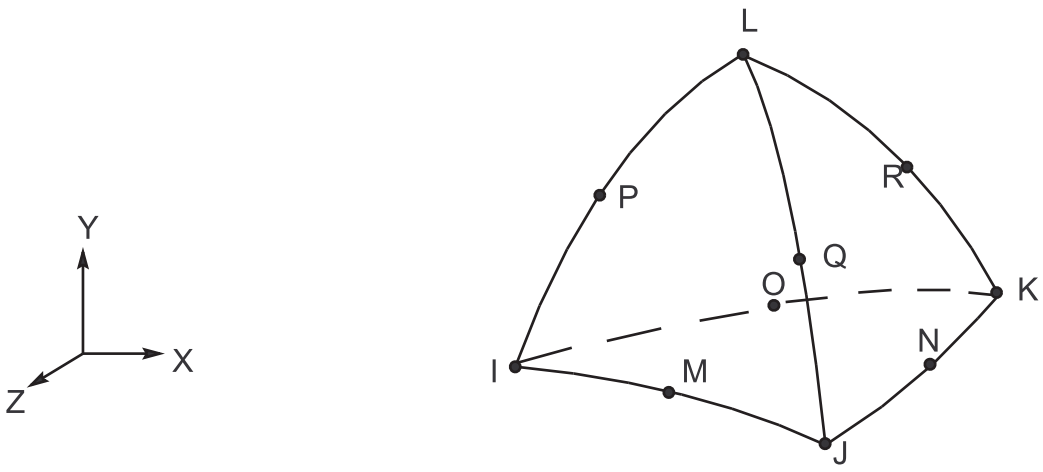


Matrix or Vector	Geometry	Shape Functions	Integration Points
Dielectric Permittivity and Electrical Conductivity Coefficient Matrices, Charge Density Load Vector	Brick	Equation 11.234 (p. 355)	14
	Wedge	Equation 11.210 (p. 352)	3 x 3
	Pyramid	Equation 11.194 (p. 350)	8
	Tet	Equation 11.186 (p. 348)	4
Surface Charge Density Load Vector	Quad	Equation 11.92 (p. 338)	3 x 3
	Triangle	Equation 11.64 (p. 337)	6

13.122.1. Other Applicable Sections

Electromagnetics (p. 177) describes the derivation of electrostatic element matrices and load vectors as well as electric field evaluations.

13.123. SOLID123 - 3-D 10-Node Tetrahedral Electrostatic Solid

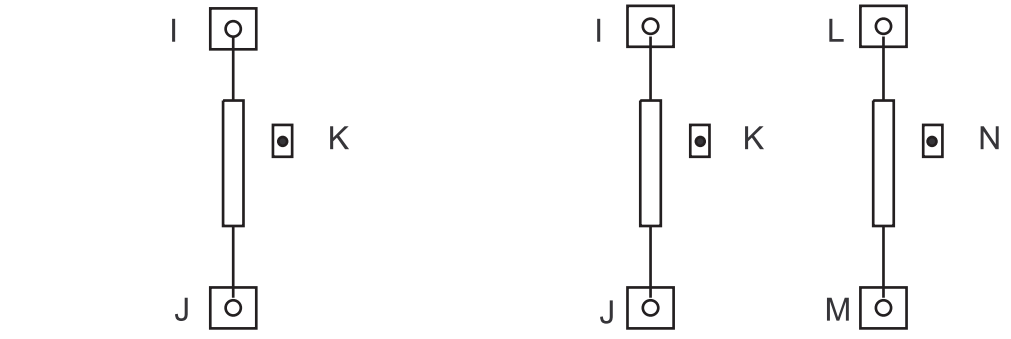


Matrix or Vector	Shape Functions	Integration Points
Dielectric Permittivity and Electrical Conductivity Coefficient Matrices, Charge Density Load Vector	Equation 11.186 (p. 348)	4
Charge Density Surface Load Vector	Equation 11.186 (p. 348) specialized to the face	6

13.123.1. Other Applicable Sections

Electromagnetics (p. 177) describes the derivation of electrostatic element matrices and load vectors as well as electric field evaluations.

13.124. CIRCU124 - Electric Circuit



(a) Independent Circuit Elements (b) Dependent Circuit Elements

Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (lumped)	None

Matrix or Vector	Shape Functions	Integration Points
Damping Matrix	None (lumped, harmonic analysis only)	None
Load Vector	None (lumped)	None

13.124.1. Electric Circuit Elements

CIRCU124 contains 13 linear electric circuit element options. These may be classified into two groups:

1. Independent Circuit Element options, defined by 2 or 3 nodes:

   Resistor (KEYOPT(1) = 0)

   Inductor (KEYOPT(1) = 1)

   Capacitor (KEYOPT(1) = 2)

   Current Source (KEYOPT(1) = 3)

   Voltage Source (KEYOPT(1) = 4)

2. Dependent Circuit Element options, defined by 3, 4, 5, or 6 nodes:

   Stranded coil current source (KEYOPT(1) = 5)

   2-D massive conductor voltage source (KEYOPT(1) = 6)

   3-D massive conductor voltage source (KEYOPT(1) = 7)

   Mutual inductor (KEYOPT(1) = 8)

   Voltage-controlled current source (KEYOPT(1) = 9)

   Voltage-controlled voltage source (KEYOPT(1) = 10)

   Current-controlled voltage source (KEYOPT(1) = 11)

   Current-controlled current source (KEYOPT(1) = 12)

13.124.2. Electric Circuit Element Matrices

All circuit options in CIRCU124 are based on Kirchhoff's Current Law. These options use stiffness matrices based on a simple lumped circuit model.

For transient analysis, an inductor with nodes I and J can be presented by:

$$\frac{\theta\Delta t}{L}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}\begin{Bmatrix} V_I^{n+1} \\ V_J^{n+1} \end{Bmatrix} = \begin{Bmatrix} I_L^{n+1} \\ I_L^{n+1} \end{Bmatrix} \tag{13.206}$$

where:

L = inductance
V_I = voltage at node I
V_J = voltage at node J

Δt = time increment
θ = time integration parameter
n = time step n

$$I_L^{n+1} = \frac{(1-\theta)\Delta t}{L}(V_I^n - V_J^n) + i_L^n$$

$$i_L^{n+1} = \frac{\theta \Delta t}{L}(V_I^{n+1} - V_J^{n+1}) + I_L^{n+1}$$

A capacitor with nodes I and J is represented by:

$$\frac{C}{\theta \Delta t}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} V_I^{n+1} \\ V_J^{n+1} \end{Bmatrix} = \begin{Bmatrix} -I_c^{n+1} \\ I_c^{n+1} \end{Bmatrix} \tag{13.207}$$

where:

C = capacitance

$$I_c^{n+1} = -\frac{c}{\theta \Delta t}(V_I^n - V_J^n) - \frac{1-\theta}{\theta} i_c^n$$

$$i_c^{n+1} = \frac{c}{\theta \Delta t}(V_I^{n+1} V_J^{n+1}) + I_c^{n+1}$$

Similarly, a mutual inductor with nodes I, J, K and L has the following matrix:

$$\frac{\theta \Delta t}{L_1 L_2 - M^2}\begin{bmatrix} L_2 & -L_2 & -M & M \\ -L_2 & L_2 & M & -M \\ -M & M & L_1 & -L_1 \\ M & -M & -L_1 & L_1 \end{bmatrix}\begin{Bmatrix} V_I \\ V_J \\ V_K \\ V_L \end{Bmatrix} = \begin{Bmatrix} -I_1^{n+1} \\ I_1^{n+1} \\ -I_2^{n+1} \\ I_2^{n+1} \end{Bmatrix} \tag{13.208}$$

where:

L1 = input side inductance
L2 = output side inductance
M = mutual inductance

$$I_1^{n+1} = \frac{(1-\theta)\Delta t}{L_1 L_2 - M^2}[L_2(V_I^n - V_J^n) - M(V_K^n - V_L^n)] + i_1^n$$

$$I_2^{n+1} = \frac{(1-\theta)\Delta t}{L_1 L_2 - M^2}[-M(V_I^n - V_J^n) + L_1(V_K^n - V_L^n)] + i_2^n$$

$$i_1^{n+1} = \frac{\theta \Delta t}{L_1 L_2 - M^2}[L_2(V_I^{n+1} - V_J^{n+1}) - M(V_K^{n+1} - V_L^{n+1})] + I_1^{n+1}$$

$$i_2^{n+1} = \frac{\theta \Delta t}{L_1 L_2 - M^2}[-M(V_I^{n+1} - V_J^{n+1}) + L_1(V_K^{n+1} - V_L^{n+1})] + I_1^{n+1}$$

For harmonic analysis, the above three circuit element options have only a damping matrix. For an inductor:

$$\left(-\frac{1}{\omega^2 L}\right)\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix} \tag{13.209}$$

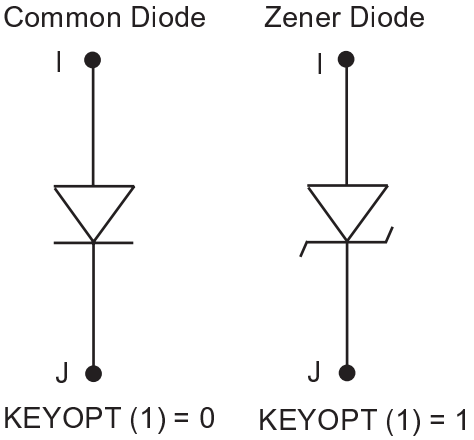
for a capacitor:

$$j\omega C\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix} \tag{13.210}$$

and for a mutual inductor:

$$\left(-\frac{1}{\omega^2(L_1L_2-M^2)}\right)\begin{bmatrix}L_2 & -L_2 & -M & M\\ -L_2 & L_2 & M & -M\\ -M & M & L_1 & -L_1\\ M & -M & -L_1 & L_1\end{bmatrix} \tag{13.211}$$

13.125. CIRCU125 - Diode



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (lumped)	None
Damping Matrix	None	None
Load Vector	None (lumped)	None

13.125.1. Diode Elements

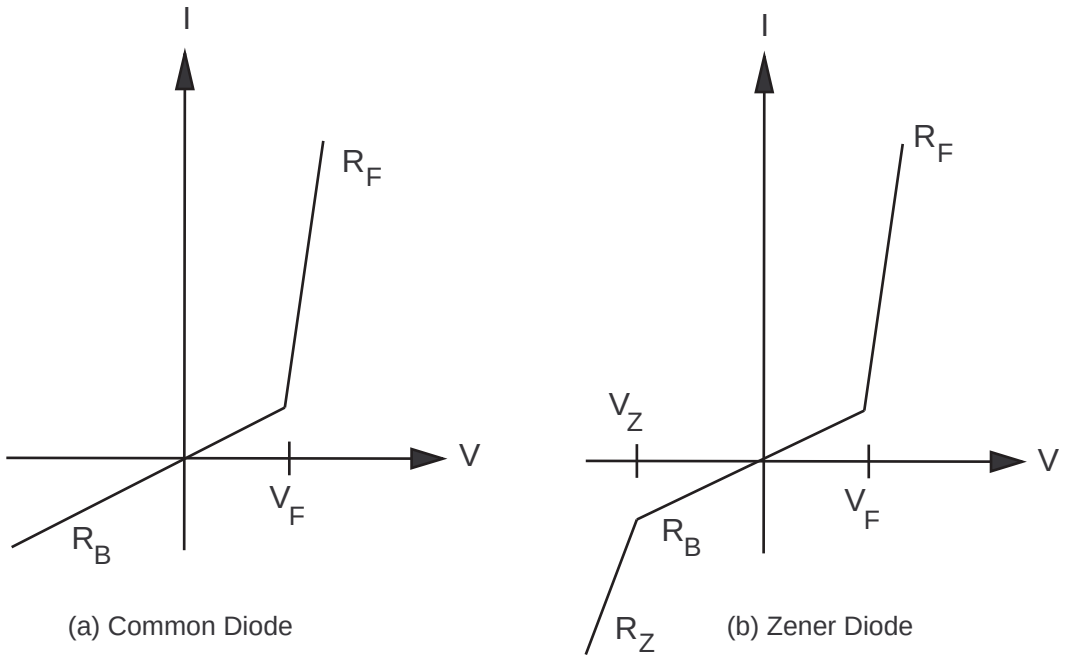
CIRCU125 has two highly nonlinear electric circuit element options:

- Common Diode (KEYOPT(1) = 0)
- Zener Diode (KEYOPT(1) = 1)

The I-V characteristics of common and Zener Diodes are plotted in Figure 13.22: I-V (Current-Voltage) Characteristics of CIRCU125 (p. 512).

As can be seen, the characteristics of the diodes are approximated by a piece-wise linear curve. The common diode has two sections corresponding to open and close states. The Zener diode has three sections corresponding to open, block, and Zener states. The parameters of the piece-wise linear curves are described by real constants depending on KEYOPT(1) selection.

Figure 13.22: I-V (Current-Voltage) Characteristics of CIRCU125



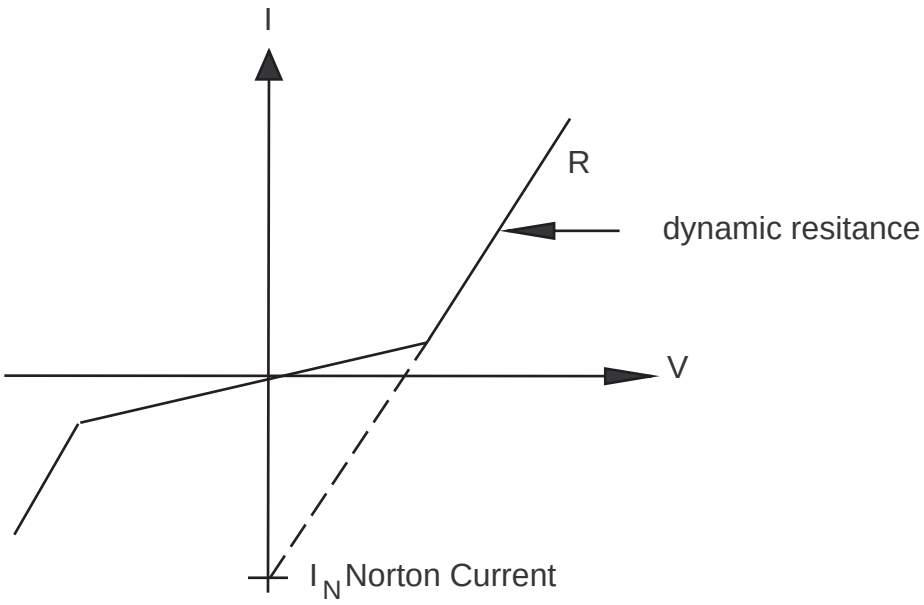
Legend:
V_F = Forward voltage
V_Z = Zener voltage
R_F = Slope of forward resistance
R_B = Slope of blocking resistance
R_Z = Slope of Zener resistance

13.125.2. Norton Equivalents

The behavior of a diode in a given state is described by the Norton equivalent circuit representation (see Figure 13.23: Norton Current Definition (p. 513)).

The Norton equivalent conductance, G, is the derivative (steepness) of the I-V curve to a pertinent diode state. The Norton equivalent current generator, I, is the current where the extension of the linear section of the I-V curve intersects the I-axis.

Figure 13.23: Norton Current Definition



13.125.3. Element Matrix and Load Vector

The element matrix and load vectors are obtained by using the nodal potential formulation, a circuit analysis technique which suits perfectly for coupling lumped circuit elements to distributed finite element models.

The stiffness matrix is:

$$K = G\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.212}$$

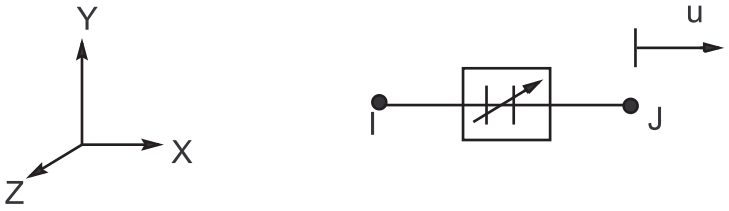
The load vector is:

$$F = I\begin{Bmatrix} 1 \\ -1 \end{Bmatrix} \tag{13.213}$$

where:

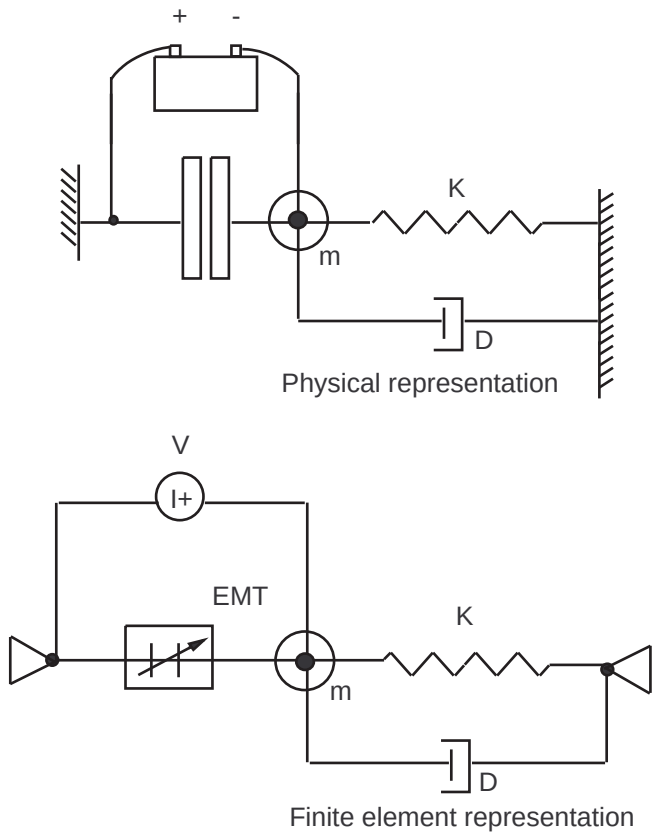
G and I = Norton equivalents of the diode in the pertinent state of operation.

13.126. TRANS126 - Electromechanical Transducer



The line electromechanical transducer element, TRANS126, realizes strong coupling between distributed and lumped mechanical and electrostatic systems. For details about its theory see Gyimesi and Ostergaard([248] (p. 934)). This element is especially suitable for the analysis of Micro Electromechanical Systems (MEMS): accelerometers, pressure sensors, micro actuators, gyroscopes, torsional actuators, filters, HF switches, etc.

Figure 13.24: Electromechanical Transducer



See, for example, Figure 13.24: Electromechanical Transducer (p. 514) with a damped spring mass resonator driven by a parallel plate capacitor fed by a voltage generator constituting an electromechanical system. The left side shows the physical layout of the transducer connected to the mechanical system, the right side shows the equivalent electromechanical transducer element connected to the mechanical system.

TRANS126 is a 2 node element each node having a structural (UX, UY or UZ) and an electrical (VOLT) DOFs. The force between the plates is attractive:

$$F = \frac{1}{2}\frac{dC}{dx}V^2 \tag{13.214}$$

where:

F = force
C = capacitance
x = gap size
V = voltage between capacitor electrodes

The capacitance can be obtained by using the **CMATRIX** macro for which the theory is given in Capacitance Computation (p. 220).

The current is

$$I = C\frac{dV}{dt} + \frac{dC}{dx}vV \tag{13.215}$$

where:

I = current
t = time
$v = \text{velocity of gap opening}\left(=\frac{dx}{dt}\right)$

The first term is the usual capacitive current due to voltage change; the second term is the motion induced current.

For small signal analysis:

$$F = F_0 + D_{xv}v + D_{xv}\frac{dV}{dt} + K_{xx}\Delta x + K_{xv}\Delta V \tag{13.216}$$

$$I = I_0 + D_{vx}v + D_{vv}\frac{dV}{dt} + K_{vx}\Delta x + K_{vv}\Delta V \tag{13.217}$$

where:

F_0 = force at the operating point
I_0 = current at the operating point
[D] = linearized damping matrices
[K] = linearized stiffness matrices
Δx = gap change between the operating point and the actual solution
ΔV = voltage change between the operating point and the actual solution

The stiffness and damping matrices characterize the transducer for small signal prestressed harmonic, modal and transient analyses.

For large signal static and transient analyses, the Newton-Raphson algorithm is applied with F_0 and I_0 constituting the Newton-Raphson restoring force and [K] and [D] the tangent stiffness and damping matrices.

$$K_{xx} = \frac{dF}{dx} = \frac{1}{2}C''V^2 \tag{13.218}$$

where:

K_{xx} = electrostatic stiffness (output as ESTIF)
F = electrostatic force between capacitor plates
V = voltage between capacitor electrodes
C'' = second derivative of capacitance with respect to gap displacement

$$K_{vv} = \frac{dI}{dV} = C'v \tag{13.219}$$

where:

K_{vv} = motion conductivity (output as CONDUCT)
I = current
C' = first derivative of capacitance with respect to gap displacement
v = velocity of gap opening

Definitions of additional post items for the electromechanical transducer are as follows:

$$P_m = Fv \tag{13.220}$$

where:

P_m = mechanical power (output as MECHPOWER)
F = force between capacitor plates
v = velocity of gap opening

$$P_e = VI \tag{13.221}$$

where:

P_e = electrical power (output as ELECPOWER)
V = voltage between capacitor electrodes
I = current

$$W_c = \frac{1}{2}CV^2 \tag{13.222}$$

where:

W_c = electrostatic energy of capacitor (output as CENERGY)
V = voltage between capacitor electrodes
C = capacitance

$$F = \frac{1}{2}\frac{dC}{dx}V^2 \tag{13.223}$$

where:

F = electrostatic force between capacitor plates (output as EFORCE)
C = capacitance
x = gap size

$\frac{dC}{dx}$ = first derivative of capacitance with regard to gap

V = voltage between capacitor electrodes

$\frac{dV}{dt}$ = voltage rate (output as DVDT)

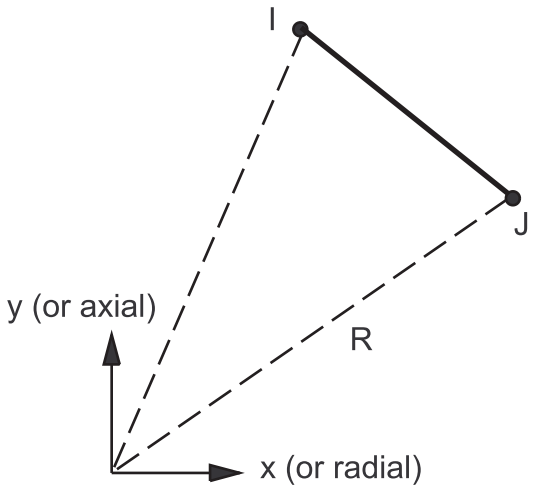
13.127. Reserved for Future Use

This section is reserved for future use.

13.128. Reserved for Future Use

This section is reserved for future use.

13.129. FLUID129 - 2-D Infinite Acoustic

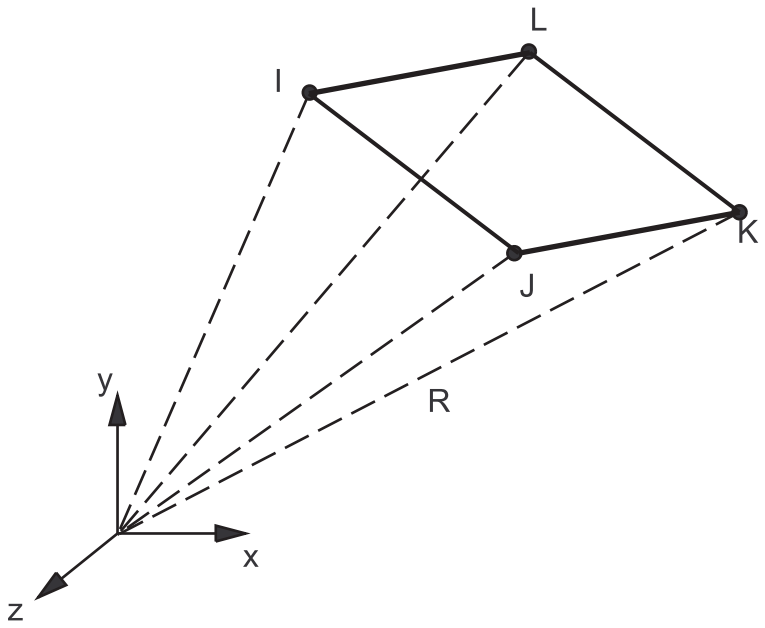


Matrix or Vector	Shape Functions	Integration Points
Fluid Stiffness and Damping Matrices	Equation 11.12 (p. 330)	2

13.129.1. Other Applicable Sections

The mathematical formulation and finite element discretization are presented in FLUID130 - 3-D Infinite Acoustic (p. 518).

13.130. FLUID130 - 3-D Infinite Acoustic



Matrix or Vector	Shape Functions	Integration Points
Fluid Stiffness and Damping Matrices	Equation 11.127 (p. 341)	2 x 2

13.130.1. Mathematical Formulation and F.E. Discretization

The exterior structural acoustics problem typically involves a structure submerged in an infinite, homogeneous, inviscid fluid. The fluid is considered linear, meaning that there is a linear relationship between pressure fluctuations and changes in density. Equation 13.224 (p. 518) is the linearized, lossless wave equation for the propagation of sound in fluids.

$$\nabla^2 P = \frac{1}{c^2}\ddot{P} \text{ in } \Omega^+ \tag{13.224}$$

where:

P = pressure
c = speed of sound in the fluid (input as SONC on **MP** command)
$\ddot{P}$ = second derivative of pressure with respect to time
Ω^+ = unbounded region occupied by the fluid

In addition to Equation 13.224 (p. 518)), the following Sommerfeld radiation condition (which simply states that the waves generated within the fluid are outgoing) needs to be satisfied at infinity:

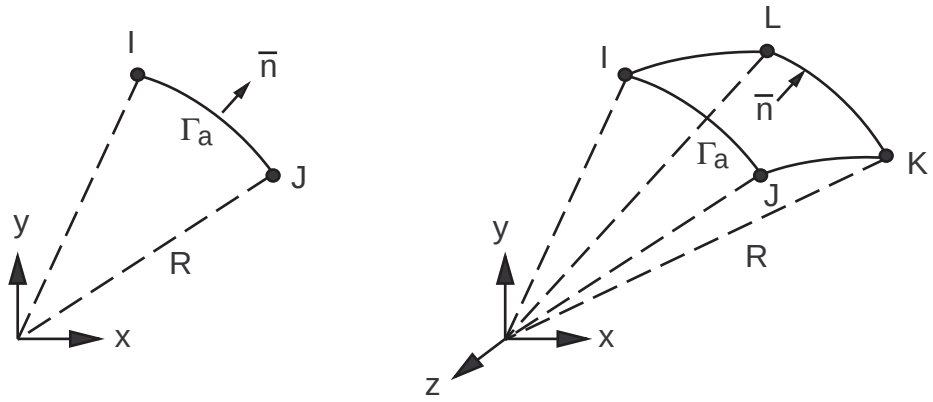
$$\lim_{r\to\infty} r\frac{d-1}{2}\left(P_r + \frac{1}{c}\dot{P}\right) = 0 \tag{13.225}$$

where:

r = distance from the origin
P_r = pressure derivative along the radial direction
d = dimensionality of the problem (i.e., d =3 or d =2 if Ω^+ is 3-D or 2-D respectively

A primary difficulty associated with the use of finite elements for the modeling of the infinite medium stems precisely from the need to satisfy the Sommerfeld radiation condition, Equation 13.225 (p. 518). A typical approach for tackling the difficulty consists of truncating the unbounded domain Ω^+ by the introduction of an absorbing (artificial) boundary Γ_a at some distance from the structure.

Figure 13.25: Absorbing Boundary



The equation of motion Equation 13.224 (p. 518) is then solved in the annular region Ω^f which is bounded by the fluid-structure interface Γ and the absorbing boundary Γ_a. In order, however, for the resulting problem in Ω^f to be well-posed, an appropriate condition needs to be specified on Γ_a. Towards this end, the following second-order conditions are used (Kallivokas et al.([218] (p. 933))) on Γ_a:

In two dimensions:

$$P_n + \gamma P_n = -\frac{1}{c}\ddot{P} + \left(\frac{1}{2}\kappa - \frac{\gamma}{c}\right)\dot{P} + \frac{1}{2}cP_{\lambda\lambda} + \left(\frac{1}{8}\kappa^2 c + \frac{1}{2}\kappa\gamma\right)P \tag{13.226}$$

where:

n = outward normal to Γ_a
P_n = pressure derivative in the normal direction
$P_{\lambda\lambda}$ = pressure derivative along Γ_a
k = curvature of Γ_a
γ = stability parameter

In three dimensions:

$$\dot{P}_n + \gamma P_n = -\frac{1}{c}\ddot{P} + \left(H - \frac{\gamma}{c}\right)\dot{P} + H\gamma P + \frac{c}{2\sqrt{EG}}\left[\left(\sqrt{\frac{G}{E}}P_u\right)_u + \left(\sqrt{\frac{G}{E}}P_v\right)_v\right] + \frac{c}{2}(H^2 - K)P \tag{13.227}$$

where:

n = outward normal
u and v = orthogonal curvilinear surface coordinates (e.g., the meridional and polar angles in spherical coordinates)
P_u, P_v = pressure derivatives in the Γ_a surface directions
H and K = mean and Gaussian curvature, respectively
E and G = usual coefficients of the first fundamental form

13.130.2. Finite Element Discretization

Following a Galerkin based procedure, Equation 13.224 (p. 518) is multiplied by a virtual quantity δP and integrated over the annular domain Ω^f. By using the divergence theorem on the resulting equation it can be shown that:

$$\frac{1}{c^2}\int_{\Omega^f}\delta P\ddot{P}d\Omega^f+\int_{\Omega^f}\nabla\delta P\cdot\nabla P d\Omega^f-\int_{\Gamma_a}\delta P P_n d\Gamma_a=-\int_{\Gamma}\delta P P_n d\Gamma \tag{13.228}$$

Upon discretization of Equation 13.228 (p. 520), the first term on the left hand side will yield the mass matrix of the fluid while the second term will yield the stiffness matrix.

Next, the following finite element approximations for quantities on the absorbing boundary Γ_a placed at a radius R and their virtual counterparts are introduced:

$$P(x,t)=\mathbf{N}_1^T(x)P(t),\qquad q^{(1)}(x,t)=\mathbf{N_2}(x)q^{(1)}(t),\qquad q^{(2)}(x,t)=\mathbf{N}_3^T(x)q^{(2)}(t) \tag{13.229}$$

$$\delta P(x)=\delta P^T\mathbf{N_1}(x),\qquad \delta q^{(1)}(x)=\delta q^{(1)T}\mathbf{N_2}(x),\ \delta q^{(2)}(x)=\delta q^{(2)T}\mathbf{N_3}(x) \tag{13.230}$$

where:

N_1, N_2, N_3 = vectors of shape functions (= {N_1}, {N_2}, {N_3})
P, $q^{(1)}$, $q^{(2)}$ = unknown nodal values (P is output as degree of freedom PRES. $q^{(1)}$ and $q^{(2)}$ are solved for but not output).

Furthermore, the shape functions in Equation 13.229 (p. 520) and Equation 13.230 (p. 520) are set to:

$$\mathbf{N}_1=\mathbf{N}_2=\mathbf{N}_3=\mathbf{N} \tag{13.231}$$

The element stiffness and damping matrices reduce to:

For two dimensional case:

$$[K_a^{2D}]=\frac{1}{8R}\begin{bmatrix} 4\int_{\Gamma_a^e}\mathbf{N}\mathbf{N}^T d\lambda_e & 4R^2\int_{\Gamma_a^e}\mathbf{N}\mathbf{N}^T d\lambda_e & -\int_{\Gamma_a^e}\mathbf{N}\mathbf{N}^T d\lambda_e \\ 4R^2\int_{\Gamma_a^e}\mathbf{N}\mathbf{N}^T d\lambda_e & -4R^2\int_{\Gamma_a^e}\mathbf{N}\mathbf{N}^T d\lambda_e & 0 \\ -\int_{\Gamma_a^e}\mathbf{N}\mathbf{N}^T d\lambda_e & 0 & \int_{\Gamma_a^e}\mathbf{N}\mathbf{N}^T d\lambda_e \end{bmatrix} \tag{13.232}$$

$$[C_a^{2D}] = \frac{1}{8c}\begin{bmatrix} 8\int_{\Gamma_a^e} \mathbf{N}\mathbf{N}^T d\lambda_e & 0 & 0 \\ 0 & 4R^2\int_{\Gamma_a^e} \mathbf{N}\mathbf{N}^T d\lambda_e & 0 \\ 0 & 0 & \int_{\Gamma_a^e} \mathbf{N}\mathbf{N}^T d\lambda_e \end{bmatrix} \tag{13.233}$$

where:

$d\lambda_e$ = arc-length differential

These matrices are 6 x 6 in size, having 2 nodes per element with 3 degrees of freedom per node (P, $q^{(1)}$, $q^{(2)}$).

For three dimensional case:

$$[K_a^{3D}] = \frac{1}{2R}\begin{bmatrix} 2\int_{\Gamma_a^e} \mathbf{N}\mathbf{N}^T dA_e & R^2\int_{\Gamma_a^e} \bar{\nabla}^s\mathbf{N}\cdot\bar{\nabla}^s\mathbf{N}^T dA_e \\ R^2\int_{\Gamma_a^e} \bar{\nabla}^s\mathbf{N}^T\cdot\bar{\nabla}^s\mathbf{N} dA_e & -R^2\int_{\Gamma_a^e} \bar{\nabla}^s\mathbf{N}\cdot\bar{\nabla}^s\mathbf{N}^T dA_e \end{bmatrix} \tag{13.234}$$

$$[C_a^{3D}] = \frac{1}{2C}\begin{bmatrix} 2\int_{\Gamma_a^e} \mathbf{N}\mathbf{N}^T dA_e & 0 \\ 0 & -R^2\int_{\Gamma_a^e} \bar{\nabla}^s\mathbf{N}\cdot\bar{\nabla}^s\mathbf{N}^T dA_e \end{bmatrix} \tag{13.235}$$

where:

dA_e = area differential

These matrices are 8 x 8 in size, having 4 nodes per element with 2 degrees of freedom per node (P, q) (Barry et al.([217] (p. 933))).

For axisymmetric case:

$$[K_a^{2Da}] = \frac{\pi}{R}\begin{bmatrix} 2\int_{\Gamma_a^e} \mathbf{N}\mathbf{N}^T x d\lambda_e & R^2\int_{\Gamma_a^e} \mathbf{N}\mathbf{N}^T x d\lambda_e \\ R^2\int_{\Gamma_a^e} \mathbf{N}^T\mathbf{N} x d\lambda_e & -R^2\int_{\Gamma_a^e} \mathbf{N}\mathbf{N}^T x d\lambda_e \end{bmatrix} \tag{13.236}$$

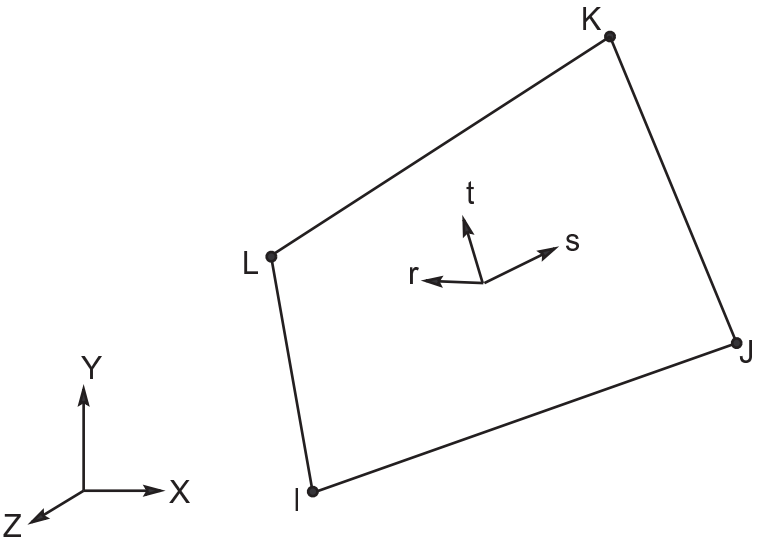
$$[C_a^{2Da}] = \frac{\pi}{C}\begin{bmatrix} 2\int_{\Gamma_a^e} \mathbf{N}\mathbf{N}^T x d\lambda_e & 0 \\ 0 & -R^2\int_{\Gamma_a^e} \mathbf{N}\mathbf{N}^T x d\lambda_e \end{bmatrix} \tag{13.237}$$

where:

x = radius

These matrices are 4 x 4 in size having 2 nodes per element with 2 degrees of freedom per node (P, q).

13.131. SHELL131 - 4-Node Layered Thermal Shell

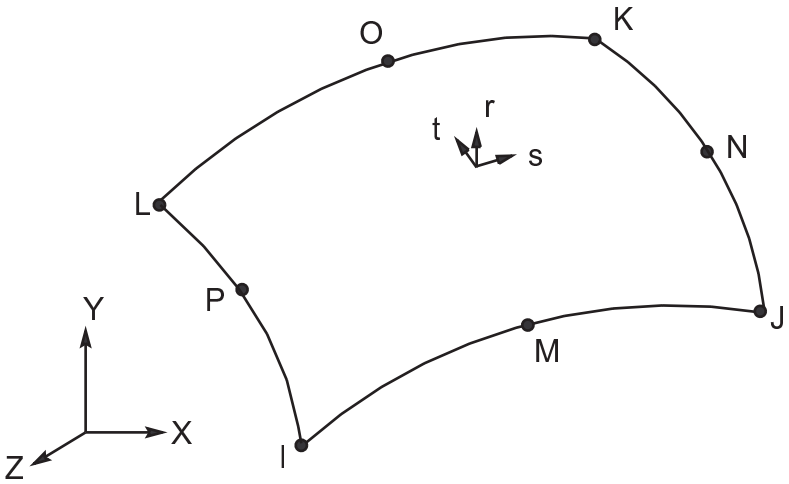


Matrix or Vector	Geometry	Layer Shape Functions	Layer Integration Points
Conductivity Matrix, Heat Generation Load Vector, and Convection Surface Matrix and Load Vector	3 unknowns per node per layer (KEYOPT(3) = 0)	In-Plane: Equation 11.79 (p. 337)	In-Plane: 2 x 2
		Thru Thickness: Equation 11.25 (p. 331)	Thru Thickness: 2
	2 unknowns per node per layer (KEYOPT(3) = 1)	In-Plane: Equation 11.79 (p. 337)	In-Plane: 2 x 2
		Thru Thickness: Equation 11.13 (p. 330)	Thru Thickness: 1
	1 unknown per node per layer (KEYOPT(3) = 2)	In-Plane: Equation 11.79 (p. 337)	In-Plane: 2 x 2
		Thru Thickness: Constant	Thru Thickness: 1
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Lumped Matrices (p. 391)		Same as conductivity matrix

13.131.1. Other Applicable Sections

Heat Flow (p. 227) describes the derivation of the thermal element matrices and load vectors as well as heat flux evaluations.

13.132. SHELL132 - 8-Node Layered Thermal Shell



Matrix or Vector	Geometry	Layer Shape Functions		Layer Integration Points
Conductivity Matrix, Heat Generation Load Vector, Specific Heat Matrix and Convection Surface Matrix and Load Vector	3 unknowns per node per layer (KEYOPT(3) = 0)	In-Plane	Equation 11.91 (p. 338)	Quad: 3 x 3 Triangle: 3
		Thru Thickness	Equation 11.25 (p. 331)	2
	2 unknowns per node per layer (KEYOPT(3) = 1)	In-Plane	Equation 11.91 (p. 338)	Quad: 3 x 3 Triangle: 3
		Thru Thickness	Equation 11.13 (p. 330)	1
	1 unknown per node per layer (KEYOPT(3) = 2)	In-Plane	Equation 11.91 (p. 338)	Quad: 3 x 3 Triangle: 3
		Thru Thickness	Constant	1

13.132.1. Other Applicable Sections

Heat Flow (p. 227) describes the derivation of the thermal element matrices and load vectors as well as heat flux evaluations.

13.133. Reserved for Future Use

This section is reserved for future use.

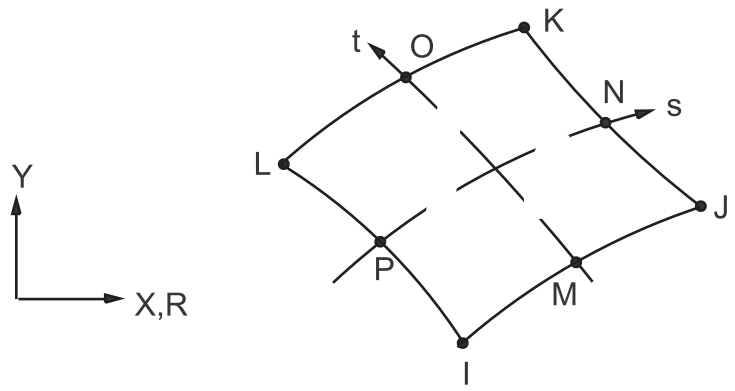
13.134. Reserved for Future Use

This section is reserved for future use.

13.135. Reserved for Future Use

This section is reserved for future use.

13.136. FLUID136 - 3-D Squeeze Film Fluid Element



Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix and Velocity Load Vector	Quad, if KEYOPT(2) = 0	Equation 11.78 (p. 337)	2 x 2 (4-node)
	Quad, if KEYOPT(2) = 1	Equation 11.107 (p. 340)	3 x 3 (8-node)
Damping Matrix	Same as conductivity matrix. If KEYOPT(1) = 1, matrix is diagonalized as described in Lumped Matrices (p. 391)		Same as conductivity matrix

13.136.1. Other Applicable Sections

Squeeze Film Theory (p. 245) describes the governing squeeze film equations used as a basis for forming the element matrices.

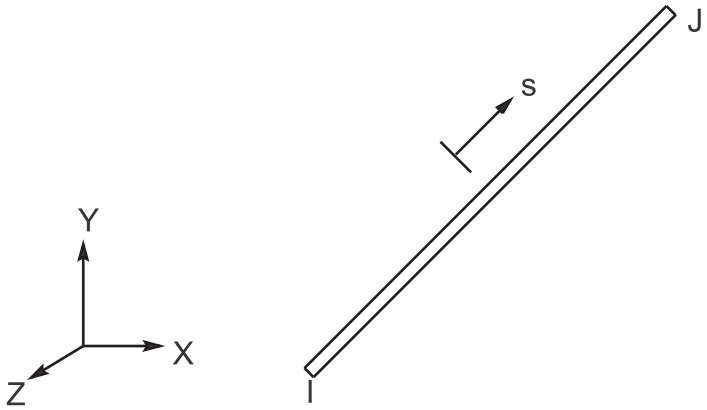
13.136.2. Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the pressure varies linearly along that edge.

13.137. Reserved for Future Use

This section is reserved for future use.

13.138. FLUID138 - 3-D Viscous Fluid Link Element



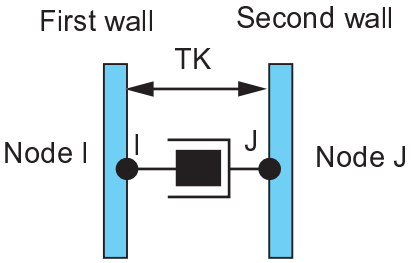
Matrix or Vector	Shape Functions	Integration Points
Pressure and Damping Matrices	Equation 11.12 (p. 330)	None

13.138.1. Other Applicable Sections

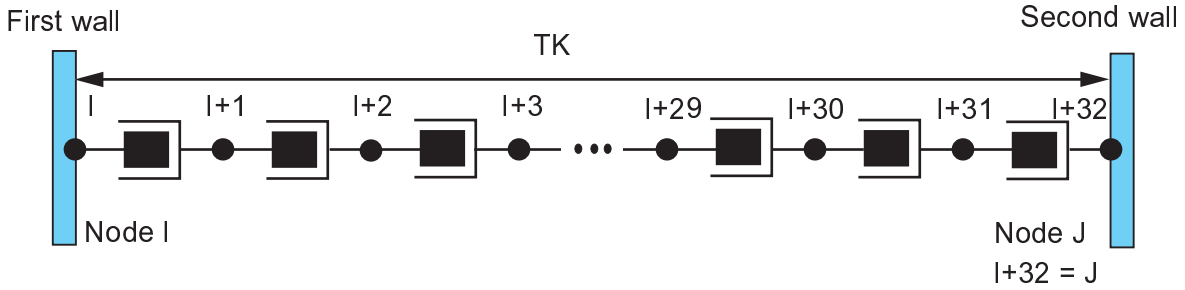
Squeeze Film Theory (p. 245) describes the governing squeeze film equations used as a basis for forming the element matrices.

13.139. FLUID139 - 3-D Slide Film Fluid Element

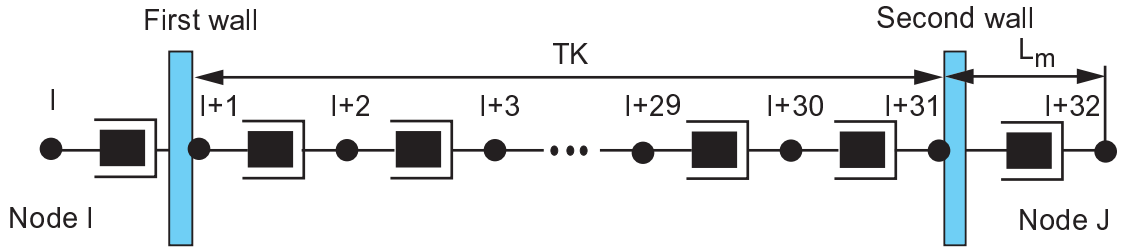
KEYOPT(2)=0 and KEYOPT(3)=0



KEYOPT(2)=1 and KEYOPT(3)=0



KEYOPT(2)=1 and KEYOPT(3)=1



Matrix or Vector	Shape Functions	Integration Points
Fluid, Stiffness, Mass, and Damping Matrices	Analytical Formula	None

13.139.1. Other Applicable Sections

Slide Film Theory (p. 250) describes the governing slide film equations used as a basis for forming the element matrices.

13.140. Reserved for Future Use

This section is reserved for future use.

13.141. Reserved for Future Use

This section is reserved for future use.

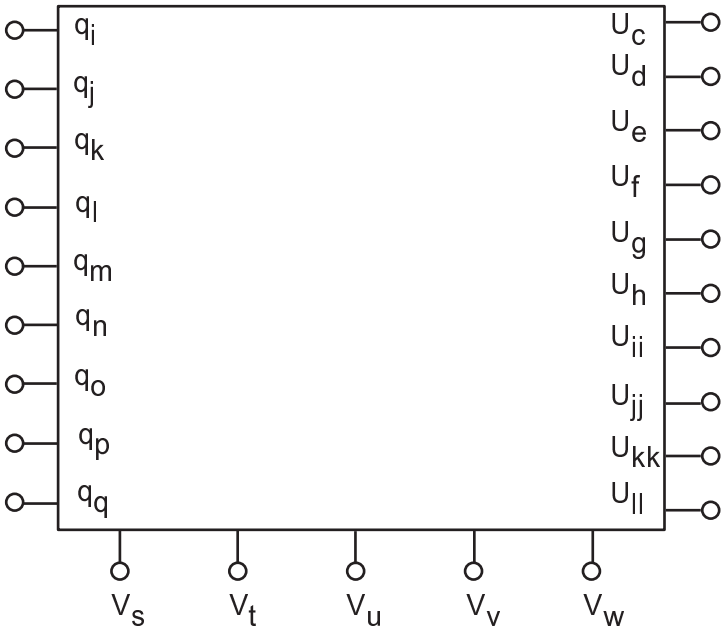
13.142. Reserved for Future Use

This section is reserved for future use.

13.143. Not Documented

No detail or element available at this time.

13.144. ROM144 - Reduced Order Electrostatic-Structural



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (lumped)	None
Damping Matrix	None (lumped)	None
Mass Matrix	None (lumped)	None
Load Vector	None (lumped)	None

ROM144 represents a reduced order model of distributed electostatic-structural systems. The element is derived from a series of uncoupled static FEM analyses using electrostatic and structural elements (Reduced Order Modeling of Coupled Domains (p. 706)). The element fully couples the electrostatic-structural domains and is suitable for simulating the electromechanical response of micro-electromechanical systems (MEMS) such as clamped beams, micromirror actuators, and RF switches.

ROM144 is defined by either 20 (KEYOPT(1) = 0) or 30 nodes (KEYOPT(1) = 1). The first 10 nodes are associated with modal amplitudes, and represented by the EMF DOF labels. Nodes 11 to 20 have electric potential (VOLT) DOFs, of which only the first five are used. The last 10 optional nodes (21 to 30) have structural (UX) DOF to represent master node displacements in the operating direction of the device. For each master node, ROM144 internally uses additional structural DOFs (UY) to account for Lagrange multipliers used to represent internal nodal forces.

13.144.1. Element Matrices and Load Vectors

The FE equations of the 20-node option of ROM144 are derived from the system of governing equations of a coupled electrostatic-structural system in modal coordinates (Equation 14.143 (p. 710) and Equation 14.144 (p. 710))

$$\begin{bmatrix} K^{qq} & K^{qV} \\ K^{Vq} & K^{VV} \end{bmatrix} \begin{bmatrix} q \\ V \end{bmatrix} + \begin{bmatrix} D^{qq} & 0 \\ D^{Vq} & D^{VV} \end{bmatrix} \begin{bmatrix} \dot{q} \\ \dot{V} \end{bmatrix} + \begin{bmatrix} M^{qq} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \ddot{V} \end{bmatrix} = \begin{bmatrix} F \\ I \end{bmatrix} \tag{13.238}$$

where:

K = stiffness matrix
D = damping matrix
M = mass matrix
$q, \dot{q}, \ddot{q}$ = modal amplitude and its first and second derivatives with respect to time
$V, \dot{V}, \ddot{V}$ = electrode voltage and its first and second derivatives with respect to time
F = force
I = electric current

The system of Equation 13.238 (p. 528) is similar to that of the TRANS126 - Electromechanical Transducer (p. 513) element with the difference that the structural DOFs are generalized coordinates (modal amplitudes) and the electrical DOFs are the electrode voltages of the multiple conductors of the electromechanical device.

The contribution to the ROM144 FE matrices and load vectors from the electrostatic domain is calculated based on the electrostatic co-energy W_{el} (Reduced Order Modeling of Coupled Domains (p. 706)).

The electrostatic forces are the first derivative of the co-energy with respect to the modal coordinates:

$$F_k = -\frac{\partial W_{el}}{\partial q_k} \tag{13.239}$$

where:

F_k = electrostatic force
W_{el} = co-energy
q_k = modal coordinate
k = index of modal coordinate

Electrode charges are the first derivatives of the co-energy with respect to the conductor voltage:

$$Q_i = \frac{\partial W_{el}}{\partial V_i} \tag{13.240}$$

where:

Q_i = electrode charge
V_i = conductor voltage
i = index of conductor

The corresponding electrode current I_i is calculated as a time-derivative of the electrode charge Q_i. Both, electrostatic forces and the electrode currents are stored in the Newton-Raphson restoring force vector.

The stiffness matrix terms for the electrostatic domain are computed as follows:

$$K_{kl}^{qq} = \frac{\partial F_k}{\partial q_l} \tag{13.241}$$

$$K_{ki}^{qV} = \frac{\partial F_k}{\partial V_i} \tag{13.242}$$

$$K_{ik}^{Vq} = \frac{\partial I_i}{\partial q_k} \tag{13.243}$$

$$K_{ij}^{VV} = \frac{\partial I_i}{\partial V_j} \tag{13.244}$$

where:

l = index of modal coordinate
j = index of conductor

The damping matrix terms for the electrostatic domain are calculated as follows:

$$D^{qq} = D^{qV} = 0 \tag{13.245}$$

$$D_{ik}^{Vq} = \frac{\partial I_i}{\partial \dot{q}_k} \tag{13.246}$$

$$D_{ij}^{VV} = \frac{\partial I_i}{\partial \dot{V}_j} \tag{13.247}$$

There is no contribution to the mass matrix from the electrostatic domain.

The contribution to the FE matrices and load vectors from the structural domain is calculated based on the strain energy W_{SENE} (Reduced Order Modeling of Coupled Domains (p. 706)). The Newton-Raphson restoring force F, stiffness K, mass M, and damping matrix D are computed according to Equation 13.248 (p. 529) to Equation 13.251 (p. 529).

$$F_i = \frac{\partial W_{SENE}}{\partial q_i} \tag{13.248}$$

$$K_{ij}^{qq} = \frac{\partial^2 W_{SENE}}{\partial q_j \ \partial q_i} \tag{13.249}$$

$$M_{ii} = \frac{1}{\omega_i^2} \ \frac{\partial^2 W_{SENE}}{\partial q_i^2} \tag{13.250}$$

$$D_{ii} = 2 \xi_i \omega_i \ M_{ii} \tag{13.251}$$

where:

i, j = indices of modal coordinates
ω_i = angular frequency of ith eigenmode
ξ_i = modal damping factor (input as *Damp* on the **RMMRANGE** command

13.144.2. Combination of Modal Coordinates and Nodal Displacement at Master Nodes

For the 30-node option of ROM144, it is necessary to establish a self-consistent description of both modal coordinates and nodal displacements at master nodes (defined on the **RMASTER** command defining the generation pass) in order to connect ROM144 to other structural elements UX DOF or to apply nonzero structural displacement constraints or forces.

Modal coordinates q_i describe the amplitude of a global deflection state that affects the entire structure. On the other hand, a nodal displacement u_i is related to a special point of the structure and represents the true local deflection state.

Both modal and nodal descriptions can be transformed into each other. The relationship between modal coordinates q_j and nodal displacements u_i is given by:

$$u_i = \sum_{j=1}^{m} \phi_{ij} q_j \tag{13.252}$$

where:

φ_{ij} = jth eigenmode shape at node i
m = number of eigenmodes considered

Similarly, nodal forces F_i can be transformed into modal forces f_j by:

$$f_j = \sum_{i=1}^{n} \phi_{ij} F_i \tag{13.253}$$

where:

n = number of master nodes

Both the displacement boundary conditions at master nodes u_i and attached elements create internal nodal forces F_i in the operating direction. The latter are additional unknowns in the total equation system, and can be viewed as Lagrange multipliers λ_i mapped to the UY DOF. Hence each master UX DOF requires two equations in the system FE equations in order to obtain a unique solution. This is illustrated on the example of a FE equation (stiffness matrix only) with 3 modal amplitude DOFs (q_1, q_2, q_3), 2 conductors (V_1, V_2), and 2 master UX DOFs (u_1, u_2):

$$\cdots + \begin{bmatrix} K_{11}^{qq} & K_{12}^{qq} & K_{13}^{qq} & K_{11}^{qV} & K_{12}^{qV} & \phi_{11} & \phi_{21} & 0 & 0 \\ K_{21}^{qq} & K_{22}^{qq} & K_{23}^{qq} & K_{21}^{qV} & K_{22}^{qV} & \phi_{12} & \phi_{22} & 0 & 0 \\ K_{31}^{qq} & K_{32}^{qq} & K_{33}^{qq} & K_{13}^{qV} & K_{32}^{qV} & \phi_{13} & \phi_{23} & 0 & 0 \\ K_{11}^{Vq} & K_{12}^{Vq} & K_{13}^{Vq} & K_{11}^{VV} & K_{12}^{VV} & 0 & 0 & 0 & 0 \\ K_{21}^{Vq} & K_{22}^{Vq} & K_{23}^{Vq} & K_{21}^{VV} & K_{22}^{VV} & 0 & 0 & 0 & 0 \\ \phi_{11} & \phi_{12} & \phi_{13} & 0 & 0 & 0 & 0 & -1 & 0 \\ \phi_{21} & \phi_{22} & \phi_{23} & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & K_{11}^{uu} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & K_{22}^{uu} \end{bmatrix} * \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ V_1 \\ V_2 \\ -\lambda_1 \\ -\lambda_2 \\ u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ I_1 \\ I_2 \\ 0 \\ 0 \\ F_1^a \\ F_2^a \end{bmatrix} \begin{matrix} \text{Modal amplitude 1 (EMF)} \\ \text{Modal amplitude 2 (EMF)} \\ \text{Modal amplitude 3 (EMF)} \\ \text{Electrode voltage 1 (VOLT)} \\ \text{Electrode voltage 2 (VOLT)} \\ \text{Lagrange multiplier 1 (UY)} \\ \text{Lagrange multiplier 2 (UY)} \\ \text{Master displacement 1 (UX)} \\ \text{Master displacement 2 (UX)} \end{matrix} \quad (13.254)$$

Rows 6 and 7 of Equation 13.254 (p. 531) correspond to the modal and nodal displacement relationship of Equation 13.252 (p. 530), while column 6 and 7 - to nodal and modal force relationship (Equation 13.253 (p. 530)). Rows and columns (8) and (9) correspond to the force-displacement relationship for the UX DOF at master nodes:

$$K_{ij}u_i = F_i^a - \lambda_i \quad (13.255)$$

$$\lambda_i = F_i \quad (13.256)$$

where K_{ii}^{uu} is set to zero by the ROM144 element. These matrix coefficients represent the stiffness caused by other elements attached to the master node UX DOF of ROM144.

13.144.3. Element Loads

In the generation pass of the ROM tool, the ith mode contribution factors e_i^j for each element load case j (Reduced Order Modeling of Coupled Domains (p. 706)) are calculated and stored in the ROM database file. In the Use Pass, the element loads can be scaled and superimposed in order to define special load situations such as acting gravity, external acceleration or a pressure difference. The corresponding modal forces for the jth load case f_j^E (Equation 14.143 (p. 710)) is:

$$f_j^E = e_i^j K_{ii}^{qq}(0) \quad (13.257)$$

where:

$K_{ii}^{qq}(0)$ = modal stiffness of the ith eigenmode at the inital position ($q_i = 0$ for all modes)

13.145. Reserved for Future Use

This section is reserved for future use.

13.146. Reserved for Future Use

This section is reserved for future use.

13.147. Reserved for Future Use

This section is reserved for future use.

13.148. Reserved for Future Use

This section is reserved for future use.

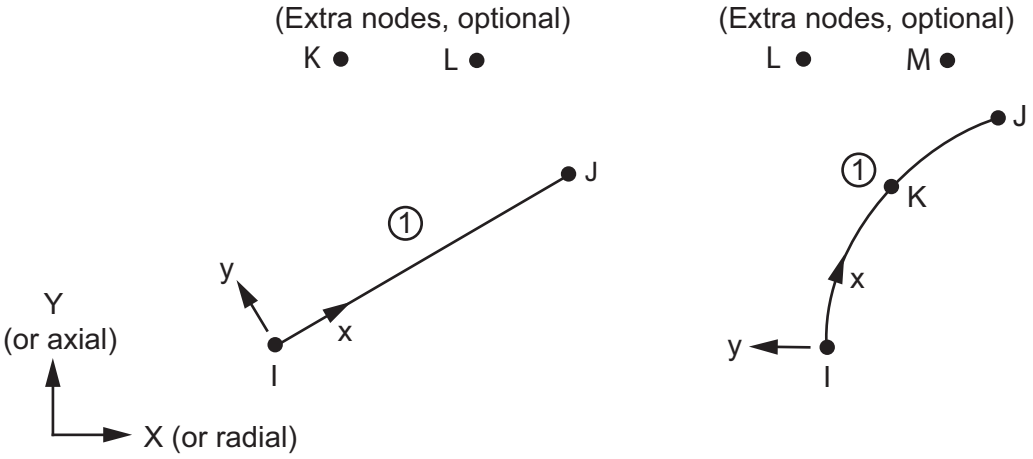
13.149. Reserved for Future Use

This section is reserved for future use.

13.150. Reserved for Future Use

This section is reserved for future use.

13.151. SURF151 - 2-D Thermal Surface Effect



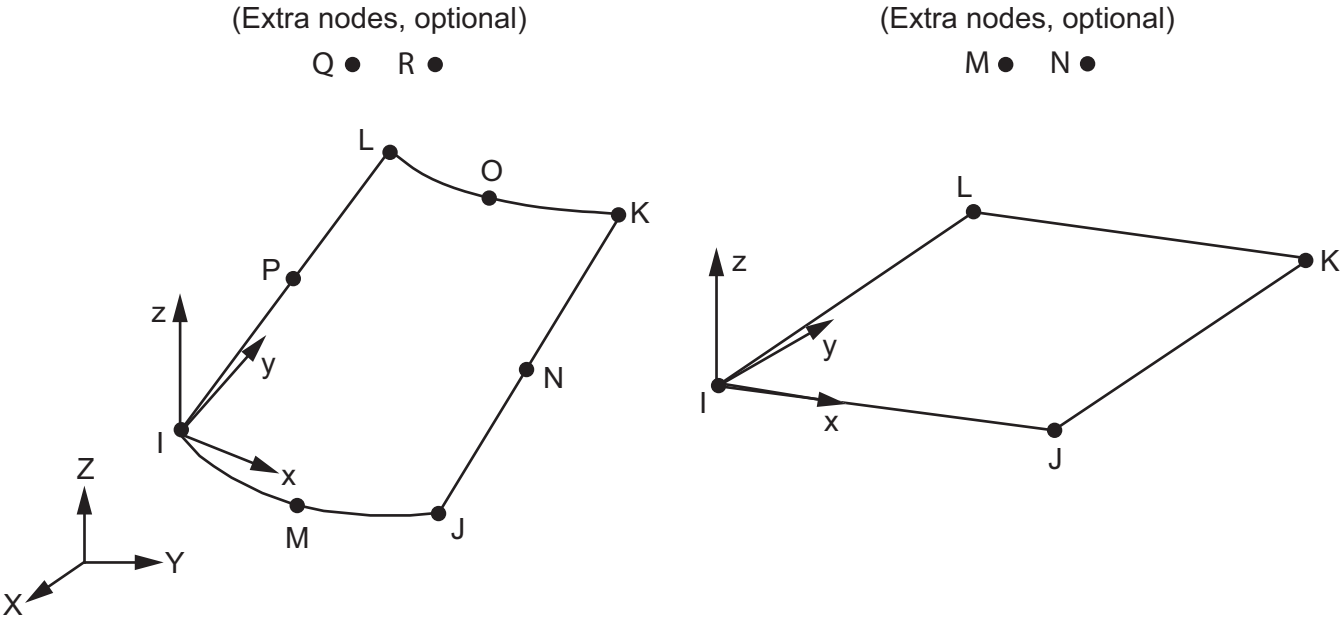
Matrix or Vector	Midside Nodes [1]	Shape Functions	Integration Points
All	With midside nodes	$w = C_1 + C2_x + C_3x^2$	2
	Without midside nodes	$w = C_1 + C_2x$	2

1. Midside node setting is controlled by KEYOPT(4).

Load Type	Distribution
All Loads	Same as shape functions

The logic is very similar to that given for SURF152 - 3-D Thermal Surface Effect (p. 533).

13.152. SURF152 - 3-D Thermal Surface Effect



Matrix or Vector	Geometry / Midside Nodes [1]	Shape Functions	Integration Points
Convection Surface Matrix and Load Vector; and Heat Generation Load Vector	Quad with midside nodes	Equation 11.91 (p. 338)	3 x 3
	Quad without midside nodes	Equation 11.79 (p. 337)	2 x 2
	Triangle with midside nodes	Equation 11.63 (p. 337)	6
	Triangle without midside nodes	Equation 11.106 (p. 340)	3

1. Midside node setting is controlled by KEYOPT(4).

Load Type	Distribution
All Loads	Same as shape functions

13.152.1. Matrices and Load Vectors

When the extra node is not present, the logic is the same as given and as described in Derivation of Heat Flow Matrices (p. 235). The discussion below relates to theory that uses the extra node.

The conductivity matrix is based on one-dimensional flow to and away from the surface. The form is conceptually the same as for LINK33 (Equation 13.39 (p. 432)) except that the surface has four or eight nodes instead of only one node. Using the example of convection and no midside nodes are requested (KEYOPT(4) = 1) (resulting in a 5 x 5 matrix), the first four terms of the main diagonal are:

$$\int_{\text{area}} h_f \{N\}\, d(\text{area}) \tag{13.258}$$

where:

$$h_f = \begin{cases} \text{film coefficient (input on } \textbf{SFE} \text{ command with KVAL=1)} \\ h_u \text{ (If KEYOPT(5) = 1 and user programmable} \\ \text{feature USRSURF116 output argument KEY(1) = 1,} \\ \text{this definition supercedes the other.)} \end{cases}$$

h_u = output argument for film coefficient of USRSURF116
{N} = vector of shape functions

which represents the main diagonal of the upper-left corner of the conductivity matrix. The remaining terms of this corner are all zero. The last main diagonal term is simply the sum of all four terms of Equation 13.258 (p. 533) and the off-diagonal terms in the fifth column and row are the negative of the main diagonal of each row and column, respectively.

If midside nodes are present (KEYOPT(4) = 0) (resulting in a 9 x 9 matrix) Equation 13.258 (p. 533) is replaced by:

$$\int_{\text{area}} h_f \{N\}\{N\}^T \, d(\text{area}) \tag{13.259}$$

which represents the upper-left corner of the conductivity matrix. The last main diagonal is simply the sum of all 64 terms of Equation 13.259 (p. 534) and the off-diagonal terms in the ninth column and row are the negative of the sum of each row and column respectively.

Radiation is handled similarly, except that the approach discussed for LINK31 in LINK31 - Radiation Link (p. 430) is used. A load vector is also generated. The area used is the area of the element. The form factor is discussed in a subsequent section.

An additional load vector is formed when using the extra node by:

$$\{Q^c\} = [K^{tc}]\{T^{ve}\} \tag{13.260}$$

where:

$\{Q^c\}$ = load vector to be formed
$[K^{tc}]$ = element conductivity matrix due to convection

$$\{T^{ve}\} = \begin{bmatrix} 0 & 0 & \cdots & 0 & T_v^G \end{bmatrix}^T$$

$$T_v^G = \begin{cases} \text{output argument TEMVEL if the user} \\ \text{programmable feature USRSURF116} \\ \text{is used.} \\ \\ T_v \text{ if KEYOPT(6) = 1} \\ \text{(see next section)} \\ \\ \text{0.0 for all other cases} \end{cases}$$

TEMVEL from USRSURF116 is the difference between the bulk temperature and the temperature of the extra node.

13.152.2. Adiabatic Wall Temperature as Bulk Temperature

There is special logic that accesses FLUID116 information where FLUID116 has had KEYOPT(2) set equal to 1. This logic uses SURF151 or SURF152 with the extra node present (KEYOPT(5) = 1) and computes an adiabatic wall temperature (KEYOPT(6) = 1). For this case, T_v, as used above, is defined as:

$$T_v = \begin{cases} \dfrac{F_R(V_{rel})^2 - V_{abs}^2}{2g_c J_c C_p^f} & \text{if KEYOPT(1) = 0} \\ \dfrac{F_R(V_{rel})^2 - (\Omega_{ref} F_s R)^2}{2g_c J_c C_p^f} & \text{if KEYOPT(1) = 1} \\ F_R \dfrac{(V_{116})^2}{2g_c J_c C_p^f} & \text{if KEYOPT(1) = 2} \end{cases} \tag{13.261}$$

where:

F_R = recovery factor (see Equation 13.262 (p. 535))

$$V_{rel} = \begin{cases} V_{abs} - \Omega R & \text{if KEYOPT(1) = 0} \\ \Omega_{ref} F_s R - \Omega R & \text{if KEYOPT(1) = 1} \end{cases}$$

V_{abs} = absolute value of fluid velocity (input as VABS on **R** command)

Ω = angular velocity of moving wall (input as OMEGA on **R** command)

$$R = \text{distance of element centroid from } \begin{cases} \text{global Y axis} & \text{for SURF151} \\ \text{global axis selected with KEYOPT(3)} & \text{for SURF152} \end{cases}$$

Ω_{ref} = reference angular velocity (input as $(A_n)_I$ and $(A_n)_J$ on **R** command of FLUID116)

F_s = slip factor (input as SLIPFAI, SLIPFAJ on **R** command of FLUID116)

V_{116} = velocity of fluid at extra node from FLUID116

g_c = gravitational constant used for units consistency (input as GC on **R** command)

J_c = Joule constant used to convert work units to heat units (input as JC on **R** command)

C_p^f = specific heat of fluid (from FLUID116)

The recovery factor is computed as follows:

$$F_R = \begin{cases} C_n & \text{if KEYOPT(2) = 0} \\ Pr^{Cn} & \text{if KEYOPT(2) = 1} \\ Pr^n & \text{if KEYOPT(2) = 2} \end{cases} \tag{13.262}$$

where:

C_n = constant used for recovery factor calculation (input as NRF on **R** command)

$$Pr = \frac{C_p^f \mu^f}{K_x^f} = \text{Prandtl number}$$

$$n = \begin{cases} 0.5000 & \text{if Re < 2500.0} \\ 0.3333 & \text{if Re > 2500.0} \end{cases}$$

μ^f = viscosity of fluid (from FLUID116)

K_x^f = conductivity of fluid (from FLUID116)

$Re = \frac{\rho^f V D}{\mu^f}$ = Reynold's number

ρ^f = density of fluid (from FLUID116)

D = diameter of fluid pipe (from FLUID116)

$$V = \begin{cases} |V_{Rel}| & \text{if KEYOPT(1) = 0,1} \\ |V_{116}| & \text{if KEYOPT(1) = 2} \end{cases} \quad (13.263)$$

where:

V = velocity used to compute Reynold's number

The adiabatic wall temperature is reported as:

$$T_{aw} = T_{ex} + T_v \quad (13.264)$$

where:

T_{aw} = adiabatic wall temperature

T_{ex} = temperature of extra node

KEYOPT(1) = 0 or 1 is ordinarily used for turbomachinry analysis, whereas KEYOPT(1) = 2 is ordinarily used for flow past stationary objects. For turbomachinery analyses T_{ex} is assumed to be the total temperature, but for flow past stationary objects T_{ex} is assumed to be the static temperature.

13.152.3. Film Coefficient Adjustment

After the first coefficient has been determined, it is adjusted if KEYOPT(7) = 1:

$$h_f' = h_f(|T_S - T_B|)^n \quad (13.265)$$

where:

h_f' = adjusted film coefficient

h_f = unadjusted film coefficient

T_S = surface temperature

T_B = bulk temperature (T_{aw}, if defined)

n = real constant (input as ENN on **RMORE** command)

13.152.4. Radiation Form Factor Calculation

The form factor is computed as:

$$F = \begin{cases} \text{input (FORMF on R command)} & \text{if KEYOPT(9) = 1} \\ B & \text{if KEYOPT(9) = 2 or 3} \end{cases} \quad (13.266)$$

also,

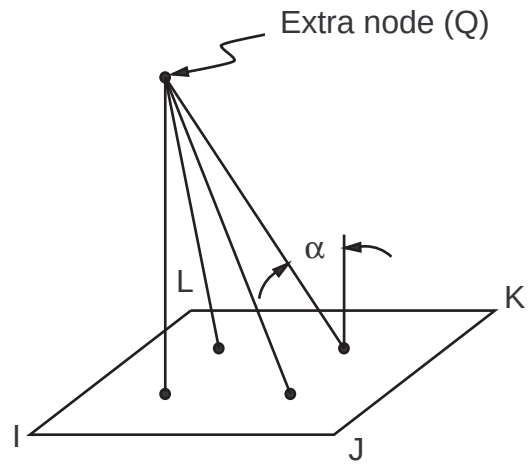
F = form factor (output as FORM FACTOR)

Developing B further

$$B = \begin{cases} \cos\alpha & \text{if } \alpha \le 90^\circ \\ -\cos\alpha & \text{if } \alpha > 90^\circ \text{ and KEYOPT(9) = 2} \\ 0 & \text{if } \alpha > 90^\circ \text{ and KEYOPT(9) = 3} \end{cases}$$

α = angle between element z axis at integration point being processed and the line connecting the integration point and the extra node (see Figure 13.26: Form Factor Calculation (p. 537))

Figure 13.26: Form Factor Calculation



F is then used in the two-surface radiation equation:

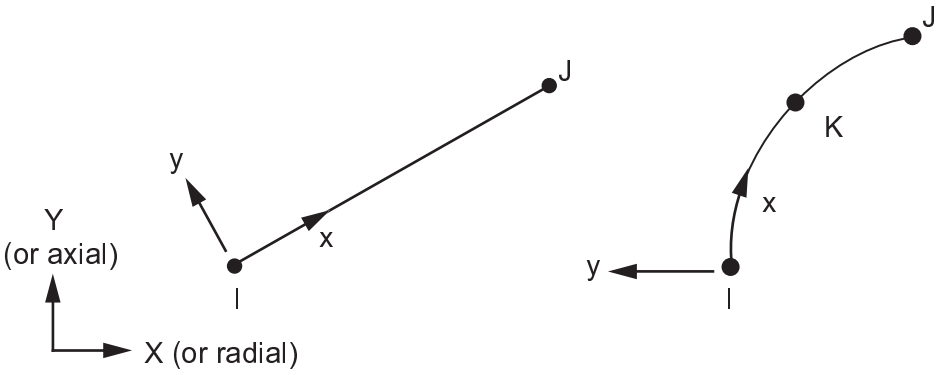
$$Q_e^r = \sigma\varepsilon AF(T^4 - T_Q^4) \tag{13.267}$$

where:

σ = Stefan-Boltzmann constant (input as SBCONST on **R** command)
ε = emissivity (input as EMIS on **MP** command)
A = element area

Note that this "form factor" does not have any distance affects. Thus, if distances are to be included, they must all be similar in size, as in an object on or near the earth being warmed by the sun. For this case, distance affects can be included by an adjusted value of σ.

13.153. SURF153 - 2-D Structural Surface Effect



Matrix or Vector	Midside Nodes [1]	Shape Functions	Integration Points
All	With midside nodes	$w = C_1 + C_2x + C_3x^2$	3
All	Without midside nodes	$w = C_1 + C_2x$	2

1. Midside node setting is controlled by KEYOPT(4).

Load Type	Distribution
All Loads	Same as shape functions

The logic is very similar to that given for SURF154 in SURF154 - 3-D Structural Surface Effect (p. 539) with the differences noted below:

1. For surface tension (input as SURT on **R** command)) on axisymmetric models (KEYOPT(3) = 1), an average force is used on both end nodes.

2. For surface tension with midside nodes, no load is applied at the middle node, and only the component directed towards the other end node is used.

3. When using large deflections, the area on which pressure is applied changes. The updated distance between the two end nodes is used. For plain strain problems, the thickness (distance normal to the X-Y plane) remains at 1.0, by definition. For plane stress problems, the thickness is adjusted:

$$t_u = t(1-\varepsilon_z) \tag{13.268}$$

where:

t_u = final thickness used.

$$t = \begin{cases} 1.0 & \text{if KEYOPT(3) = 0} \\ t_i & \text{if KEYOPT(3) = 3} \end{cases}$$

t_i = thickness for user input option (input as TKPS on **R** command)

ε_z = strain in thickness direction (normal to X-Y plane)

Using the assumption of constant volume:

$$\varepsilon_x + \varepsilon_y + \varepsilon_z = 0 \tag{13.269}$$

where:

ε_x = strain along the length of the element
ε_y = strain normal to the underlying solid.

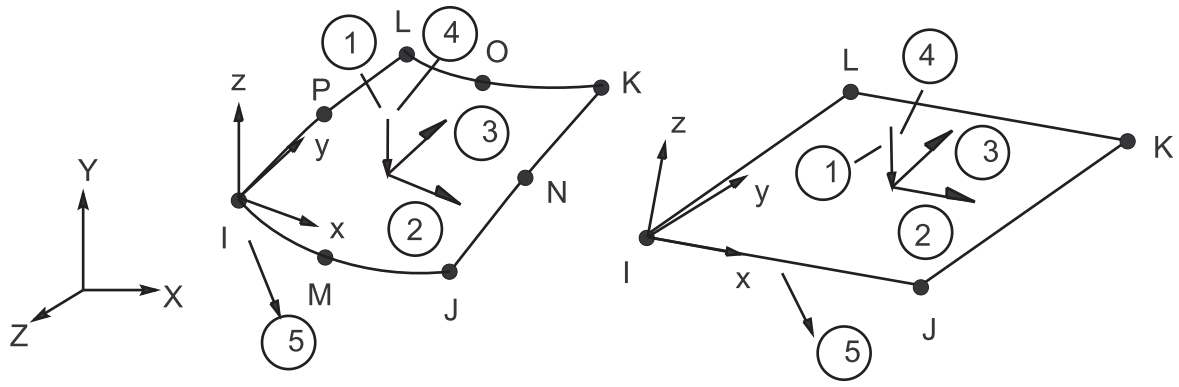
Assuming further that:

$$\varepsilon_y = \varepsilon_z \tag{13.270}$$

yields:

$$\varepsilon_z = -\frac{\varepsilon_x}{2} \tag{13.271}$$

13.154. SURF154 - 3-D Structural Surface Effect



Matrix or Vector	Geometry / Midside Nodes [1]	Shape Functions	Integration Points
Stiffness and Damping Matrices, and Pressure Load Vector	Quad with midside nodes	Equation 11.86 (p. 338)	3 x 3
	Quad without midside nodes	Equation 11.71 (p. 337)	2 x 2
	Triangle with midside nodes	Equation 11.114 (p. 341)	6
	Triangle without midside nodes	Equation 11.68 (p. 337)	3
Mass and Stress Stiffness Matrices	Quad with midside nodes	Equation 11.84 (p. 338), Equation 11.85 (p. 338) and Equation 11.86 (p. 338)	3 x 3
	Quad without midside nodes	Equation 11.69 (p. 337), Equation 11.70 (p. 337) and Equation 11.71 (p. 337)	2 x 2

Matrix or Vector	Geometry / Midside Nodes [1]	Shape Functions	Integration Points
	Triangle with midside nodes	Equation 11.114 (p. 341)	6
	Triangle without midside nodes	Equation 11.66 (p. 337), Equation 11.67 (p. 337) and Equation 11.68 (p. 337)	3
Surface Tension Load Vector	Quad with midside nodes	Equation 11.84 (p. 338) and Equation 11.85 (p. 338)	3 x 3
	Quad without midside nodes	Equation 11.69 (p. 337) and Equation 11.70 (p. 337)	2 x 2
	Triangle with midside nodes	Equation 11.112 (p. 340) and Equation 11.113 (p. 340)	6
	Triangle without midside nodes	Equation 11.66 (p. 337) and Equation 11.67 (p. 337)	3

1. Midside node setting is controlled by KEYOPT(4).

Load Type	Distribution
All Loads	Same as shape functions

The stiffness matrix is:

$$[K_e^f] = \text{element foundation stiffness matrix} = k^f \int_A \{N_Z\}\{N_Z\}^T dA \quad (13.272)$$

where:

k^f = foundation stiffness (input as EFS on **R** command)
A = area of element
$\{N_z\}$ = vector of shape functions representing motions normal to the surface

The mass matrix is:

$$[M_e] = \text{element mass matrix} = \rho \int_A t_h \{N\}\{N\}^T dA + A_d \int_A \{N\}\{N\}^T dA \quad (13.273)$$

where:

t_h = thickness (input as TKI, TKJ, TKK, TKL on **RMORE** command)
ρ = density (input as DENS on **MP** command)
{N} = vector of shape functions
A_d = added mass per unit area (input as ADMSUA on **R** command)

If the command **LUMPM**,ON is used, $[M_e]$ is diagonalized as described in Lumped Matrices (p. 391).

The element damping matrix is:

$$[C_e] = \mu \int_A \{N\}\{N\}^T dA = \text{element damping matrix} \tag{13.274}$$

where:

μ = dissipation (input as VISC on **MP** command)

The element stress stiffness matrix is:

$$[S_e] = \int_A [S_g]^T [S_m][S_g] dA = \text{element mass matrix} \tag{13.275}$$

where:

$[S_g]$ = derivatives of shape functions of normal motions

$$[S_m] = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

s = in-plane force per unit length (input as SURT on **R** command)

If pressure is applied to face 1, the pressure load stiffness matrix is computed as described in Pressure Load Stiffness (p. 46).

The element load vector is:

$$\{F_e\} = \{F_e^{st}\} + \{F_e^{pr}\} \tag{13.276}$$

where:

$$\{F_e^{st}\} = s \int_E \{N_p\} dE = \text{surface tension force vector}$$

$\{N_p\}$ = vector of shape functions representing in-plane motions normal to the edge
E = edge of element

$$\{F_e^{pr}\} = \int_A (\{N_x^P\}P_x + \{N_y^P\}P_y + \{N_z^P\}P_z + P_v Z_f(\tau_x\{N_X\} + \tau_y\{N_Y\} + \tau_z\{N_Z\}))dA$$
$$= \text{pressure load vector}$$

$$\{N_x^P\} = \begin{cases} \{N_x\} & \text{if KEYOPT(2) = 0} \\ \{N_x^e\} & \text{if KEYOPT(2) = 1} \end{cases}$$

$$\{N_y^P\} = \begin{cases} \{N_y\} & \text{if KEYOPT(2) = 0} \\ \{N_y^e\} & \text{if KEYOPT(2) = 1} \end{cases}$$

$$\{N_z^P\} = \begin{cases} \{N_z\} & \text{if KEYOPT(2) = 0} \\ \{N_z^e\} & \text{if KEYOPT(2) = 1} \end{cases}$$

$\{N_x\}$ = vector of shape functions representing motion in element x direction

$\{N_y\}$ = vector of shape functions representing motion in element y direction

$\{N_x^e\}$ = vector of shape functions representing motion in the local coordinate x direction

$\{N_y^e\}$ = vector of shape functions representing motion in the local coordinate y direction

$\{N_z^e\}$ = vector of shape functions representing motion in the local coordinate z direction

$$P_x, P_y, P_z = \begin{cases} \text{distributed pressures over element in element x, y, and z directions (input as VAL1 thru VAL4} \\ \text{with LKEY = 2,3,1, respectively, on SFE command, if KEYOPT(2) = 0} \\ \text{distributed pressures over element in local x, y, and z directions (input as VAL1 thru VAL4} \\ \text{with LKEY = 1,2,3, respectively, on SFE command, if KEYOPT(2) = 1} \end{cases}$$

P_v = uniform pressure magnitude

$$P_v = \begin{cases} P_1 \cos\theta & \text{if KEYOPT(11) = 0 or 1} \\ P_1 & \text{if KEYOPT(11) = 2} \end{cases}$$

P_1 = input (VAL1 with LKEY = 5 on **SFE** command)

θ = angle between element normal and applied load direction

$$Z_f = \begin{cases} 1.0 & \text{if KEYOPT(12) = 0 or } \cos\theta \le 0.0 \\ 0.0 & \text{if KEYOPT(12) = 1 and } \cos\theta > 0.0 \end{cases}$$

$$\tau_x = \begin{cases} D_x \Big/ \sqrt{D_x^2 + D_y^2 + D_z^2} & \text{if KEYOPT(11)} \neq 1 \\ 0.0 & \text{if KEYOPT(11) = 1} \end{cases}$$

$$\tau_y = \begin{cases} D_y \Big/ \sqrt{D_x^2 + D_y^2 + D_z^2} & \text{if KEYOPT(11)} \neq 1 \\ 0.0 & \text{if KEYOPT(11) = 1} \end{cases}$$

$$\tau_z = D_z \Big/ \sqrt{D_x^2 + D_y^2 + D_z^2}$$

D_x, D_y, D_z = vector directions (input as VAL2 thru VAL4 with LKEY = 5 on **SFE** command)

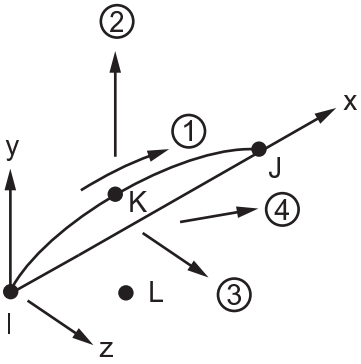
$\{N_X\}$, $\{N_Y\}$, $\{N_Z\}$ = vectors of shape functions in global Cartesian coordinates

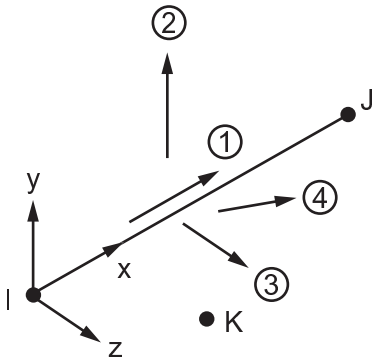
The integration used to arrive at $\{F_e^{pr}\}$ is the usual numerical integration, even if KEYOPT(6) $\neq$ 0. The output quantities "average face pressures" are the average of the pressure values at the integration points.

13.155. Reserved for Future Use

This section is reserved for future use.

13.156. SURF156 - 3-D Structural Surface Line Load Effect



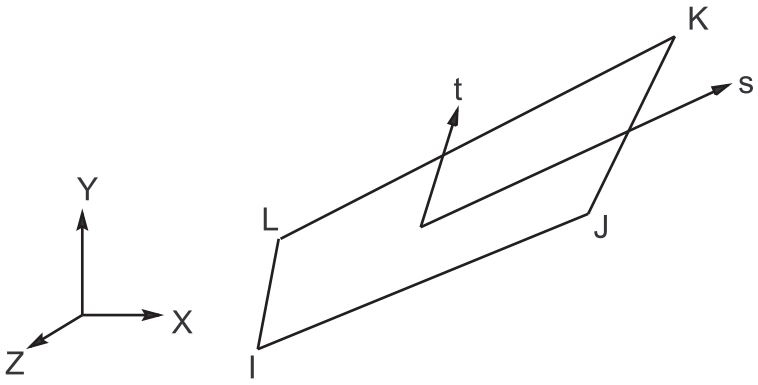


Matrix or Vector	Midside Nodes [1]	Shape Functions	Integration Points
Pressure Load Vector	With midside nodes	Equation 11.19 (p. 330), Equation 11.20 (p. 330), Equation 11.21 (p. 330)	3
	Without midside nodes	Equation 11.15 (p. 330), Equation 11.16 (p. 330), Equation 11.17 (p. 330)	2

1. Midside node setting is controlled by KEYOPT(4).

Load Type	Distribution
Pressures	Linear along length for faces 1, 2, and 3; constant along length for face 4

13.157. SHELL157 - Thermal-Electric Shell



Matrix or Vector	Geometry	Shape Functions	Integration Points
Electrical Conductivity Matrix	Quad	Equation 11.80 (p. 338). No variation thru thickness	2 x 2
	Triangle	Equation 11.80 (p. 338). No variation thru thickness	1
Thermal Conductivity Matrix; Heat Generation Load and Convection Surface Matrix and Load Vectors	Quad	Equation 11.79 (p. 337) and Equation 11.80 (p. 338). No variation thru thickness	2 x 2
	Triangle	Equation 11.106 (p. 340) and Equation 11.80 (p. 338). No variation thru thickness	1
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Lumped Matrices (p. 391)		Same as conductivity matrix

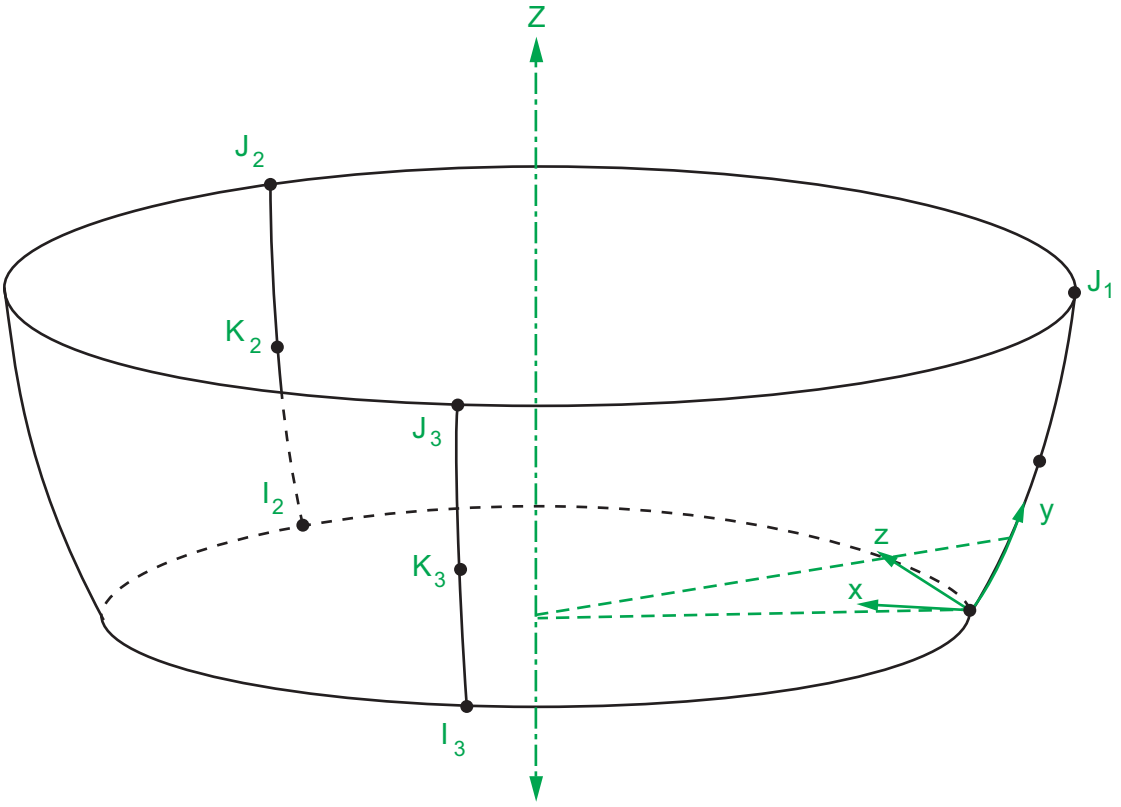
13.157.1. Other Applicable Sections

Coupling (p. 293) discusses coupled effects.

13.158. Not Documented

No detail or element available at this time.

13.159. SURF159 - General Axisymmetric Surface with 2 or 3 Nodes



Matrix or Vector	Midside Nodes [1]	Shape Functions	Integration Points[2]
Mass Matrix	With midside nodes	Equation 11.46 (p. 335) through Equation 11.48 (p. 335)	3 x N_c
	Without midside nodes	Equation 11.43 (p. 335) through Equation 11.45 (p. 335)	2 x N_c
Stress Stiffness Matrix	With midside nodes	Same as mass matrix.	2 x N_c
	Without midside nodes		
Pressure Load Vector	With midside nodes	Same as mass matrix.	2 x N_c
	Without midside nodes		

1. Midside node setting is controlled by KEYOPT(4).
2. N_c = the number of node planes in the circumferential direction. The N_c integration points are circumferentially located at:
   - the nodal planes, and
   - midway between the nodal planes (that is, at the integration planes)

   so that $N_c = (2 * N_{np})$, where N_{np} = number of nodal planes (KEYOPT(2)).

   Exception: If KEYOPT(2) = 1, then $N_c = 1$.

Load Type	Distribution
Pressure	Linear along each face in both directions.

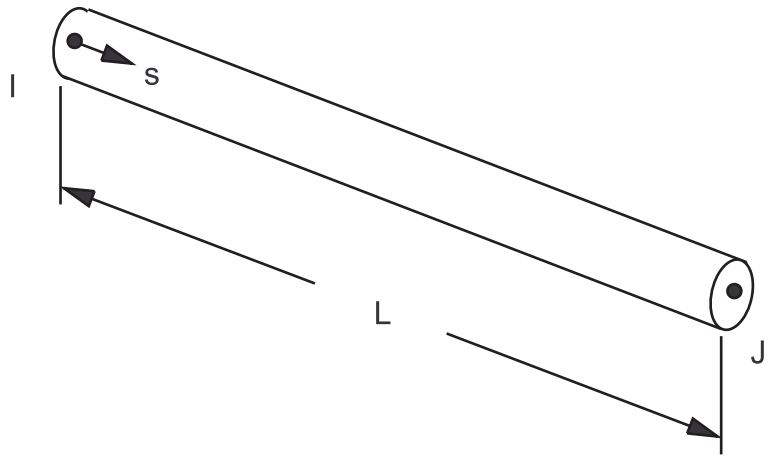
13.159.1. Other Applicable Sections

General Element Formulations (p. 50) gives the general element formulations used by this element.

13.159.2. Assumptions and Restrictions

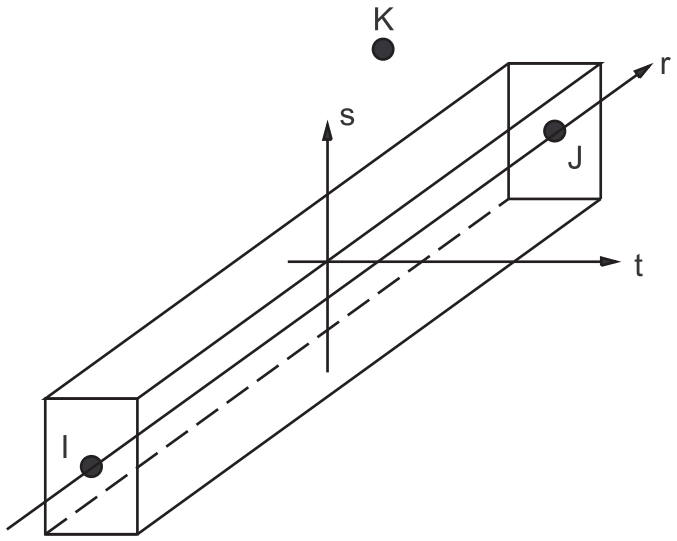
Although the elements are initially axisymmetric, the loads and deformation can be general in non-axisymmetric 3-D. The displacements are interpolated in elemental coordinate system by interpolation functions, but the user can define the nodal displacements in any direction.

13.160. LINK160 - Explicit 3-D Spar (or Truss)



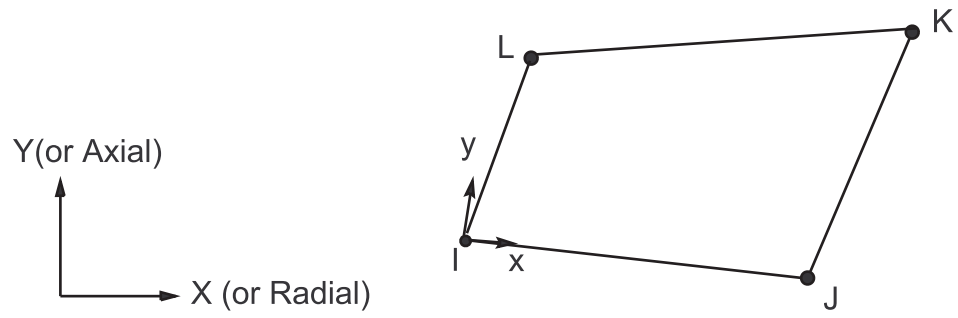
For all theoretical information about this element, see the *LS-DYNA Theoretical Manual*([199] (p. 932)).

13.161. BEAM161 - Explicit 3-D Beam



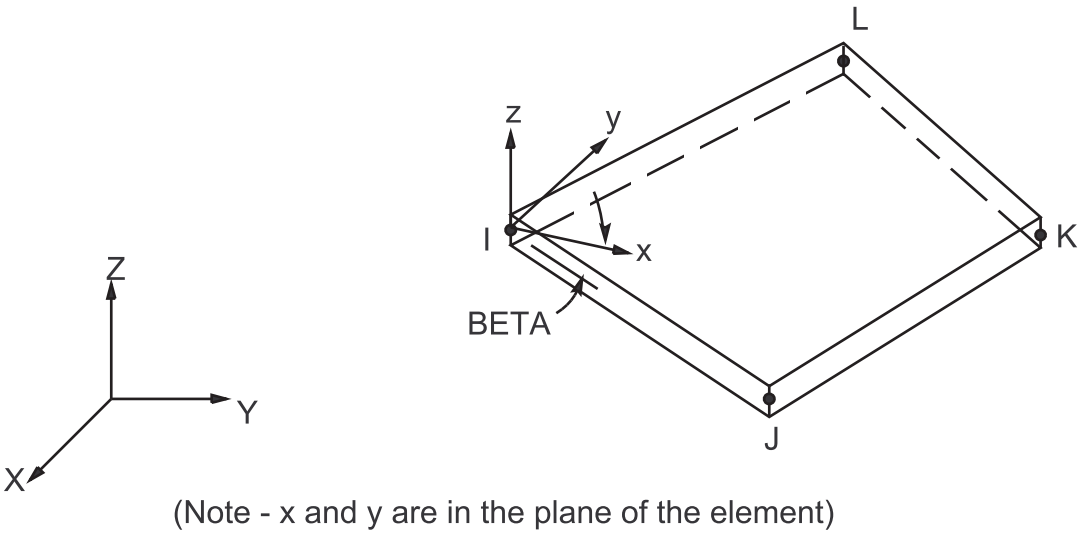
For all theoretical information about this element, see the *LS-DYNA Theoretical Manual*([199] (p. 932)).

13.162. PLANE162 - Explicit 2-D Structural Solid



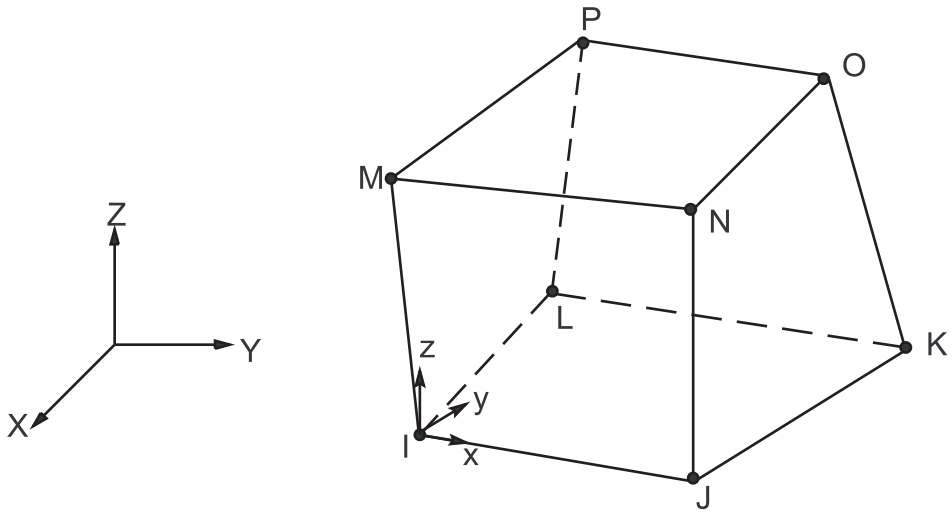
For all theoretical information about this element, see the *LS-DYNA Theoretical Manual*([199] (p. 932)).

13.163. SHELL163 - Explicit Thin Structural Shell



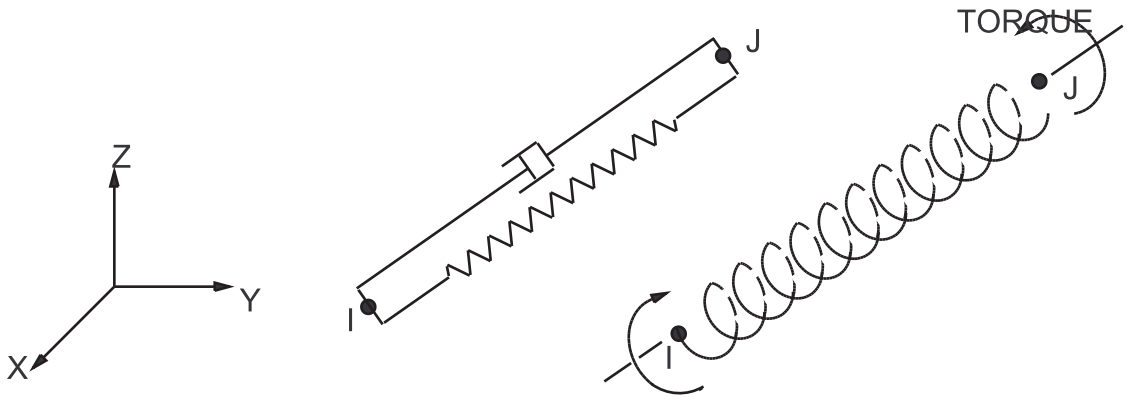
For all theoretical information about this element, see the *LS-DYNA Theoretical Manual*([199] (p. 932)).

13.164. SOLID164 - Explicit 3-D Structural Solid



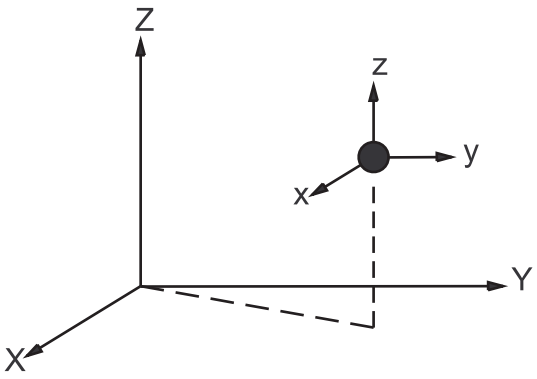
For all theoretical information about this element, see the *LS-DYNA Theoretical Manual*([199] (p. 932)).

13.165. COMBI165 - Explicit Spring-Damper



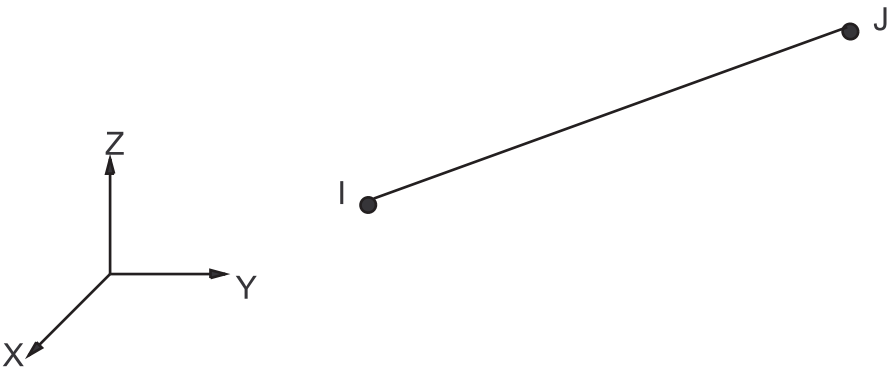
For all theoretical information about this element, see the *LS-DYNA Theoretical Manual*([199] (p. 932)).

13.166. MASS166 - Explicit 3-D Structural Mass



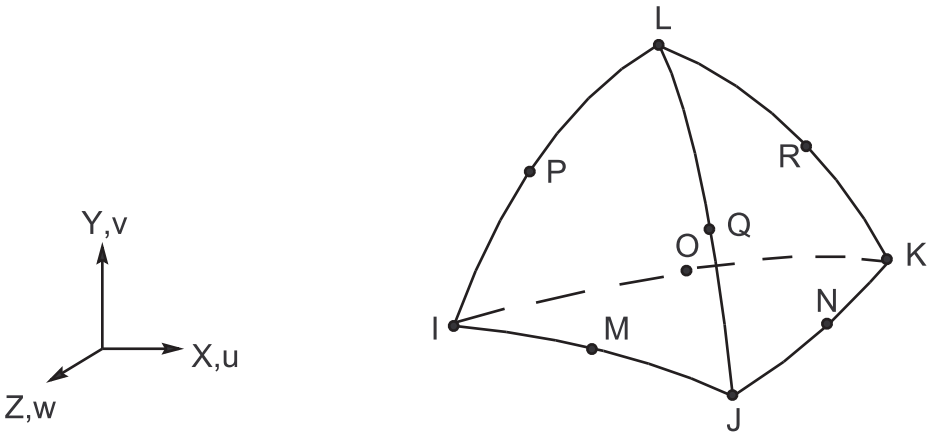
For all theoretical information about this element, see the *LS-DYNA Theoretical Manual*([199] (p. 932)).

13.167. LINK167 - Explicit Tension-Only Spar



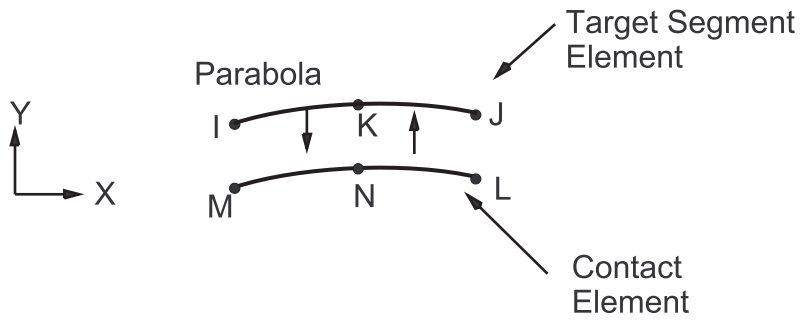
For all theoretical information about this element, see the *LS-DYNA Theoretical Manual*([199] (p. 932)).

13.168. SOLID168 - Explicit 3-D 10-Node Tetrahedral Structural Solid



For all theoretical information about this element, see the *LS-DYNA Theoretical Manual*([199] (p. 932)).

13.169. TARGE169 - 2-D Target Segment



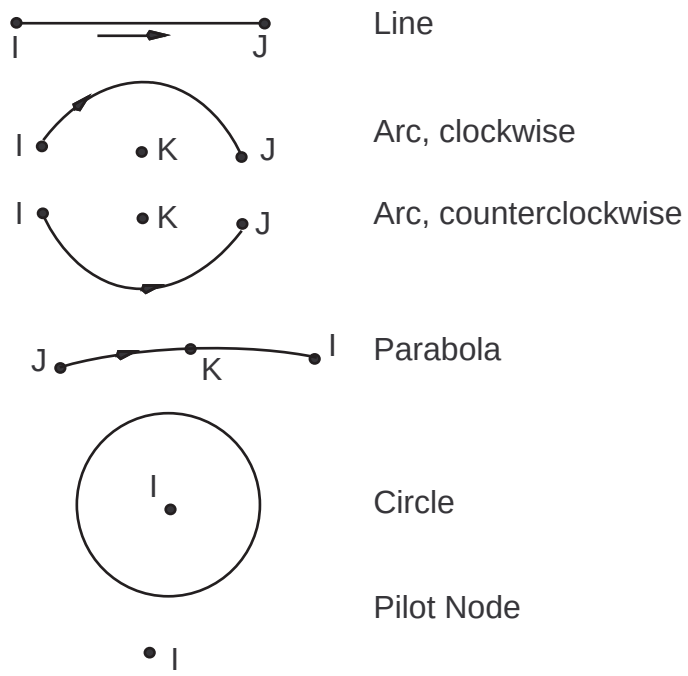
13.169.1. Other Applicable Sections

TARGE170 - 3-D Target Segment (p. 551) discusses Target Elements.

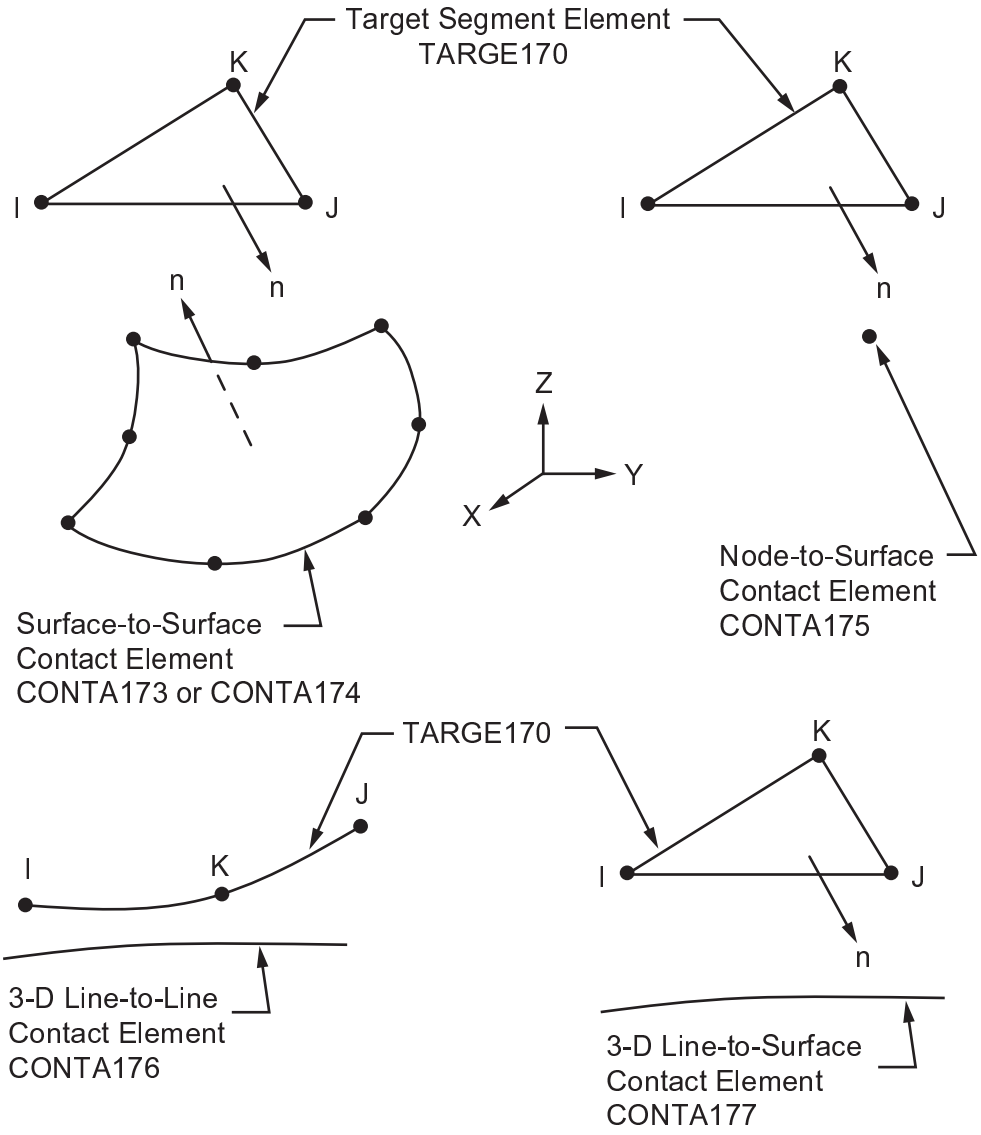
13.169.2. Segment Types

TARGE169 supports six 2-D segment types:

Figure 13.27: 2-D Segment Types



13.170. TARGE170 - 3-D Target Segment



13.170.1. Introduction

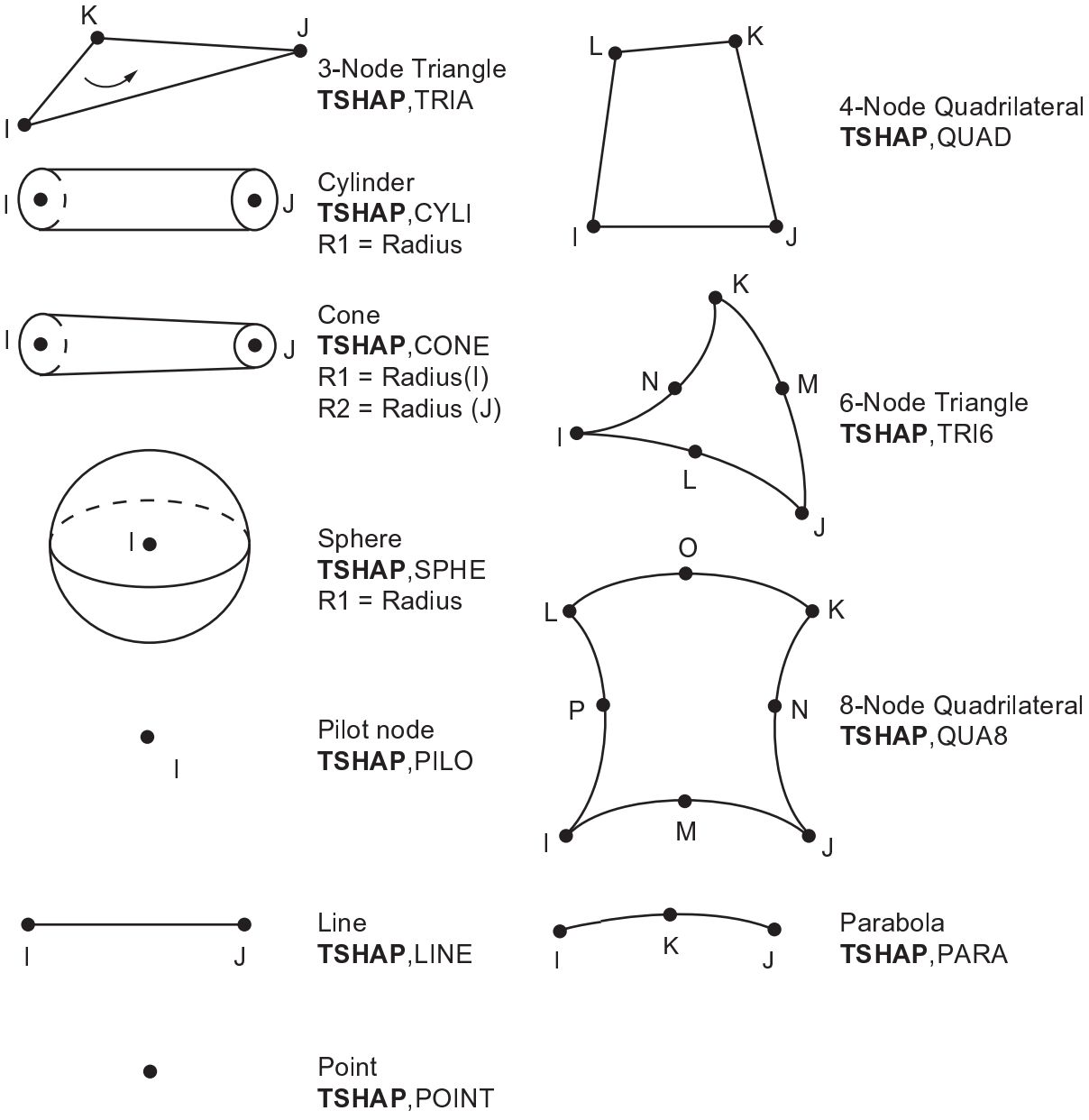
In studying the contact between two bodies, the surface of one body is conventionally taken as a contact surface and the surface of the other body as a target surface. The "contact-target" pair concept has been widely used in finite element simulations. For rigid-flexible contact, the contact surface is associated with the deformable body; and the target surface must be the rigid surface. For flexible-flexible contact, both contact and target surfaces are associated with deformable bodies. The contact and target surfaces constitute a "Contact Pair".

TARGE170 is used to represent various 3-D target surfaces for the associated contact elements (CONTA173, CONTA174, CONTA175, CONTA176, and CONTA177). The contact elements themselves overlay the solid elements, line elements, or shell element edges describing the boundary of a deformable body that is potentially in contact with the rigid target surface, defined by TARGE170. Hence, a "target" is simply a geometric entity in space that senses and responds when one or more contact elements move into a target segment element.

13.170.2. Segment Types

The target surface is modelled through a set of target segments; typically several target segments comprise one target surface. Each target segment is a single element with a specific shape or segment type. TARGE170 supports ten 3-D segment types; see Figure 13.28: 3-D Segment Types (p. 552)

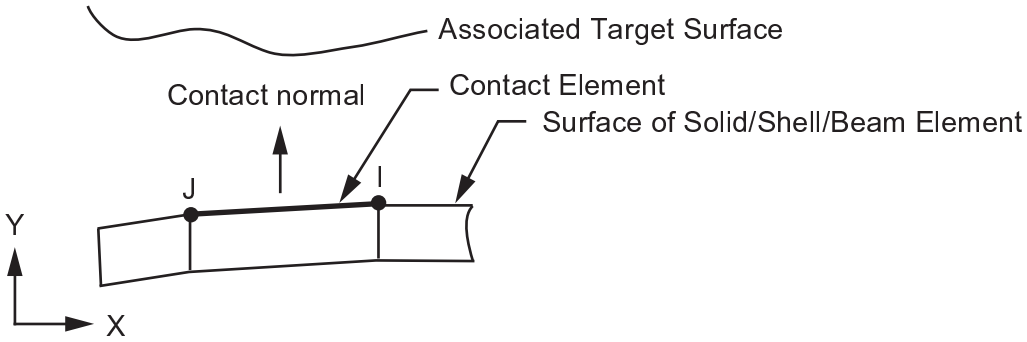
Figure 13.28: 3-D Segment Types



13.170.3. Reaction Forces

The reaction forces on the entire rigid target surface are obtained by summing all the nodal forces of the associated contact elements. The reaction forces are accumulated on the pilot node. If the pilot node has not been explicitly defined by the user, one of the target nodes (generally the one with the smallest number) will be used to accumulate the reaction forces.

13.171. CONTA171 - 2-D 2-Node Surface-to-Surface Contact

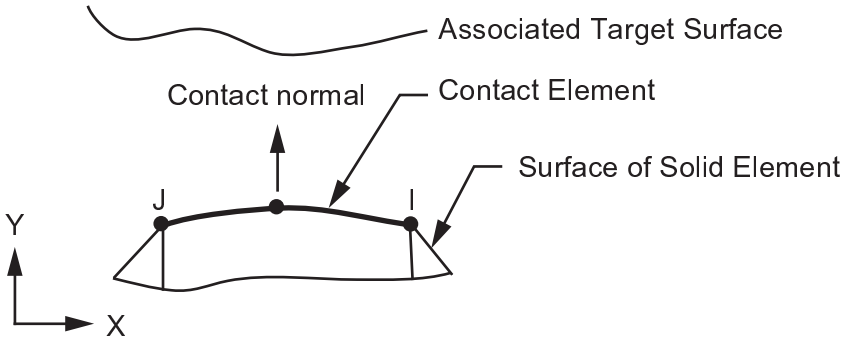


Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	$W = C_1 + C_2 x$	2

13.171.1. Other Applicable Sections

The CONTA171 description is the same as for CONTA174 - 3-D 8-Node Surface-to-Surface Contact (p. 554) except that it is 2-D and there are no midside nodes.

13.172. CONTA172 - 2-D 3-Node Surface-to-Surface Contact

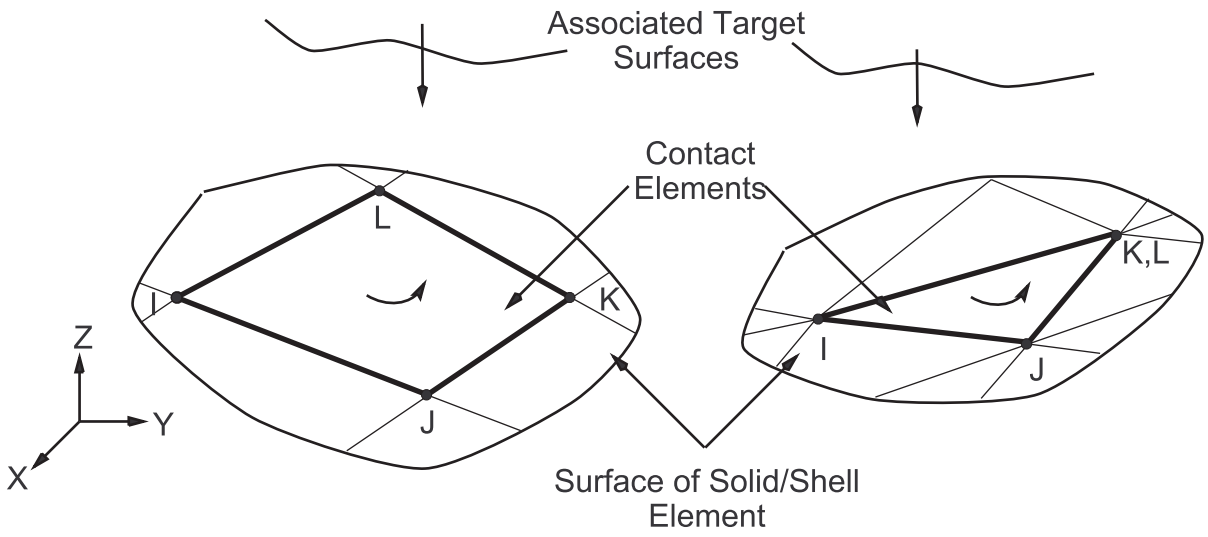


Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	$W = C_1 + C_2 x + C_3 x^2$	2

13.172.1. Other Applicable Sections

The CONTA172 description is the same as for CONTA174 - 3-D 8-Node Surface-to-Surface Contact (p. 554) except that it is 2-D.

13.173. CONTA173 - 3-D 4-Node Surface-to-Surface Contact

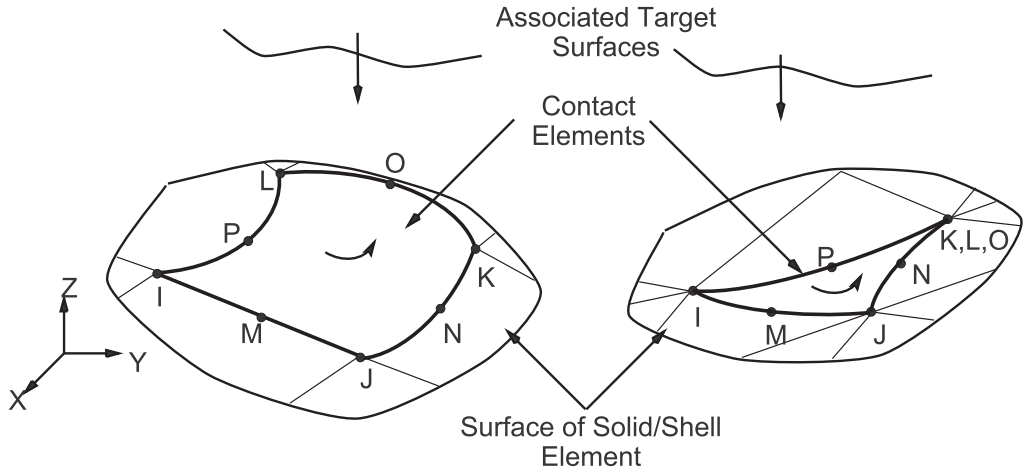


Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices	Quad	Equation 11.69 (p. 337), Equation 11.70 (p. 337), and Equation 11.71 (p. 337)	2 x 2
	Triangle	Equation 11.66 (p. 337), Equation 11.67 (p. 337), and Equation 11.68 (p. 337)	3

13.173.1. Other Applicable Sections

The CONTA173 description is the same as for CONTA174 - 3-D 8-Node Surface-to-Surface Contact (p. 554) except there are no midside nodes.

13.174. CONTA174 - 3-D 8-Node Surface-to-Surface Contact



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices	Quad	If KEYOPT(4) = 0 (has midside nodes) Equation 11.84 (p. 338), Equation 11.85 (p. 338), and Equation 11.86 (p. 338)	2 x 2
	Triangle	If KEYOPT(4) = 0 (has midside nodes) Equation 11.114 (p. 341)	3

The following CONTA174 topics are available:

13.174.1. Introduction

CONTA174 is an 8-node element that is intended for general rigid-flexible and flexible-flexible contact analysis. In a general contact analysis, the area of contact between two (or more) bodies is generally not known in advance. CONTA174 is applicable to 3-D geometries. It may be applied for contact between solid bodies or shells.

13.174.2. Contact Kinematics

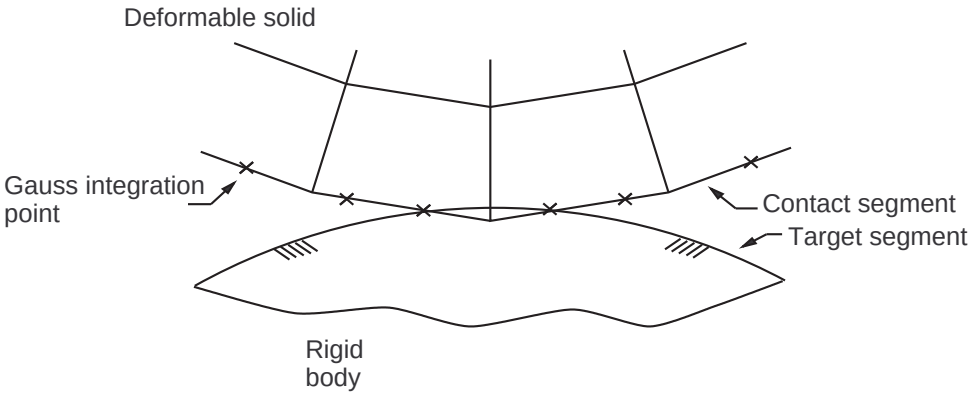
Contact Pair

In studying the contact between two bodies, the surface of one body is conventionally taken as a contact surface and the surface of the other body as a target surface. For rigid-flexible contact, the contact surface is associated with the deformable body; and the target surface must be the rigid surface. For flexible-flexible contact, both contact and target surfaces are associated with deformable bodies. The contact and target surfaces constitute a "Contact Pair".

The CONTA174 contact element is associated with the 3-D target segment elements (TARGE170) using a shared real constant set number. This element is located on the surface of 3-D solid, shell elements (called underlying element). It has the same geometric characteristics as the underlying elements. The contact surface can be either side or both sides of the shell or beam elements.

Location of Contact Detection

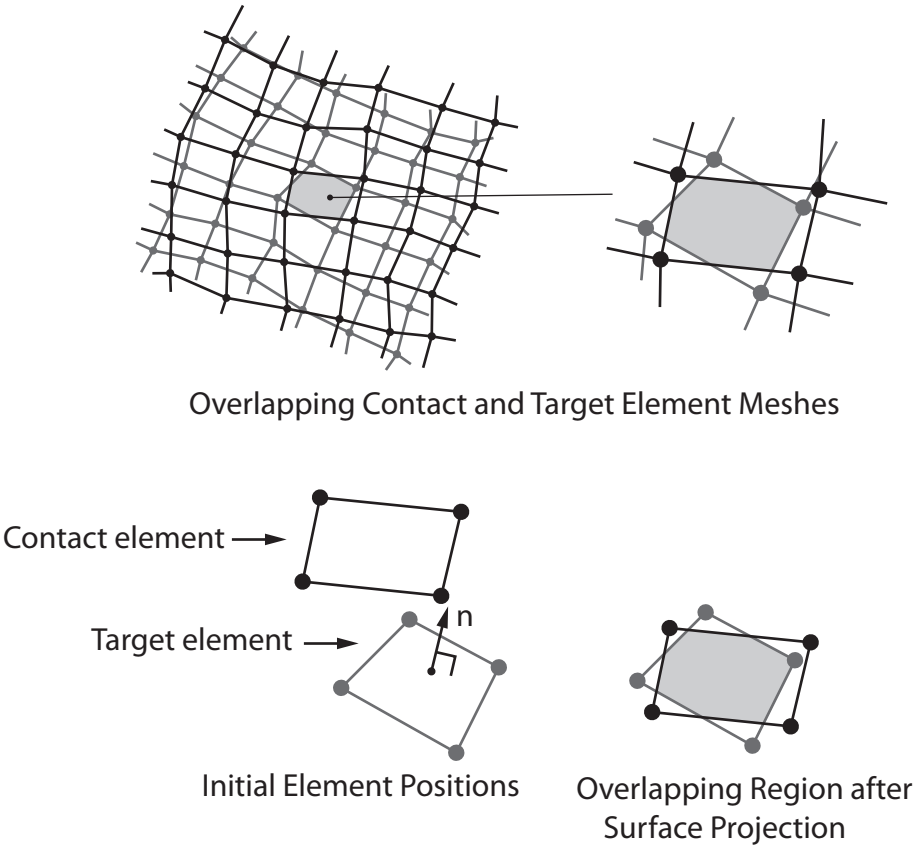
Figure 13.29: Contact Detection Point Location at Gauss Point



CONTA174 is a surface-to-surface contact element. The contact detection points are the integration points and are located either at nodal points or Gauss points. The contact element is constrained against penetration into the target surface at its integration points. However, the target surface can, in principle, penetrate through into the contact surface. See Figure 13.29: Contact Detection Point Location at Gauss Point (p. 556). CONTA174 uses Gauss integration points by default (Cescotto and Charlier([213] (p. 932)), Cescotto and Zhu([214] (p. 932))), which generally provides more accurate results than when using the nodes themselves as the integration points. A disadvantage of using nodal contact points is that, for a uniform pressure, the kinematically equivalent forces at the nodes are unrepresentative and indicate release at corners.

The surface projection method (based on Puso and Laursen ([382] (p. 942) and [383] (p. 942))) is available for the 3-D elements, CONTA173 and CONTA174. This option enforces contact constraints on an overlapping region of contact and target surfaces (see Figure 13.30: Surface-Projection-Based Contact (p. 557)) rather than on individual contact nodes or Gauss points. This method passes patch tests and generally offers smoother stress distributions (for both contact elements and underlying elements) than the other contact detection options. The contact forces vary smoothly during large sliding, and the forces do not jump when contact nodes slide off edges of target surfaces. The surface projection method also satisfies moment equilibrium for frictional or bonded contact. Other contact detection options may not satisfy moment equilibrium exactly when an offset exists along the contact normal direction under certain circumstances (for example, when using a penalty based formulation, when accounting for shell thickness effects, or when using a user-defined contact offset).

Figure 13.30: Surface-Projection-Based Contact



Penetration Distance

The penetration distance is measured along the normal direction of the contact surface (at integration points) to the target surface (Cescotto and Charlier([214] (p. 932))). See Figure 13.31: Penetration Distance (p. 557). It is uniquely defined even when the geometry of the target surface is not smooth. Such discontinuities may be due to physical corners on the target surface, or may be introduced by a numerical discretization process (e.g. finite elements). Based on the present way of calculating penetration distance, there is no restriction on the shape of the rigid target surface. Smoothing is not always necessary for the concave corner. For the convex corner, it is still recommended to smooth out the region of abrupt curvature changes (see Figure 13.32: Smoothing Convex Corner (p. 558)).

Figure 13.31: Penetration Distance

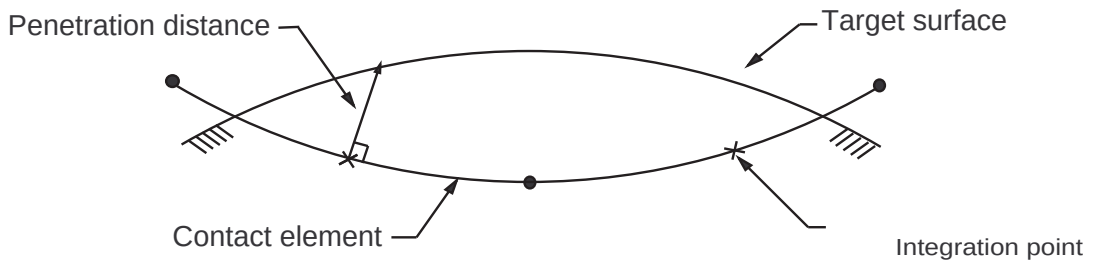
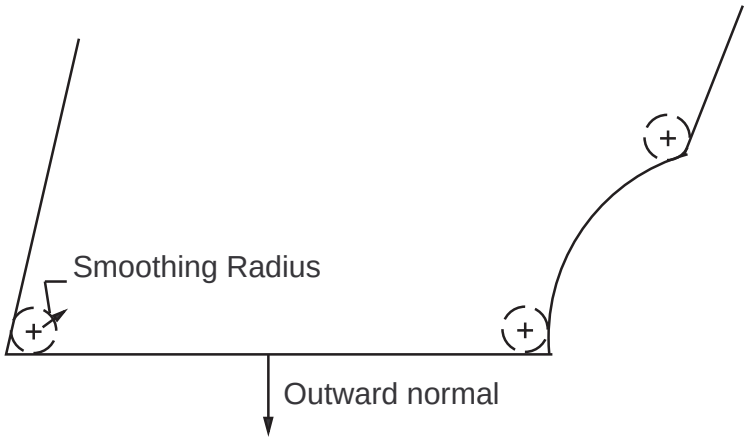


Figure 13.32: Smoothing Convex Corner



For the surface-projection-based contact method, the contact detection remains at contact nodal points and the contact normal is perpendicular to the contact surface. However, the contact penetration distance is computed over the overlapping region (see Figure 13.30: Surface-Projection-Based Contact (p. 557)) in an average sense.

Pinball Algorithm

The position and the motion of a contact element relative to its associated target surface determine the contact element status. The program monitors each contact element and assigns a status:

STAT = 0 Open far-field contact
STAT = 1 Open near-field contact
STAT = 2 Sliding contact
STAT = 3 Sticking contact

A contact element is considered to be in near-field contact when the element enters a pinball region, which is centered on the integration point of the contact element. The computational cost of searching for contact depends on the size of the pinball region. Far-field contact element calculations are simple and add few computational demands. The near-field calculations (for contact elements that are nearly or actually in contact) are slower and more complex. The most complex calculations occur the elements are in actual contact.

Setting a proper pinball region is useful to overcome spurious contact definitions if the target surface has several convex regions. The current default setting should be appropriate for most contact problems.

13.174.3. Frictional Model

Coulomb's Law

In the basic Coulomb friction model, two contacting surfaces can carry shear stresses. When the equivalent shear stress is less than a limit frictional stress (τ_{lim}), no motion occurs between the two surfaces. This state is known as sticking. The Coulomb friction model is defined as:

$$\tau_{lim} = \mu P + b \tag{13.277}$$

$$\left\| \tau \right\| \le \tau_{lim} \tag{13.278}$$

where:

τ_{lim} = limit frictional stress

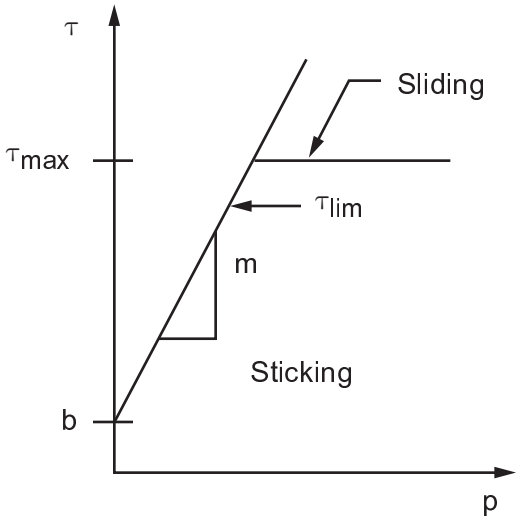
$$\|\tau\| = \begin{cases} |\tau| & \text{equivalent stress for 2-D contact} \\ \sqrt{\tau_1^2 + \tau_2^2} & \text{equivalent stress for 3-D contact} \end{cases}$$

μ = coefficient of friction for isotropic friction (input as MU using either **TB** command with *Lab* = FRIC or **MP** command; orthotropic friction defined below
P = contact normal pressure
b = contact cohesion (input as COHE on **R** command)

Once the equivalent frictional stress exceeds τ_{lim}, the contact and target surfaces will slide relative to each other. This state is known as sliding. The sticking/sliding calculations determine when a point transitions from sticking to sliding or vice versa. The contact cohesion provides sliding resistance even with zero normal pressure.

CONTA174 provides an option for defining a maximum equivalent frictional stress τ_{max} (input as TAUMAX on **RMORE** command) so that, regardless of the magnitude of the contact pressure, sliding will occur if the magnitude of the equivalent frictional stress reaches this value.

Figure 13.33: Friction Model



Contact elements offer two models for Coulomb friction: isotropic friction and orthotropic friction.

Isotropic Friction

The isotropic friction model uses a single coefficient of friction μ_{iso} based on the assumption of uniform stick-slip behavior in all directions. It is available with all 2-D and 3-D contact elements (CONTA171, CONTA172, CONTA173, CONTA174, CONTA175, CONTA176, CONTA177, and CONTA178).

Orthotropic Friction

The orthotropic friction model is based on two coefficients of friction, μ_1 and μ_2, to model different stick-slip behavior in different directions. Use orthotropic friction model in 3-D contact only (CONTA173, CONTA174, CONTA175, CONTA176, and CONTA177). The two coefficients are defined in two orthogonal sliding directions called the principal directions (see *Element Reference* for more details). The frictional stress in principal direction i=1,2 is given by:

$$\tau_i = \mu_i P + b \tag{13.279}$$

By appropriately scaling the frictional stresses in principal directions the expressions for scaled limit frictional stress (τ'_{lim}) and scaled equivalent frictional stress ($\|\tau'\|$) for orthotropic friction can be written in a form similar to isotropic friction (Michalowski and Mroz([361] (p. 941))):

$$\tau'_i = \left(\frac{\mu_{eq}}{\mu_i}\right)\tau_i \tag{13.280}$$

$$\mu_{eq} = \sqrt{\frac{(\mu_1^2 + \mu_2^2)}{2}} \tag{13.281}$$

$$\tau'_{lim} = \mu_{eq} P + b \tag{13.282}$$

$$\|\tau'\| = \sqrt{\tau'^2_1 + \tau'^2_2} \le \tau'_{lim} \tag{13.283}$$

where:

τ'_i = scaled frictional stress in direction i = 1,2

$\|\tau'\|$ = scaled equivalent frictional stress

τ'_{lim} = scaled limit frictional stress

μ_{eq} = equivalent coefficient of friction for orthotropic friction

μ_1, μ_2 = coefficients of friction in first and second principal directions (input as MU1 and MU2 using **TB** command with *Lab* = FRIC)

While scaled frictional stresses are used for friction computations, actual frictional stresses are output after applying the inverse scaling in Equation 13.280 (p. 560).

The coefficient of friction (μ_1 and μ_2) can have dependence on time, temperature, normal pressure, sliding distance, or sliding relative velocity (defined as fields on the **TBFIELD** command). Suitable combinations of up to two fields can be used to define dependency, for example, temperature or temperature and sliding distance; see Contact Friction in the *Material Reference* for details.

Static and Dynamic Friction

CONTA174 provides the exponential friction model, which is used to smooth the transition between the static coefficient of friction and the dynamic coefficient of friction according to the formula (Benson and Hallquist([317] (p. 938))):

$$\mu(v) = \mu_d + (\mu_s - \mu_d)e^{-c|v|} \tag{13.284}$$

where:

v = slip rate

$$\mu_d = \begin{cases} \mu_{iso}, \text{ coefficient of friction for isotropic friction (input as MU using either TB command with } Lab \text{ = FRIC or MP command)} \\ \mu_{eq}, \text{ equivalent coefficient of friction for orthotropic friction (defined below)} \end{cases}$$

$\mu_s = R_f \mu_d$ = static coefficient of friction

R_f = ratio of static and dynamic friction (input as FACT on **RMORE** command)

c = decay coefficient (input as DC on **RMORE** command)

Integration of Frictional Law

The integration of the frictional mode is similar to that of nonassociated theory of plasticity (see Rate-Independent Plasticity (p. 64)). In each substep that sliding friction occurs, an elastic predictor is computed in contact traction space. The predictor is modified with a radial return mapping function, providing both a small elastic deformation along with a sliding response as developed by Giannakopoulos([135] (p. 928)).

The flow rule giving the slip increment for orthotropic friction can be written as:

$$du_i = \lambda \left\{ \frac{\partial(\|\tau'\| - \tau'_{lim})}{\partial \tau_i} \right\} \tag{13.285}$$

where:

du_i = slip increment in principal direction i = 1, 2
λ = Lagrange multiplier for friction

By appropriately scaling the slip increment, it can be shown that the Lagrange multiplier is equal to the scaled equivalent slip increment:

$$\lambda = \|du'\| = \sqrt{du'^2_1 + du'^2_2} \tag{13.286}$$

$$du'_i = \left(\frac{\mu_i}{\mu_{eq}} \right) du_i \tag{13.287}$$

and the direction of scaled slip increment is same as that of scaled frictional stress.

$$\frac{du'_i}{\|du'\|} = \frac{\tau'_i}{\|\tau'\|} \tag{13.288}$$

Thus, computations for orthotropic friction use the same framework as isotropic friction except for scaled slip increments and scaled frictional stresses which are converted to actual values for output.

User-defined Friction

For friction models that do not follow Coulomb's law, you can write a USERFRIC subroutine. Refer to the *Guide to User-Programmable Features* for a detailed description on writing a USERFRIC subroutine. You can use it with any 2-D or 3-D contact element (CONTA171, CONTA172, CONTA173, CONTA174, CONTA175, CONTA176, CONTA177, and CONTA178) with penalty method for tangential contact (select KEYOPT(2) = 0, 1, or 3). Use the **TB**,FRIC command with *TBOPT* = USER to choose the user define friction option, and specify the friction properties on the **TBDATA** command.

Friction models involve nonlinear material behavior, so only experienced users who have a good understanding of the theory and finite element programming should attempt to write a USERFRIC subroutine.

Constitutive Relationship for Frictional Sliding

Regarding frictional sliding, the linearized constitutive relationship between tangential stresses and sliding increments can be expressed:

$$d\tau_i = t_i\left(\mu + P\frac{\partial\mu}{\partial P}\right)dP + \left(\delta_{ij} - t_i t_j\right)\frac{\tau_{lim}}{\left\|u^{pred}\right\|}du_j + t_i t_j\frac{P}{\Delta t}\frac{\partial\mu}{\partial\left\|\dot{u}\right\|}du_j \tag{13.289}$$

where:

$\left\|u^{pred}\right\|$ = equivalent slip predictor over the current substep (Laursen and Simo[216] (p. 933))

$\left\|\dot{u}\right\|$ = current equivalent slip rate

t_i = the unit vector in the direction of slip (i = 1, 2)

$\frac{\partial\mu}{\partial P}$ can be obtained from the tabular input (define the friction coefficient to be a function of pressure via the **TB**,FRIC command).

$\frac{\partial\mu}{\partial\left\|\dot{u}\right\|}$ can be obtained from the tabular input (define friction coefficient to be a function of sliding velocity via **TB**,FRIC command) or from the definition of *Static and Dynamic Friction* (p. 560) via real constants FACT and DC.

Algorithmic Symmetrization

Contact problems involving friction produce non-symmetric stiffness. Using an unsymmetric solver (**NROPT**,UNSYM) is more computationally expensive than a symmetric solver for each iteration. For this reason, a symmetrization algorithm developed by Laursen and Simo([216] (p. 933)) is used by which most frictional contact problems can be solved using solvers for symmetric systems. If frictional stresses have a substantial influence on the overall displacement field and the magnitude of the frictional stresses is highly solution dependent, any symmetric approximation to the stiffness matrix may provide a low rate of convergence. In such cases, the use of an unsymmetric stiffness matrix is more computationally efficient.

Forced Frictional Sliding Using Velocity Input

In a static analysis, you can model steady-state frictional sliding between two flexible bodies or between a flexible and a rigid body with different velocities. In this case the sliding velocities no longer follow the nodal displacements, and they are predefined through the **CMROTATE** command. This command sets the velocities on the nodes of the element component as an initial condition at the start of a load step. The program determines the sliding direction based on the direction of the sliding velocities. The frictional stresses then can be expressed by:

$$\tau_i = \tau_{lim}\frac{\dot{u}_i}{\left\|\dot{u}\right\|} \tag{13.290}$$

where $\dot{u}_i$ is obtained from the **CMROTATE** command.

The linearized frictional stress with respect to the sliding velocity can be expressed in the following form (Bajer, Belsky, and Kung[397] (p. 943)):

$$d\tau_i = t_i t_j P\frac{\partial\mu}{\partial\left\|\dot{u}\right\|}d\dot{u}_j + t_i\left(\delta_{ij} - t_i t_j\right)\frac{\tau_{lim}}{\left\|\dot{u}\right\|}d\dot{u}_j \tag{13.291}$$

In the complex eigenvalue extraction analysis using the QRDAMP or DAMP method, the above terms will be included in the damping matrix. The first term is, in general, negative in the case of a negative friction-sliding velocity gradient, and it can destabilize the system. Therefore, it is referred to as destabilizing squeal damping. The destabilizing squeal damping is proportional to the contact pressure, but it follows the sliding direction. The second term is proportional to the equivalent friction stress but orthogonal to the sliding direction. It may be needed to avoid false unstable modes and is, therefore, referred to as stabilizing squeal damping (Bajer, Belsky, and Kung[397] (p. 943)). To study the effects of squeal damping on the complex eigenvalue extraction analysis, the above equation is rewritten:

$$d\tau_i = t_i t_j c_d P \frac{\partial \mu}{\partial \|\dot{u}\|} d\dot{u}_j + t_i \left(\delta_{ij} - t_i t_j\right) c_s \frac{\tau_{lim}}{\|\dot{u}\|} d\dot{u}_j \tag{13.292}$$

where:

c_d = a scaling factor for destabilizing squeal damping (real constant FDMD when KEYOPT(16) = 0)

c_s = a scaling factor for stabilizing squeal damping (real constant FDMS when KEYOPT(16) = 0)

$\frac{\partial \mu}{\partial \|\dot{u}\|}$ = friction-sliding velocity gradient (real constant FDMD when KEYOPT(16) = 1)

$c_d P \frac{\partial \mu}{\partial \|\dot{u}\|}$ = destabilizing damping coefficient (real constant FDMD when KEYOPT(16) = 2)

$c_s \frac{\tau_{lim}}{\|\dot{u}\|}$ = stabilizing damping coefficient (real constant FDMS when KEYOPT(16) = 1 or 2)

13.174.4. Contact Algorithm

Four different contact algorithms are implemented in this element (selected by KEYOPT(2)).

- Pure penalty method
- Augmented Lagrangian method (Simo and Laursens([215] (p. 932)))
- Pure Lagrange multiplier method (Bathe([2] (p. 921)))
- Lagrange multiplier on contact normal and penalty on frictional direction

Pure Penalty Method

This method requires both contact normal and tangential stiffness. The main drawback is that the amount penetration between the two surfaces depends on this stiffness. Higher stiffness values decrease the amount of penetration but can lead to ill-conditioning of the global stiffness matrix and to convergence difficulties. Ideally, you want a high enough stiffness that contact penetration is acceptably small, but a low enough stiffness that the problem will be well-behaved in terms of convergence or matrix ill-conditioning.

The contact traction vector is:

$$\begin{Bmatrix} P \\ \tau_1 \\ \tau_2 \end{Bmatrix} \tag{13.293}$$

where:

P = normal contact pressure
τ_1 = frictional stress in direction 1
τ_2 = frictional stress in direction 2

The contact pressure is:

$$P = \begin{cases} 0 & \text{if } u_n > 0 \\ K_n u_n & \text{if } u_n \le 0 \end{cases} \tag{13.294}$$

where:

K_n = contact normal stiffness
u_n = contact gap size

The frictional stress for isotropic friction is obtained by Coulomb's law:

$$\tau_i = \begin{cases} \tau_i^{n-1} + K_s \Delta u_i & \text{if } \|\tau\| = \sqrt{\tau_1^2 + \tau_2^2} - \mu_{iso} P < 0 \,(\text{sticking}) \\ \mu_{iso} P \dfrac{\Delta u_i}{\|\Delta u\|} & \text{if } \|\tau\| = \sqrt{\tau_1^2 + \tau_2^2} - \mu_{iso} P = 0 \,(\text{sliding}) \end{cases} \tag{13.295}$$

where:

K_s = tangential contact stiffness (input as FKT on **R** command)
Δu_i = slip increment in direction i over the current substep; In the case of sticking, Δu_i is the elastic slip increment, which represents the reversible tangential motion within the current substep.
$\|\Delta u\|$ = equivalent slip increment over the current substep
μ_{iso} = coefficient of friction
τ_i^{n-1} = frictional stress in direction i = 1,2 at the end of previous substep

For orthotropic friction, slip increment and frictional stress are scaled so that

$$\tau_i' = \begin{cases} \tau_i'^{n-1} + K_{si}' \Delta u_i' & \text{if } \|\tau'\| = \sqrt{\tau_1'^2 + \tau_2'^2} - \mu_{eq} P < 0 \,(\text{sticking}) \\ \mu_{eq} P \dfrac{\Delta u_i'}{\|\Delta u'\|} & \text{if } \|\tau'\| = \sqrt{\tau_1'^2 + \tau_2'^2} - \mu_{eq} P = 0 \,(\text{sliding}) \end{cases} \tag{13.296}$$

where:

K_{si}' = scaled tangential contact stiffness in principal direction i = 1, 2
$\Delta u_i'$ = slip increment in principal direction i = 1, 2 over the current substep
$\|\Delta u'\|$ = scaled equivalent slip increment over the current substep
μ_{eq} = equivalent coefficient of friction
$\tau_i'^{n-1}$ = scaled frictional stress in principal direction i = 1, 2 at the end of previous substep

For consistency between scaled friction stress and scaled slip increment, the scaled tangential contact stiffness in principal direction i = 1, 2 must be defined as:

$$K'_{si} = \left(\frac{\mu_{eq}}{\mu_i} \right)^2 K_s \tag{13.297}$$

Augmented Lagrangian Method

The augmented Lagrangian method is an iterative series of penalty updates to find the Lagrange multipliers (i.e., contact tractions). Compared to the penalty method, the augmented Lagrangian method usually leads to better conditioning and is less sensitive to the magnitude of the contact stiffness coefficient. However, in some analyses, the augmented Lagrangian method may require additional iterations, especially if the deformed mesh becomes excessively distorted.

The contact pressure is defined by:

$$P = \begin{cases} 0 & \text{if } u_n > 0 \\ K_n u_n + \lambda_{i+1} & \text{if } u_n \le 0 \end{cases} \tag{13.298}$$

where:

$$\lambda_{i+1} = \begin{cases} \lambda_i + K_n u_n & \text{if } |u_n| > \varepsilon \\ \lambda_i & \text{if } |u_n| < \varepsilon \end{cases}$$

ε = compatibility tolerance (input as FTOLN on **R** command)
λ_i = Lagrange multiplier component at iteration i

The Lagrange multiplier component λ_i is computed locally (for each element) and iteratively.

Pure Lagrange Multiplier Method

The pure Lagrange multiplier method does not require contact stiffness. Instead it requires chattering control parameters. Theoretically, the pure Lagrange multiplier method enforces zero penetration when contact is closed and "zero slip" when sticking contact occurs. However the pure Lagrange multiplier method adds additional degrees of freedom to the model and requires additional iterations to stabilize contact conditions. This will increase the computational cost. This algorithm has chattering problems due to contact status changes between open and closed or between sliding and sticking. The other main drawback of the Lagrange multiplier method is overconstraint in the model. The model is overconstrained when a contact constraint condition at a node conflicts with a prescribed boundary condition on that degree of freedom (e.g., **D** command) at the same node. Overconstraints can lead to convergence difficulties and/or inaccurate results. The Lagrange multiplier method also introduces zero diagonal terms in the stiffness matrix, so that iterative solvers (e.g., PCG) can not be used.

The contact traction components (i.e., Lagrange multiplier parameters) become unknown DOFs for each element. The associated Newton-Raphson load vector is:

$$\{F_{nr}\} = \lfloor P, \tau_1, \tau_2, u_n, \Delta u_1, \Delta u_2 \rfloor^T \tag{13.299}$$

Lagrange Multiplier on Contact Normal and Penalty on Frictional Direction

In this method only the contact normal pressure is treated as a Lagrange multiplier. The tangential contact stresses are calculated based on the penalty method (see Equation 13.295 (p. 564)).

This method allows only a very small amount of slip for a sticking contact condition. It overcomes chattering problems due to contact status change between sliding and sticking which often occurs in the pure Lagrange Multiplier method. Therefore this algorithm treats frictional sliding contact problems much better than the pure Lagrange method.

13.174.5. Viscous Damping

Viscous damping is primarily used to damp relative motions between the contact and target surfaces for open contact. It provides a certain resistance to reduce the risk of rigid body motion. The viscous damping traction is proportional but opposite to the relative velocities along contact normal or tangential directions:

$$P_{dn} = d_n \dot{u}_n \tag{13.300}$$

$$P_{d1} = d_t \dot{u}_1 \tag{13.301}$$

$$P_{d2} = d_t \dot{u}_2 \tag{13.302}$$

where:

d_n = damping coefficient in normal direction
d_t = damping coefficient in tangential direction

The total damping force is calculated by the integration of damping traction over the contact elements.

The damping coefficient is specified as a function of the opening gap:

$$d_n = \begin{cases} d_{n0}(u_{pinb} - u_n)/u_{pinb} & \text{if } u_n \le u_{pinb} \\ 0 & \text{if } u_n > u_{pinb} \text{ or } u_n \le 0 \end{cases} \tag{13.303}$$

$$d_t = \begin{cases} d_{n0} d_{t0} (u_{pinb} - u_n)/u_{pinb} & \text{if } u_n \le u_{pinb} \\ 0 & \text{if } u_n > u_{pinb} \text{ or } u_n \le 0 \end{cases} \tag{13.304}$$

where:

u_{pinb} = pinball radius (real constant PINB)
d_{n0} = normal contact damping coefficient (real constant FDMN)
d_{t0} = tangential contact damping coefficient (real constant FDMT)

13.174.6. Energy and Momentum Conserving Contact

To correctly model the physical interaction between contact and target surfaces in a transient dynamic analysis, the contact forces must maintain force and energy balance, and ensure proper transfer of linear momentum. This requires imposing additional constraints on relative velocities between contact and target surfaces (see Laursen and Chawla ([376] (p. 942)), and Armero and Pet cz ([377] (p. 942))).

Impact Constraints and Contact Forces

In ANSYS the penetration constraints and the relative velocity constraints between contact and target surfaces are collectively referred to as impact constraints. These constraints can be selected by setting KEYOPT(7) = 4 for any of the 2D or 3D contact elements and are valid for all types of contact interactions (flexible-to-flexible, flexible-to-rigid, and rigid-to-rigid) with and without friction.

An automatic time stepping scheme is used to predict the time of impact and adjust the size of the time increment to minimize penetration. When contact is detected, the relative velocity constraints are imposed using one of the four contact algorithms: pure penalty method, augmented Lagrangian method, pure Lagrange multiplier method, or Lagrange multiplier in contact normal and penalty in frictional direction method. In the case of rough contact (KEYOPT(12) = 1) the relative velocity constraint is imposed in the tangential direction also to prevent slip. In the case of standard contact (KEYOPT(12) = 0) with friction, the slip increment and frictional stress are computed by taking the relative velocity constraint into consideration.

For the pure penalty method, contact pressure *P* and friction stresses τ_i for isotropic friction are defined as:

$$P = \begin{cases} 0 & \text{if } \bar{u}_n > 0 \text{ or } \bar{u}_n < u_n \le 0 \\ P^{n-1} + K_n (u_n - \bar{u}_n) & \text{if } u_n \le \bar{u}_n \le 0 \end{cases} \tag{13.305}$$

where:

K_n = contact normal stiffness

u_n = contact gap size

$\bar{u}_n$ = algorithmic contact gap size (based on the relative velocity constraint)

P^{n-1} = normal contact pressure at the end of previous substep

and:

$$\tau_i = \begin{cases} \tau_i^{n-1} + K_s (\Delta u_i - \Delta \bar{u}_i) & \text{if } \|\tau\| = \sqrt{\tau_1^2 + \tau_2^2} - \mu_{iso} P < 0 \text{ (sticking)} \\ \mu_{iso} P \dfrac{(\Delta u_i - \Delta \bar{u}_i)}{\|(\Delta u_i - \Delta \bar{u}_i)\|} & \text{if } \|\tau\| = \sqrt{\tau_1^2 + \tau_2^2} - \mu_{iso} P = 0 \text{ (sliding)} \end{cases} \tag{13.306}$$

where:

K_s = tangential contact stiffness (input as FKT on **R** command)

Δu_i = slip increment in direction i over the current substep

$\|\Delta u\|$ = equivalent slip increment over the current substep

$\Delta \bar{u}_i$ = algorithmic slip increment in direction i over the current substep

$\|\Delta \bar{u}\|$ = algorithmic equivalent slip increment over the current substep

μ_{iso} = coefficient of friction

τ_i^{n-1} = frictional stress in direction i = 1,2 at the end of previous substep

For other contact algorithms, the expressions for contact pressure and frictional stresses are defined in a similar manner as shown in Equation 13.298 (p. 565) and Equation 13.299 (p. 565) but with additional variables as shown above in Equation 13.305 (p. 567) and Equation 13.306 (p. 567).

Energy and Momentum Balance

Imposition of the impact constraints at Gauss points of contact elements ensures satisfaction of momentum and energy balance in a finite element sense. Since the impact constraints act only on the contact/target interface, energy balance is not enforced for the underlying finite elements used to model the interior of the contact and target bodies. Total energy at the contact/target interface is conserved for frictionless or rough contact when relative velocity constraints are satisfied exactly. If the relative velocity constraints are not satisfied to a tight tolerance there may be some loss of kinetic energy.

When friction is specified for contact elements, energy is conserved when the contact and target surfaces are not slipping (STICK) with respect to each other, and energy equal to the work done by frictional forces is dissipated when the contact and target surfaces are slipping (SLIP) with respect to each other.

Energy is also lost when numerical damping is used for the time integration scheme.

As per the classical theory of impact, exact conservation of energy during impact between rigid bodies is identified with *elastic impact*. It corresponds to a *coefficient of restitution* (COR) of 1. The impact constraints in ANSYS for impact between rigid bodies satisfy the conditions of elastic impact when the constraints are satisfied exactly and no numerical damping or friction is specified. Energy loss during impact between rigid bodies can be modeled by specifying a coefficient of restitution value (input as real constant COR) smaller than 1.

Time Integration Scheme

The impact constraints are formulated such that they can be used with both methods available for implicit transient dynamic analysis in ANSYS, the Newmark method and the HHT method. An important reason for using the impact constraints is that they make the time integration scheme numerically more stable without using large numerical damping. A small amount of numerical damping may still be needed to suppress high frequency noise.

13.174.7. Debonding

Debonding refers to separation of bonded contact (KEYOPT(12) = 2, 3, 4, 5 or 6). It is activated by associating a cohesive zone material model (input with **TB**,CZM) with contact elements. Debonding is available only for pure penalty method and augmented Lagrangian method (KEYOPT(2) = 0,1) with contact elements CONTA171, CONTA172, CONTA173, CONTA174, CONTA175, CONTA176, and CONTA177.

A cohesive zone material model is provided with bilinear behavior (Alfano and Crisfield([365] (p. 941))) for debonding. The model defines contact stresses as:

$$P = K_n u_n (1 - d) \tag{13.307}$$

$$\tau_1 = K_t u_1 (1 - d) \tag{13.308}$$

and

$$\tau_2 = K_t u_2 (1 - d) \tag{13.309}$$

where:

P = normal contact stress (tension)

τ_1 = tangential contact stress in direction 1
τ_2 = tangential contact stress in direction 2
K_n = normal contact stiffness
K_t = tangential contact stiffness
u_n = contact gap
u_1 = contact slip distance in direction 1
u_2 = contact slip distance in direction 2
d = debonding parameter
direction 1 and direction 2 = principal directions in tangent plane

The debonding parameter is defined as:

$$d = \left(\frac{\Delta - 1}{\Delta} \right) \chi \tag{13.310}$$

with d = 0 for $\Delta \leq 1$ and $0 < d \leq 1$ for $\Delta > 1$, and Δ and χ are defined below.

Debonding allows three modes of separation: mode I, mode II and mixed mode.

Mode I debonding is defined by setting

$$\Delta = \Delta_n = \frac{u_n}{\bar{u}_n} \tag{13.311}$$

and

$$\chi = \chi_n = \left(\frac{u_n^c}{u_n^c - \bar{u}_n} \right) \tag{13.312}$$

where:

$\bar{u}_n$ = contact gap at the maximum normal contact traction (tension)

u_n^c = contact gap at the completion of debonding (input on **TBDATA** command as *C2* using **TB**,CZM)

Mode II debonding is defined by setting

$$\Delta = \Delta_t = \frac{u_t}{\bar{u}_t} \tag{13.313}$$

$$u_t = \sqrt{u_1^2 + u_2^2} \tag{13.314}$$

and

$$\chi = \chi_t = \left(\frac{u_t^c}{u_t^c - \bar{u}_t} \right) \tag{13.315}$$

where:

$\bar{u}_t$ = equivalent tangential slip distance at the maximum equivalent tangential stress, $\sqrt{\tau_1^2 + \tau_2^2}$

u_t^c = equivalent tangential slip distance at the completion of debonding (input on **TBDATA** command as *C4* using **TB**,CZM)

Mixed mode debonding is defined by setting

$$\Delta_m = \sqrt{\Delta_n^2 + \Delta_t^2} \tag{13.316}$$

and

$$\chi = \left(\frac{u_n^c}{u_n^c - \bar{u}_n} \right) = \left(\frac{u_t^c}{u_t^c - \bar{u}_t} \right) \tag{13.317}$$

The constraint on χ that the ratio of the contact gap distances be same as the ratio of tangential slip distances is enforced automatically by appropriately scaling the contact stiffness values.

For mixed mode, debonding is complete when the energy criterion is satisfied:

$$\left(\frac{G_n}{G_{cn}} \right)^2 + \left(\frac{G_t}{G_{ct}} \right)^2 = 1 \tag{13.318}$$

with

$$G_n = \int P du_n \tag{13.319}$$

$$G_t = \int \sqrt{\tau_1^2 + \tau_2^2}\, du_t \tag{13.320}$$

$$G_{cn} = \frac{1}{2} \sigma_{max} u_n^c \tag{13.321}$$

$$G_{ct} = \frac{1}{2} \tau_{max} u_t^c \tag{13.322}$$

where:

σ_{max} = maximum normal contact stress (input on **TBDATA** command as *C1* using **TB**,CZM)
τ_{max} = maximum equivalent tangential contact stress (input on **TBDATA** command as *C3* using **TB**,CZM)

Verification of satisfaction of energy criterion can be done during postprocessing of results.

The debonding modes are based on input data:

1. Mode I for normal data (input on **TBDATA** command as *C1*, *C2*, and *C5*).
2. Mode II for tangential data (input on **TBDATA** command as *C3*, *C4*, and *C5*).
3. Mixed mode for normal and tangential data (input on **TBDATA** command as *C1*, *C2*, *C3*, *C4*, *C5* and *C6*).

Artificial damping can be used to overcome convergence difficulties associated with debonding. It is activated by specifying the damping coefficient η (input on **TBDATA** command as *C5* using **TB**,CZM).

Tangential slip under compressive normal contact stress for mixed mode debonding is controlled by appropriately setting the flag β (input on **TBDATA** command as *C6* using **TB**,CZM). Settings on β are:

β = 0 (default) indicates no tangential slip
β = 1 indicates tangential slip is allowed

After debonding is completed the surface interaction is governed by standard contact constraints for normal and tangential directions. Frictional contact is used if friction is specified for contact elements.

13.174.8. Contact Surface Wear

Progressive loss of material from the surface of a solid body when in contact with another body (in any phase) is called wear. At a micro-scale, the process of wear is quite complex with various mechanical and chemical processes resulting in material failure and loss. At a continuum scale, wear is approximated by phenomenological models that relate various quantities at the contact surface with material loss (see Meng and Ludema [418] (p. 944)). ANSYS models wear using such a phenomenological approach. Material loss due to wear is approximated by repositioning the contact nodes at the contact surface. The new coordinates of the nodes are determined by a wear model. Since the contact nodes are moved to new positions, the contact variables (for example, contact pressure) change. The underlying continuum elements also experience a loss in material (and volume), thus simulating wear. After the nodes are moved to a new position, equilibrium may be lost and the program continues to iterate until equilibrium is achieved.

The following options are available for defining a wear model:

- Archard wear model (see Archard [419] (p. 944))
- User-defined wear model

Wear is activated using the **TB** command with *Lab* = WEAR for the material assigned to contact elements. The following contact elements support modeling wear: CONTA171, CONTA172, CONTA173, CONTA174, and CONTA175.

Archard Wear Model

The Archard wear model (see Archard [419] (p. 944)) is a popular sliding wear model with fairly good results for simulating wear. The original model proposed by Archard assumed that the rate of volume loss due to wear is linearly proportional to the contact pressure and sliding velocity at the contact surface. A generalized version of this model allowing proper law dependence on contact pressure and velocity is implemented in the program. Wear is assumed to occur in the inward normal direction of the surface, taken as the direction opposite the contact normal direction. The rate of wear at a contact node, $\dot{w}$, is given by:

$$\dot{w} = \frac{K}{H} P^m v_{rel}^n \tag{13.323}$$

where:

K = wear coefficient
H = material hardness
P = contact pressure
v_{rel} = the relative sliding velocity
m = pressure exponent
n = velocity exponent

The Archard model is defined using the **TB** command with *TBOPT* = ARCD and *Lab* = WEAR.

User-Defined Wear Model (USERWEAR)

If the Archard wear model does not fit your needs, you can define your own wear model by programming the subroutine USERWEAR. Most of the relevant contact results and properties are passed in the USERWEAR subroutine. You can define the wear increment and direction of the wear increment. The user-defined wear model is defined using the **TB** command with *TB0PT* = USER and *Lab* = WEAR.

13.174.9. Thermal/Structural Contact

Combined structural and thermal contact is specified if KEYOPT(1) = 1, which indicates that structural and thermal DOFs are active. Pure thermal contact is specified if KEYOPT(1) = 2. The thermal contact features (Zhu and Cescotto([280] (p. 936))) are:

Thermal Contact Conduction

$$q = K_c(T_T - T_C) \qquad \text{if STAT} \geq 2 \tag{13.324}$$

where:

q = heat flux (heat flow rate per area)
K_c = thermal contact conductance coefficient (input as TCC on **R** command)
T_T = temperature on target surface
T_C = temperature on contact surface

Heat Convection

$$q = h_f(T_e - T_C) \qquad \text{if STAT} \leq 1 \tag{13.325}$$

where:

h_f = convection coefficient (input on **SFE** command with *Lab* = CONV and KVAL = 1)

$$T_e = \begin{cases} T_T & \text{if STAT = 1} \\ \text{environmental temperature (input on } \textbf{SFE} & \text{if STAT = 0} \\ \text{command with Lab = CONV and KVAL = 2)} & \end{cases}$$

Heat Radiation

$$q = \sigma\varepsilon F\left[(T_e + T_o)^4 - (T_C + T_o)^4\right] \qquad \text{if STAT} \leq 1 \tag{13.326}$$

where:

σ = Stefan-Boltzmann constant (input as SBCT on **R** command)
ε = emissivity (input using EMIS on **MP** command)
F = radiation view factor (input as RDVF on **R** command)
T_o = temperature offset (input as VALUE on **TOFFST** command)

Heat Generation Due to Frictional Sliding

$$\left.\begin{aligned} q_c &= F_w F_f t v \\ q_T &= (1 - F_w) F_f t v \end{aligned}\right\} \text{if STAT = 2 and } \mu > 0 \tag{13.327}$$

where:

q_c = amount of frictional dissipation on contact surface
q_T = amount of frictional dissipation on target surface
F_w = weight factor for the distribution of heat between two contact and target surfaces (input as FWGT on **R** command)
F_f = fractional dissipated energy converted into heat (input on FHTG on **R** command)
t = equivalent frictional stress
v = sliding rate

Note

When KEYOPT(1) = 2, heat generation due to friction is ignored.

13.174.10. Electric Contact

Combined structural, thermal, and electric contact is specified if KEYOPT(1) = 3. Combined thermal and electric contact is specified if KEYOPT(1) = 4. Combined structural and electric contact is specified if KEYOPT(1) = 5. Pure electric contact is specified if KEYOPT(1) = 6. The electric contact features are:

Electric Current Conduction (KEYOPT(1) = 3 or 4)

$$J = \frac{\sigma}{L}(V_T - V_C) \tag{13.328}$$

where:

J = current density
σ/L = electric conductivity per unit length (input as ECC on **R** command)
V_T = voltage on target surface
V_C = voltage on contact surface

Electrostatic (KEYOPT(1) = 5 or 6)

$$\frac{Q}{A} = \frac{C}{A}(V_T - V_C) \tag{13.329}$$

where:

$\frac{Q}{A}$ = charge per unit area

$\frac{C}{A}$ = capacitance per unit area (input as ECC on **R** command)

13.174.11. Magnetic Contact

The magnetic contact is specified if KEYOPT(1) = 7. Using the magnetic scalar potential approach, the 3-D magnetic flux across the contacting interface is defined by:

$$\psi^n = C_M(\phi_t - \phi_c) - \mu_o A H_g^n \tag{13.330}$$

where:

ψ^n = magnetic flux
φ_t = magnetic potential at target surface (MAG degree of freedom)
φ_c = magnetic potential at contact surface (MAG degree of freedom)
C_M = magnetic contact permeance coefficient
μ_o = free space permeability
A = contact area
H_g^n = normal component of the "guess" magnetic field (See Equation 5.16 (p. 180))

The gap permeance is defined as the ratio of the magnetic flux in the gap to the total magnetic potential difference across the gap. The equation for gap permeance is:

$$P = \mu_o A/t \tag{13.331}$$

where:

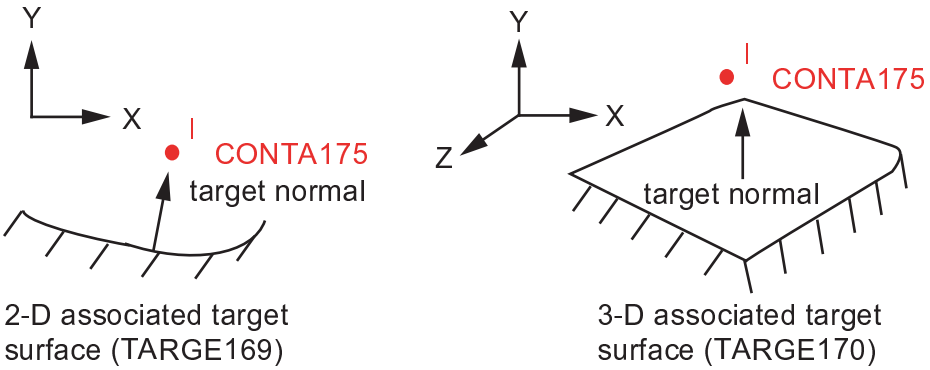
t = gap thickness

The magnetic contact permeance coefficient is defined as:

$$C_M = \mu_o/t \tag{13.332}$$

The above equations are only valid for 3-D analysis using the Magnetic Scalar Potential approach.

13.175. CONTA175 - 2-D/3-D Node-to-Surface Contact



2-D associated target surface (TARGE169)

3-D associated target surface (TARGE170)

Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Normal Direction	None	None
	Sliding Direction	None	None

13.175.1. Other Applicable Sections

The CONTA175 description is the same as for CONTA174 - 3-D 8-Node Surface-to-Surface Contact (p. 554) except that it is a one node contact element.

13.175.2. Contact Models

The contact model can be either contact force based (KEYOPT(3) = 0, default) or contact traction based (KEYOPT(3) = 1). For a contact traction based model, ANSYS can determine the area associated with the contact node. For the single point contact case, a unit area will be used which is equivalent to the contact force based model.

13.175.3. Contact Forces

In order to satisfy contact compatibility, forces are developed in a direction normal (n-direction) to the target that will tend to reduce the penetration to an acceptable numerical level. In addition to normal contact forces, friction forces are developed in directions that are tangent to the target plane.

$$F_n = \begin{cases} 0 & \text{if } u_n > 0 \\ K_n u_n & \text{if } u_n \le 0 \end{cases} \tag{13.333}$$

where:

F_n = normal contact force
K_n = contact normal stiffness (input as FKN on **R** command)
u_n = contact gap size

$$F_T = \begin{cases} K_T u_T & \text{if } |F_T| - \mu F_n < 0 \quad \text{(sticking)} \\ \mu K_n u_n & \text{if } |F_T| - \mu F_n = 0 \quad \text{(sliding)} \end{cases} \tag{13.334}$$

where:

F_T = tangential contact force
K_T = tangential contact stiffness (input as FKT on **R** command)
u_T = contact slip distance

$$\mu = \begin{cases} \mu_{iso}, \text{ coefficient of friction for isotropic friction (input as MU} \\ \text{using either } \textbf{TB} \text{ command with } Lab = \text{FRIC or } \textbf{MP} \text{ command)} \\ \mu_{eq}, \text{ equivalent coefficient of friction for orthotropic friction} \\ \text{(defined below)} \end{cases}$$

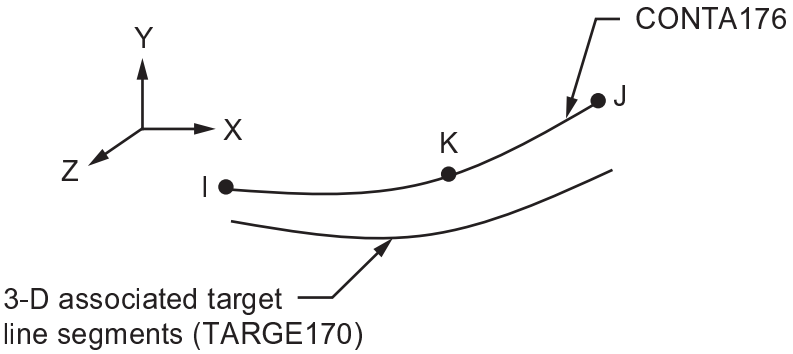
For orthotropic friction, μ_{eq} is computed using the expression:

$$\mu_{eq} = \sqrt{\frac{(\mu_1^2 + \mu_2^2)}{2}} \tag{13.335}$$

where:

μ_{eq} = equivalent coefficient of friction for orthotropic friction
μ_1, μ_2 = coefficients of friction in first and second principal directions (input as MU1 and MU2 using **TB** command with *Lab* = FRIC)

13.176. CONTA176 - 3-D Line-to-Line Contact



Matrix or Vector	Shape Function	Integration Points
Stiffness Matrix	$W = C_1 + C_2x + C_3x^2$	None

13.176.1. Other Applicable Sections

The CONTA176 description is the same as for CONTA174 - 3-D 8-Node Surface-to-Surface Contact (p. 554) except that it is a 3-D line contact element.

13.176.2. Contact Kinematics

Three different scenarios can be modeled by CONTA176:

- Internal contact where one beam (or pipe) slides inside another hollow beam (or pipe) (see Figure 13.34: Beam Sliding Inside a Hollow Beam (p. 577)).
- External contact between two beams that lie next to each other and are roughly parallel (see Figure 13.35: Parallel Beams in Contact (p. 577)).
- External contact between two beams that cross (see Figure 13.36: Crossing Beams in Contact (p. 578)).

Use KEYOPT(3) = 0 for the first two scenarios (internal contact and parallel beams). In both cases, the contact condition is only checked at contact nodes.

Use KEYOPT(3) = 1 for the third scenario (beams that cross). In this case, the contact condition is checked along the entire length of the beams. The beams with circular cross-sections are assumed to come in contact in a point-wise manner.

Figure 13.34: Beam Sliding Inside a Hollow Beam

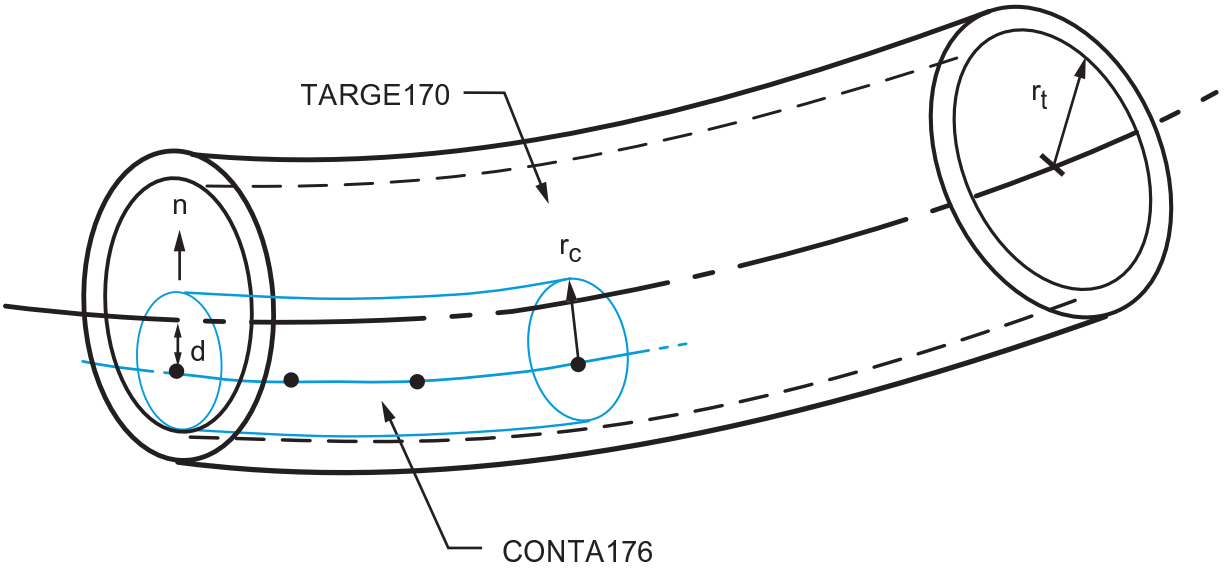


Figure 13.35: Parallel Beams in Contact

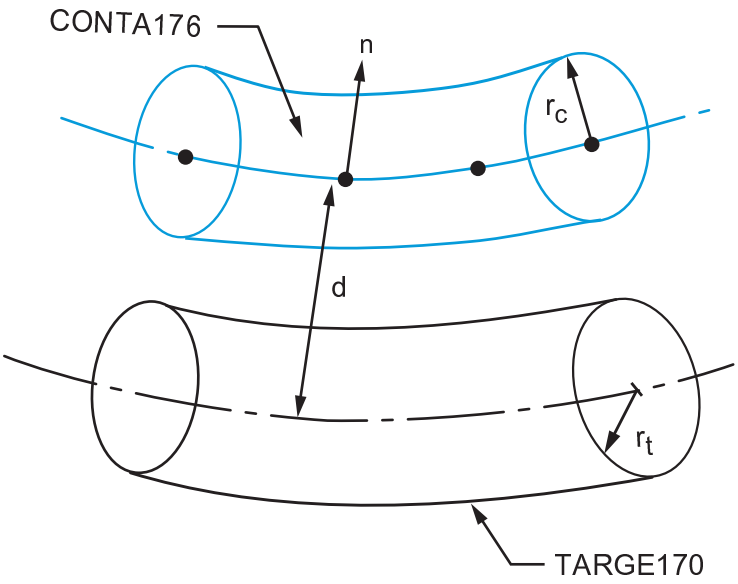
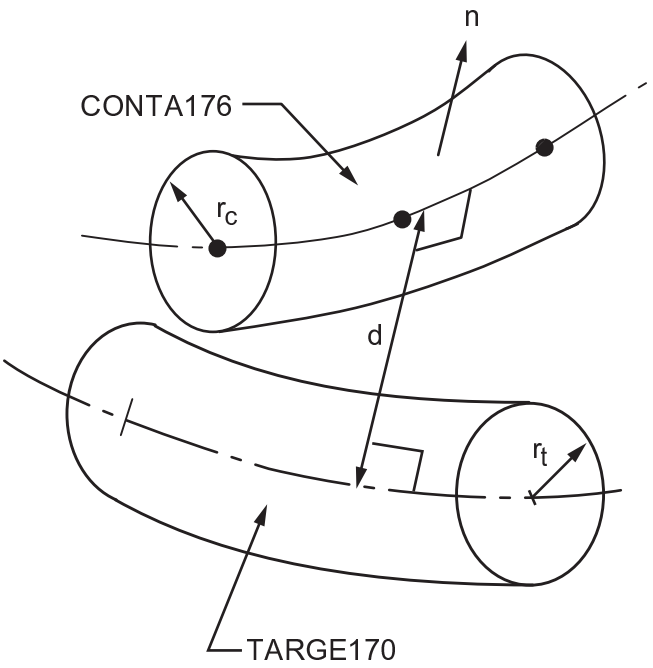


Figure 13.36: Crossing Beams in Contact



Contact is detected when two circular beams touch or overlap each other. The non-penetration condition for beams with a circular cross-section can be defined as follows.

For internal contact:

$$g = \left| r_t - r_c \right| - d \le 0 \tag{13.336}$$

and for external contact:

$$g = d - (r_c + r_t) \le 0 \tag{13.337}$$

where:

g = gap distance
r_c and r_t = radii of the cross-sections of the beam on the contact and target sides, respectively.
d = minimal distance between the two beam centerlines (also determines the contact normal direction).

Contact occurs for negative values of g.

13.176.3. Contact Forces

CONTA176 uses a contact force based model. In order to satisfy contact compatibility, forces are developed in a direction normal (n-direction) to the target that will tend to reduce the penetration to an acceptable numerical level. In addition to normal contact forces, friction forces are developed in directions that are tangent to the target plane.

$$F_n = \begin{cases} 0 & \text{if } u_n > 0 \\ K_n u_n & \text{if } u_n \le 0 \end{cases} \tag{13.338}$$

where:

F_n = normal contact force
K_n = contact normal stiffness (input as FKN on **R** command)
u_n = contact gap size

$$F_T = \begin{cases} K_T u_T & \text{if } |F_T| - \mu F_n < 0 \quad \text{(sticking)} \\ \mu K_n u_n & \text{if } |F_T| - \mu F_n = 0 \quad \text{(sliding)} \end{cases} \tag{13.339}$$

where:

F_T = tangential contact force
K_T = tangential contact stiffness (input as FKT on **R** command)
u_T = contact slip distance

$$\mu = \begin{cases} \mu_{iso}, \text{ coefficient of friction for isotropic friction (input as MU using either TB command with } Lab \text{ = FRIC or MP command)} \\ \mu_{eq}, \text{ equivalent coefficient of friction for orthotropic friction (defined below)} \end{cases}$$

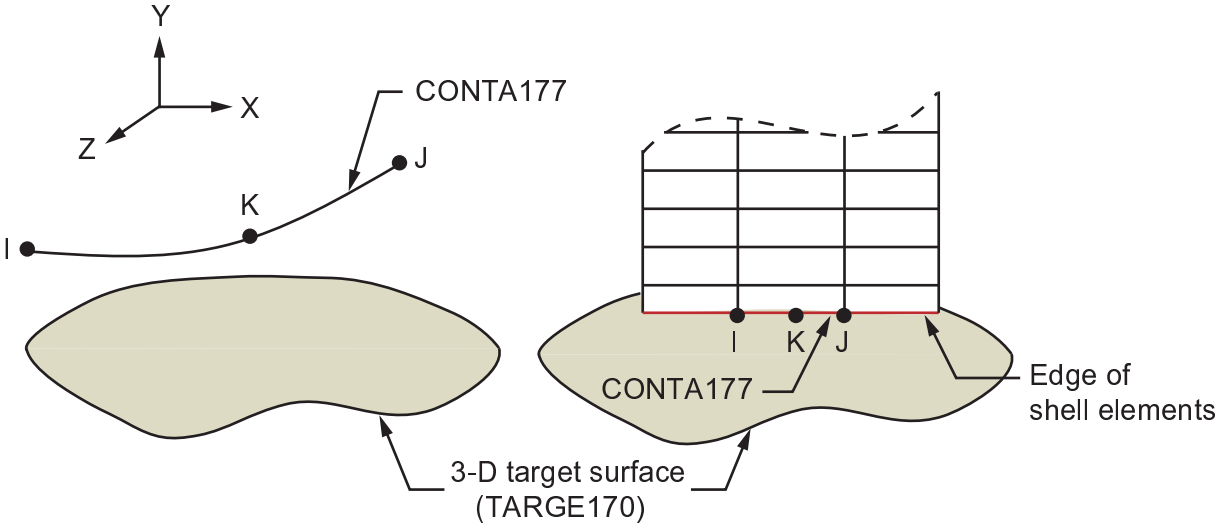
For orthotropic friction, μ_{eq} is computed using the expression:

$$\mu_{eq} = \sqrt{\frac{(\mu_1^2 + \mu_2^2)}{2}} \tag{13.340}$$

where:

μ_{eq} = equivalent coefficient of friction for orthotropic friction
μ_1, μ_2 = coefficients of friction in first and second principal directions (input as MU1 and MU2 using **TB** command with *Lab* = FRIC)

13.177. CONTA177 - 3-D Line-to-Surface Contact



Matrix or Vector	Shape Function	Integration Points
Stiffness Matrix	$W = C_1 + C_2x + C_3x^2$	None

13.177.1. Other Applicable Sections

The CONTA177 description is the same as for CONTA174 - 3-D 8-Node Surface-to-Surface Contact (p. 554) except that it is a 3-D line contact element.

13.177.2. Contact Forces

CONTA177 uses a contact force based model. In order to satisfy contact compatibility, forces are developed in a direction normal (n-direction) to the target that will tend to reduce the penetration to an acceptable numerical level. In addition to normal contact forces, friction forces are developed in directions that are tangent to the target plane.

$$F_n = \begin{cases} 0 & \text{if } u_n > 0 \\ K_n u_n & \text{if } u_n \le 0 \end{cases} \tag{13.341}$$

where:

F_n = normal contact force
K_n = contact normal stiffness (input as FKN on **R** command)
u_n = contact gap size

$$F_T = \begin{cases} K_T u_T & \text{if } |F_T| - \mu F_n < 0 \quad \text{(sticking)} \\ \mu K_n u_n & \text{if } |F_T| - \mu F_n = 0 \quad \text{(sliding)} \end{cases} \tag{13.342}$$

where:

F_T = tangential contact force
K_T = tangential contact stiffness (input as FKT on **R** command)
u_T = contact slip distance

$$\mu = \begin{cases} \mu_{iso}, \text{ coefficient of friction for isotropic friction (input as MU} \\ \text{using either TB command with } Lab \text{ = FRIC or MP command)} \\ \mu_{eq}, \text{ equivalent coefficient of friction for orthotropic friction} \\ \text{(defined below)} \end{cases}$$

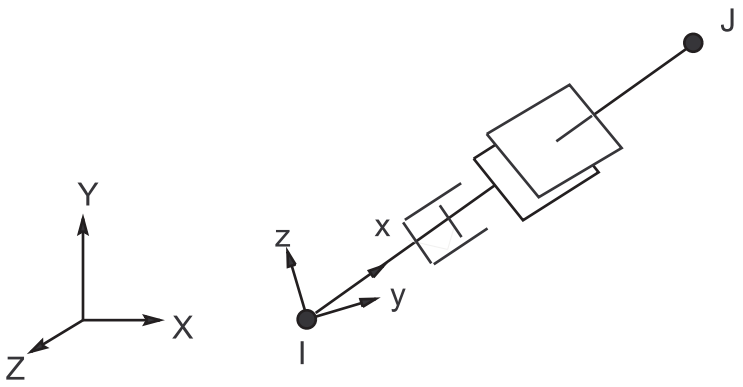
For orthotropic friction, μ_{eq} is computed using the expression:

$$\mu_{eq} = \sqrt{\frac{(\mu_1^2 + \mu_2^2)}{2}} \tag{13.343}$$

where:

μ_{eq} = equivalent coefficient of friction for orthotropic friction
μ_1, μ_2 = coefficients of friction in first and second principal directions (input as MU1 and MU2 using **TB** command with *Lab* = FRIC)

13.178. CONTA178 - 3-D Node-to-Node Contact



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Normal Direction	None	None
	Sliding Direction	None	None

Load Type	Distribution
Element Temperature	None - average used for material property evaluation
Nodal Temperature	None - average used for material property evaluation

13.178.1. Introduction

CONTA178 represents contact and sliding between any two nodes of any types of elements. This node-to-node contact element can handle cases when the contact location is known beforehand.

CONTA178 is applicable to 3-D geometries. It can also be used in 2-D and axisymmetric models by constraining the UZ degrees of freedom. The element is capable of supporting compression in the contact normal direction and Coulomb friction in the tangential direction.

13.178.2. Contact Algorithms

Four different contact algorithms are implemented in this element.

- Pure penalty method
- Augmented Lagrange method
- Pure Lagrange multiplier method
- Lagrange multiplier on contact normal penalty on frictional direction

Pure Penalty Method

The Newton-Raphson load vector is:

$$\{F_\ell^{nr}\} = \begin{Bmatrix} F_n \\ F_{sy} \\ F_{sz} \\ -F_n \\ -F_{sy} \\ -F_{sz} \end{Bmatrix} \tag{13.344}$$

where:

F_n = normal contact force
F_{sy} = tangential contact force in y direction
F_{sz} = tangential contact force in z direction

$$F_n = \begin{cases} 0 & \text{if } U_n > 0 \\ K_n U_n & \text{if } U_n \le 0 \end{cases} \tag{13.345}$$

where:

K_n = contact normal stiffness (input FKN on **R** command)
u_n = contact gap size

$$F_{sy} = \begin{cases} K_s u_y & \text{if } \sqrt{F_{sy}^2 + F_{sz}^2} - \mu F_n < 0 \text{ (sticking)} \\ \mu K_n u_n & \text{if } \sqrt{F_{sy}^2 + F_{sz}^2} - \mu F_n = 0 \text{ (sliding)} \end{cases} \tag{13.346}$$

where:

K_s = tangential contact stiffness (input as FKS on **R** command)
u_y = contact slip distance in y direction
μ = coefficient of friction (input as MU on **TB** command with *Lab* = FRIC or **MP** command)

Augmented Lagrange Method

$$F_n = \begin{cases} K_n u_n + \lambda_{i+1} & \text{if } u_n \le 0 \\ 0 & \text{if } u_n > 0 \end{cases} \tag{13.347}$$

where:

$$\lambda_{i+1} = \text{Lagrange multiplier force at iteration i+1} = \begin{cases} \tau_i + k_n u_n & \text{if } |u_n| > \varepsilon \\ \tau_i & \text{if } |u_n| \le \varepsilon \end{cases}$$

ε = user-defined compatibility tolerance (input as TOLN on **R** command)

The Lagrange multiplier component of force λ is computed locally (for each element) and iteratively.

Pure Lagrange Multiplier Method

The contact forces (i.e., Lagrange multiplier components of forces) become unknown DOFs for each element. The associated Newton-Raphson load vector is:

$$\{F^{nr}\} = \begin{Bmatrix} F_n \\ F_{sy} \\ F_{sz} \\ -F_n \\ -F_{sy} \\ -F_{sz} \\ u_n \\ u_y \\ u_z \end{Bmatrix} \tag{13.348}$$

Lagrange Multiplier on Contact Normal Penalty on Frictional Direction

In this method only the contact normal face is treated as a Lagrange multiplier. The tangential forces are calculated based on penalty method:

$$F_{sy} = \begin{cases} K_s u_y & \text{if } \sqrt{F_{sy}^2 + F_{sz}^2} - \mu F_n \le 0 \\ \mu F_n & \text{if } \sqrt{F_{sy}^2 + F_{sz}^2} - \mu F_n > 0 \end{cases} \tag{13.349}$$

13.178.3. Element Damper

The damping capability is only used for modal and transient analyses. Damping is only active in the contact normal direction when contact is closed. The damping force is computed as:

$$F_D = -C_v V \tag{13.350}$$

where:

V = relative velocity between two contact nodes in contact normal direction

$C_v = C_{v1} + C_{v2} V$

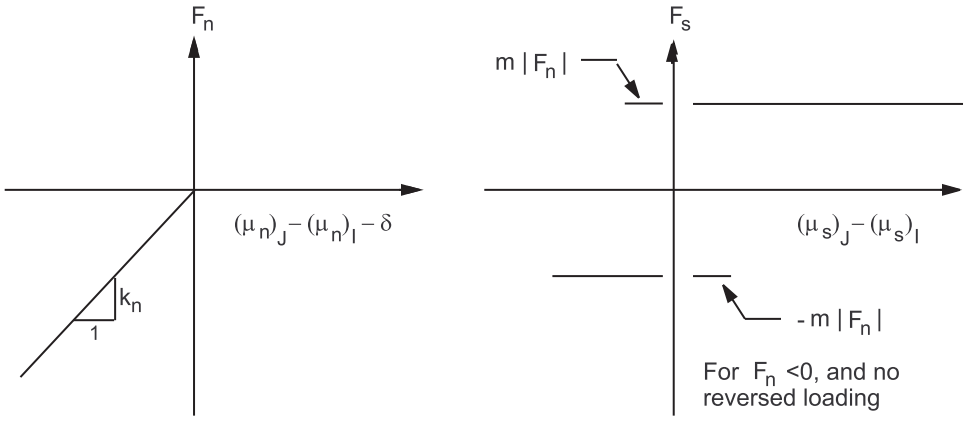
C_{v1} = constant damping coefficient (input as CV1 on **R** command)

C_{v2} = linear damping coefficient (input as CV2 on **R** command)

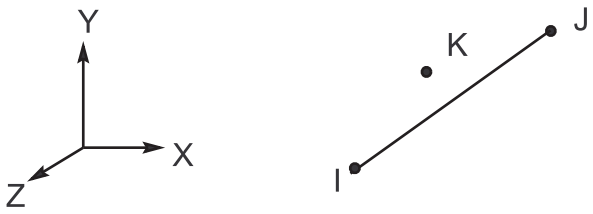
13.178.4. Rigid Coulomb Friction

If you know that a CONTA178 element will be in sliding status throughout the analysis, and that the relative displacement of the two nodes will be monotonically increasing, the rigid Coulomb friction option (KEYOPT(10) = 7) can be used to avoid convergence problems. This option removes the stiffness in the sliding direction, as shown in Figure 13.37: Force-Deflection Relations for Rigid Coulomb Option (p. 584). Note that if the relative displacement does not increase monotonically, the convergence characteristics of the rigid Coulomb friction law (KEYOPT(10) = 7) will be worse than for the elastic Coulomb friction law (KEYOPT(10) = 0).

Figure 13.37: Force-Deflection Relations for Rigid Coulomb Option



13.179. PRETS179 - Pretension



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None	None

Load Type	Distribution
Pretension Force	Applied on pretension node K across entire pretension section

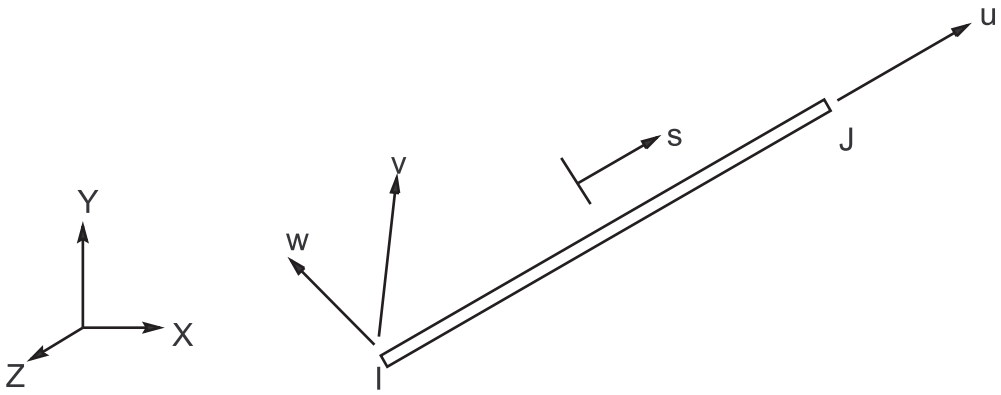
13.179.1. Introduction

The element is used to represent a two or three dimensional section for a bolted structure. The pretension section can carry a pretension load. The pretension node (K) on each section is used to control and monitor the total tension load.

13.179.2. Assumptions and Restrictions

The pretension element is not capable of carrying bending or torsion loads.

13.180. LINK180 - 3-D Spar (or Truss)



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix; and Thermal and Newton Raphson Load Vectors	Equation 11.6 (p. 329)	1
Mass and Stress Stiffening Matrices	Equation 11.6 (p. 329), Equation 11.7 (p. 330), and Equation 11.8 (p. 330)	1

Load Type	Distribution
Element Temperature	Linear along length
Nodal Temperature	Linear along length

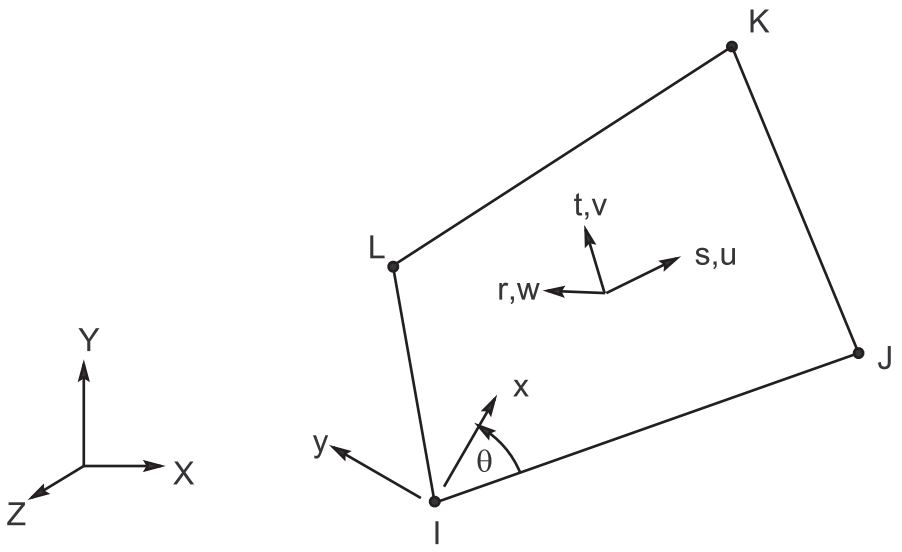
Reference: Cook et al.([117] (p. 927))

13.180.1. Assumptions and Restrictions

The theory for this element is a reduction of the theory for BEAM189 - 3-D 3-Node Beam (p. 602). The reductions include only 2 nodes, no bending or shear effects, no pressures, and the entire element as only one integration point.

The element is not capable of carrying bending loads. The stress is assumed to be uniform over the entire element.

13.181. SHELL181 - 4-Node Shell



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix; and Thermal Load Vector	Equation 11.69 (p. 337), Equation 11.70 (p. 337), Equation 11.71 (p. 337), Equation 11.72 (p. 337), Equation 11.73 (p. 337), and Equation 11.74 (p. 337)	In-plane: 1 x 1 (KEYOPT(3) = 0) 2 x 2 (KEYOPT(3) = 2) Thru-the-thickness: 1, 3, 5, 7, or 9 per layer for data input for section general shell option (KEYOPT(1) = 0) Thru-the-thickness: 1 per layer for section data input for membrane shell option (KEYOPT(1) = 1)
Consistent Mass and Stress Stiffness Matrices	Equation 11.69 (p. 337), Equation 11.70 (p. 337), Equation 11.71 (p. 337), Equation 11.72 (p. 337), Equation 11.73 (p. 337), and Equation 11.74 (p. 337)	Closed form integration
Lumped Mass Matrix	Equation 11.69 (p. 337), Equation 11.70 (p. 337), Equation 11.71 (p. 337)	Closed form integration
Transverse Pressure Load Vector	Equation 11.71 (p. 337)	2 x 2
Edge Pressure Load Vector	Equation 11.69 (p. 337) and Equation 11.70 (p. 337) specialized to the edge	2

Load Type	Distribution
Element Temperature	Bilinear in plane of element, linear thru each layer
Nodal Temperature	Bilinear in plane of element, constant thru thickness
Pressure	Bilinear in plane of element and linear along each edge

References: Ahmad([1] (p. 921)), Cook([5] (p. 921)), Dvorkin([96] (p. 926)), Dvorkin([97] (p. 926)), Bathe and Dvorkin([98] (p. 926)), Allman([113] (p. 927)), Cook([114] (p. 927)), MacNeal and Harder([115] (p. 927))

13.181.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations.

13.181.2. Assumptions and Restrictions

Normals to the centerplane are assumed to remain straight after deformation, but not necessarily normal to the centerplane.

Each set of integration points thru a layer (in the r direction) is assumed to have the same element (material) orientation.

13.181.3. Assumed Displacement Shape Functions

The assumed displacement and transverse shear strain shape functions are given in Shape Functions (p. 327). The basic functions for the transverse shear strain have been changed to avoid shear locking (Dvorkin([96] (p. 926)), Dvorkin([97] (p. 926)), Bathe and Dvorkin([98] (p. 926))).

13.181.4. Membrane Option

A membrane option is available for SHELL181 if KEYOPT(1) = 1. For this option, there is no bending stiffness or rotational degrees of freedom. There is only one integration point per layer, regardless of other input.

13.181.5. Warping

A warping factor is computed as:

$$\phi = \frac{D}{t} \tag{13.351}$$

where:

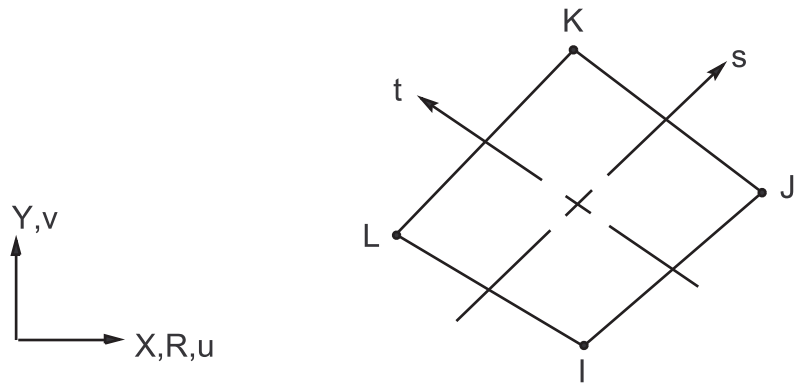
D = component of the vector from the first node to the fourth node parallel to the element normal
t = average thickness of the element

If ϕ > 1.0, a warning message is printed.

13.181.6. Shear Correction

The element uses an equivalent energy method to compute shear-correction factors. These factors are predetermined based on the section lay-up at the start of solution.

13.182. PLANE182 - 2-D 4-Node Structural Solid



Matrix or Vector	Geo-metry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Quad	Equation 11.120 (p. 341) and Equation 11.121 (p. 341)	2 x 2 if KEYOPT(1) = 0, 2, or 3 1 if KEYOPT(1) = 1
	Triangle	Equation 11.100 (p. 340) and Equation 11.101 (p. 340)	1
Mass Matrix	Quad	Same as stiffness matrix	2 x 2
	Triangle		1
Pressure Load Vector	Same as stiffness matrix, specialized to face		2

Load Type	Distribution
Element Temperature	Bilinear across element, constant thru thickness or around circum-ference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

13.182.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element.

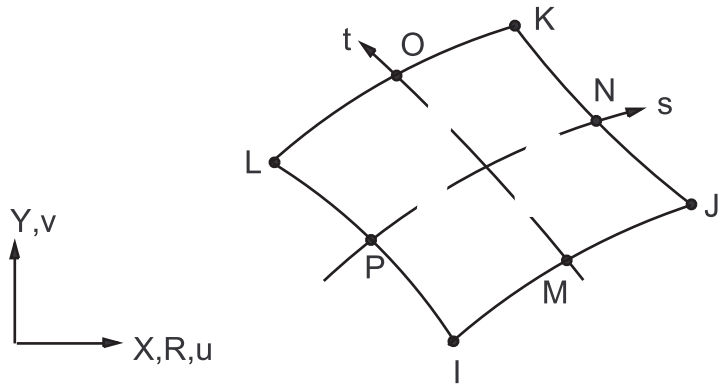
13.182.2. Theory

If KEYOPT(1) = 0, this element uses $\bar{B}$ method (selective reduced integration technique for volumetric terms) (Hughes([219] (p. 933)), Nagtegaal et al.([220] (p. 933))).

If KEYOPT(1) = 1, the uniform reduced integration technique (Flanagan and Belytschko([232] (p. 934))) is used.

If KEYOPT(1) = 2 or 3, the enhanced strain formulations from the work of Simo and Rifai([318] (p. 938)), Simo and Armero([319] (p. 939)), Simo et al.([320] (p. 939)), Andelfinger and Ramm([321] (p. 939)), and Nagtegaal and Fox([322] (p. 939)) are used. It introduces 5 internal degrees of freedom to prevent shear and volumetric locking for KEYOPT(1) = 2, and 4 internal degrees of freedom to prevent shear locking for KEYOPT(1) = 3. If mixed u-P formulation is employed with the enhanced strain formulations, only 4 degrees of freedom for overcoming shear locking are activated.

13.183. PLANE183 - 2-D 8-Node Structural Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Quad	Equation 11.134 (p. 342) and Equation 11.135 (p. 342)	2 x 2
	Triangle	Equation 11.112 (p. 340) and Equation 11.113 (p. 340)	3
Mass Matrix	Quad	Same as stiffness matrix	3 x 3
	Triangle		3
Pressure Load Vector	Same as stiffness matrix, specialized to the face		2 along face

Load Type	Distribution
Element Temperature	Same as shape functions across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

Reference: Zienkiewicz([39] (p. 922))

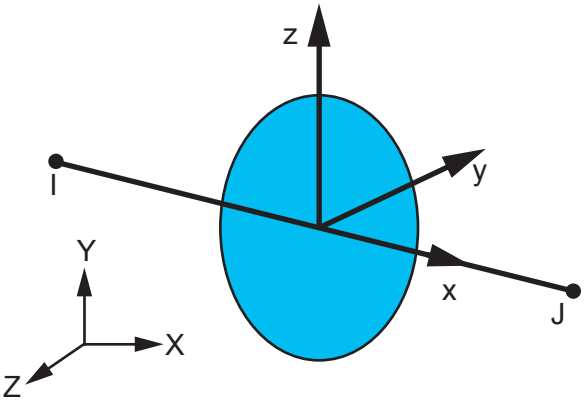
13.183.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element.

13.183.2. Assumptions and Restrictions

A dropped midside node implies that the face is and remains straight.

13.184. MPC184 - Multipoint Constraint

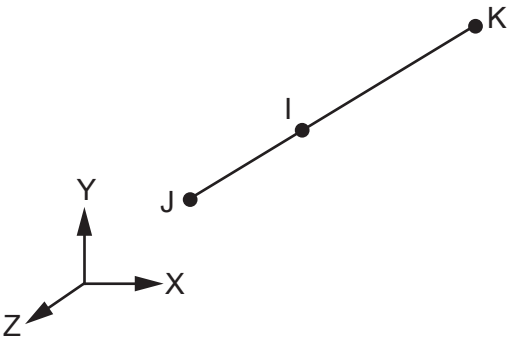


MPC184 comprises a general class of multipoint constraint elements that implement kinematic constraints using Lagrange multipliers. The elements are loosely classified here as "constraint elements" and "joint elements". All of these elements are used in situations that require you to impose some kind of constraint to meet certain requirements. Since these elements are implemented using Lagrange multipliers, the constraint forces and moments are available for output purposes. The different constraint elements and joint elements are identified by KEYOPT(1).

13.184.1. Slider Element

The slider element (KEYOPT(1) = 3) is a 3-node element that allows a "slave" node to slide on a line joining two "master" nodes.

Figure 13.38: 184.2 Slider Constraint Geometry



The constraints required to maintain the "slave" node on the line joining the two "master" nodes are as follows:

Define a unit vector **n** as:

$$\mathbf{n} = \frac{\mathbf{x}^J - \mathbf{x}^I}{\left| x^J - x^I \right|} \tag{13.352}$$

where:

$\mathbf{x}^I$, $\mathbf{x}^J$ = position vectors of nodes I and J in the current configuration

Identify unit vectors **l** and **m** such that **l**, **m**, and **n** form an orthonormal set.

The constraints are then defined as:

$$(x^K - x^I) \cdot L = 0 \tag{13.353}$$

$$(x^K - x^I) \cdot M = 0 \tag{13.354}$$

where:

x^k = position vector of the node K in the current configuration

Let i, j, and k be the global base vectors. Then we can define the unit vector **l** as:

$$l = \frac{n \times i}{|n \times i|} \qquad \text{if } n \neq i \tag{13.355}$$

If n = **l**, then:

$$l = \frac{n \times k}{|n \times k|} \tag{13.356}$$

Finally, the unit vector **m** is defined as:

$$m = n \times l \tag{13.357}$$

The virtual work contributions are obtained from taking the variations of the above equations.

13.184.2. Joint Elements

The equations for the constraints imposed in joint elements are described in the individual element descriptions:

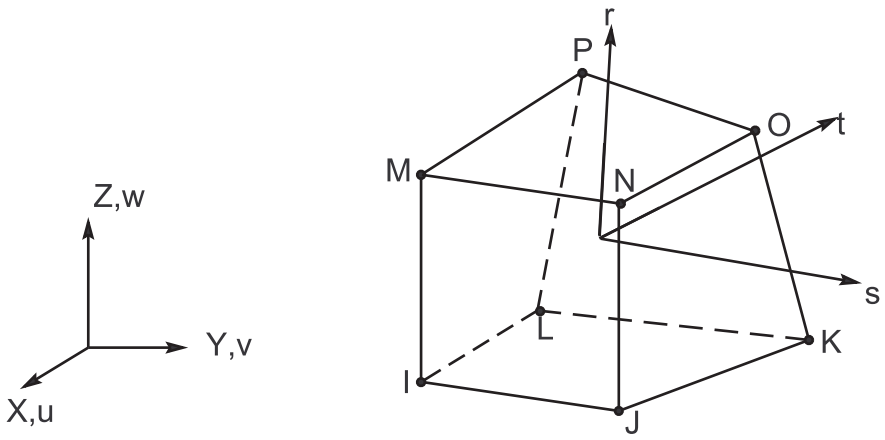
MPC184-Revolute
MPC184-Universal
MPC184-Slot
MPC184-Point
MPC184-Translational
MPC184-Cylindrical
MPC184-Planar
MPC184-Weld
MPC184-Orient
MPC184-Spherical
MPC184-General
MPC184-Screw

13.185. SOLID185 - 3-D 8-Node Structural Solid

SOLID185 is available in two forms:

- Standard (nonlayered) structural solid (KEYOPT(3) = 0, the default) - see SOLID185 - 3-D 8-Node Structural Solid (p. 592).
- Layered structural solid (KEYOPT(3) = 1) - see SOLID185 - 3-D 8-Node Layered Solid (p. 593).

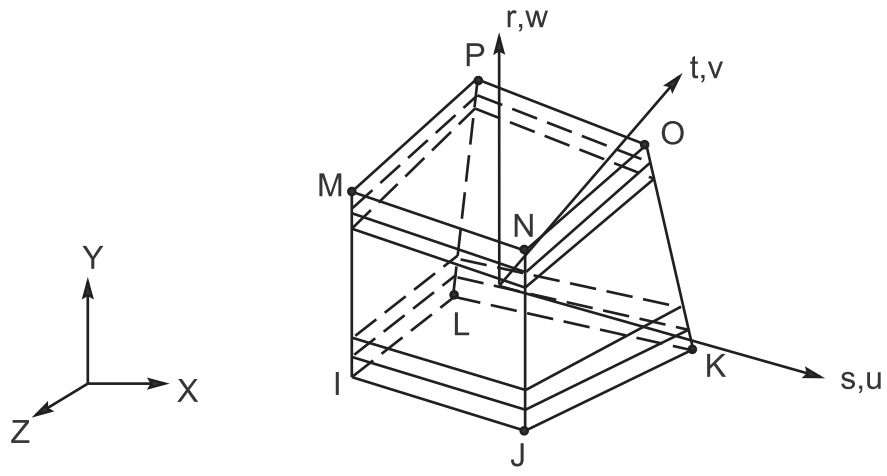
13.185.1. SOLID185 - 3-D 8-Node Structural Solid



Matrix or Vector	Shape Functions		Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Equation 11.212 (p. 353), Equation 11.213 (p. 353), and Equation 11.214 (p. 353)		2 x 2 x 2 if KEYOPT(2) = 0, 2, or 3 1 if KEYOPT(2) = 1
Mass Matrix	Same as stiffness matrix		2 x 2 x 2
Pressure Load Vector	Quad	Equation 11.69 (p. 337) and Equation 11.70 (p. 337)	2 x 2
	Triangle	Equation 11.49 (p. 336) and Equation 11.50 (p. 336)	3

Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

13.185.2. SOLID185 - 3-D 8-Node Layered Solid



Matrix or Vector	Shape Functions		Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Equation 11.212 (p. 353), Equation 11.213 (p. 353), and Equation 11.214 (p. 353)		In-plane: 2 x 2 Thru-the-thickness: 2 if no shell section defined. 1, 3, 5, 7, or 9 per layer if a shell section is defined
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Quad	Equation 11.69 (p. 337) and Equation 11.70 (p. 337)	2 x 2
	Triangle	Equation 11.49 (p. 336) and Equation 11.50 (p. 336)	3

Load Type	Distribution
Element Temperature	Bilinear in plane of element, linear thru each layer
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

13.185.3. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element.

13.185.4. Theory

If KEYOPT(2) = 0 (not applicable to layered SOLID185), this element uses $\bar{B}$ method (selective reduced integration technique for volumetric terms) (Hughes([219] (p. 933)), Nagtegaal et al.([220] (p. 933))).

If KEYOPT(2) = 1 (not applicable to layered SOLID185), the uniform reduced integration technique (Flanagan and Belytschko([232] (p. 934))) is used.

If KEYOPT(2) = 2 or 3, the enhanced strain formulations from the work of Simo and Rifai([318] (p. 938)), Simo and Armero([319] (p. 939)), Simo et al.([320] (p. 939)), Andelfinger and Ramm([321] (p. 939)), and Nagtegaal and Fox([322] (p. 939)) are used. It introduces 13 internal degrees of freedom to prevent shear and volumetric locking for KEYOPT(2) = 2, and 9 degrees of freedom to prevent shear locking only for KEYOPT(2) = 3. If mixed u-P formulation is employed with the enhanced strain formulations, only 9 degrees of freedom for overcoming shear locking are activated.

13.185.5. Shear Correction

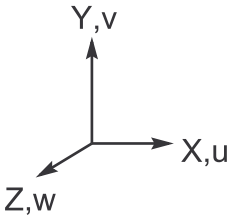
The element does not perform interlaminar shear correction. Stresses are calculated from strains as is.

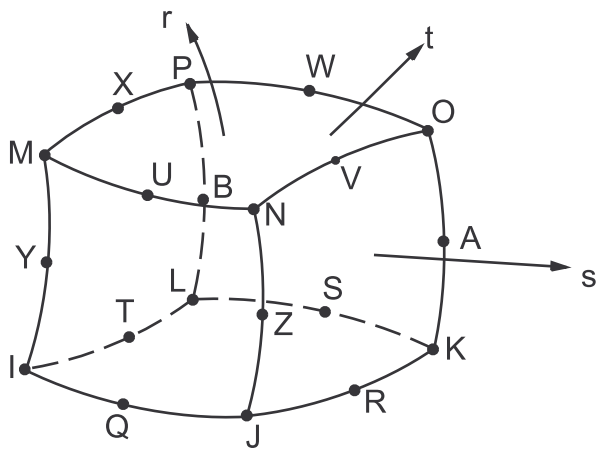
13.186. SOLID186 - 3-D 20-Node homogeneous/Layered Structural Solid

SOLID186 is available in two forms:

- homogeneous (nonlayered) structural solid (KEYOPT(3) = 0, the default) - see SOLID186 - 3-D 20-Node homogeneous Structural Solid (p. 594).
- Layered structural solid (KEYOPT(3) = 1) - see SOLID186 - 3-D 20-Node Layered Structural Solid (p. 595).

13.186.1. SOLID186 - 3-D 20-Node homogeneous Structural Solid



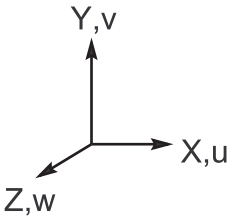


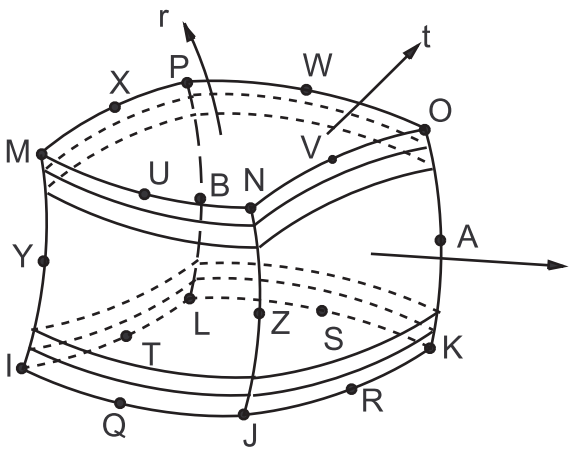
Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Brick	Equation 11.230 (p. 354), Equation 11.231 (p. 354), and Equation 11.232 (p. 355)	14 if KEYOPT(2) = 1 2 x 2 x 2 if KEYOPT(2) = 0
	Wedge	Equation 11.206 (p. 352), Equation 11.207 (p. 352), and Equation 11.208 (p. 352)	3 x 3
	Pyramid	Equation 11.190 (p. 350), Equation 11.191 (p. 350), and Equation 11.192 (p. 350)	2 x 2 x 2

Matrix or Vector	Geo-metry	Shape Functions	Integration Points
	Tet	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348)	4
Mass Matrix	Same as stiffness matrix.		3 x 3 x 3 if brick. If other shapes, same as stiffness matrix
Pressure Load Vector	Quad	Equation 11.84 (p. 338) and Equation 11.85 (p. 338)	3 x 3
	Triangle	Equation 11.57 (p. 336) and Equation 11.58 (p. 336)	6

Load Type	Distribution
Element Temperature	Same as shape functions thru element
Nodal Temperature	Same as shape functions thru element
Pressure	Bilinear across each face

13.186.2. SOLID186 - 3-D 20-Node Layered Structural Solid





Matrix or Vector	Geo-metry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Brick	Equation 11.230 (p. 354), Equation 11.231 (p. 354), and Equation 11.232 (p. 355)	In-plane: 2 x 2 Thru-the-thickness: 2 if no shell section defined. 1, 3, 5, 7, or 9 per layer if a shell section is defined

Matrix or Vector	Geo-metry	Shape Functions	Integration Points
	Wedge	Equation 11.206 (p. 352), Equation 11.207 (p. 352), and Equation 11.208 (p. 352)	In-plane: 3 Thru-the-thickness: 2 if no shell section defined. 1, 3, 5, 7, or 9 per layer if a shell section is defined
Mass Matrix	Same as stiffness matrix.		In-plane: 3 x 3 if brick 3 if wedge Thru-the-thickness: Same as stiffness matrix
Pressure Load Vector	Quad	Equation 11.84 (p. 338) and Equation 11.85 (p. 338)	3 x 3
	Triangle	Equation 11.57 (p. 336) and Equation 11.58 (p. 336)	6

Load Type	Distribution
Element Temperature	Bilinear in plane of element, linear thru each layer
Nodal Temperature	Same as shape functions thru element
Pressure	Bilinear across each face

Reference: Zienkiewicz([39] (p. 922))

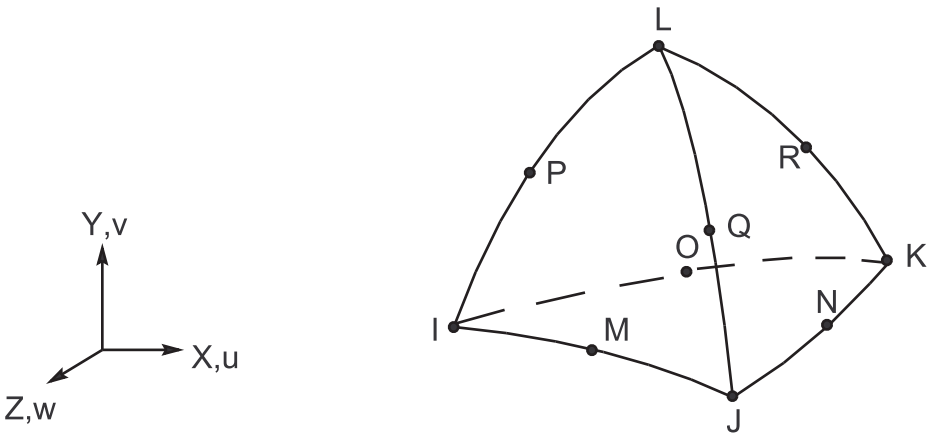
13.186.3. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element.

13.186.4. Shear Correction

The element does not perform interlaminar shear correction. Stresses are calculated from strains as is.

13.187. SOLID187 - 3-D 10-Node Tetrahedral Structural Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness, Mass, and Stress Stiffness Matrices; and Thermal Load Vector	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348)	4
Pressure Load Vector	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348) specialized to the face	6

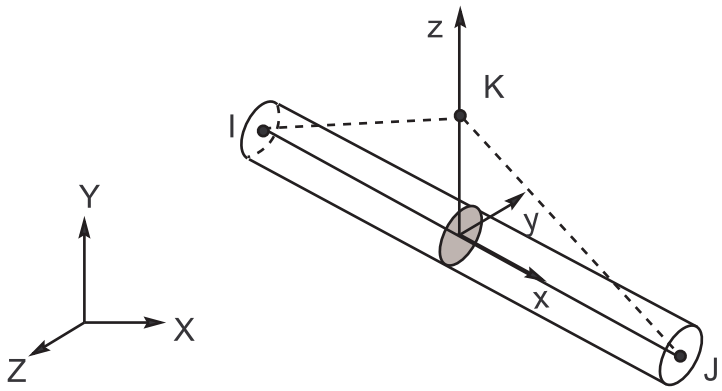
Load Type	Distribution
Element Temperature	Same as shape functions
Nodal Temperature	Same as shape functions
Pressure	Linear over each face

Reference: Zienkiewicz([39] (p. 922))

13.187.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element.

13.188. BEAM188 - 3-D 2-Node Beam



Matrix or Vector	Option	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal and Newton-Raphson Load Vectors	Linear (KEYOPT(3) = 0)	Equation 11.6 (p. 329), Equation 11.7 (p. 330), Equation 11.8 (p. 330), Equation 11.9 (p. 330), Equation 11.10 (p. 330), and Equation 11.11 (p. 330)	Along the length: 1 Across the section: see text below
	Quadratic (KEYOPT(3) = 2)	Equation 11.19 (p. 330), Equation 11.20 (p. 330), Equation 11.21 (p. 330), Equation 11.22 (p. 331), Equation 11.23 (p. 331), and Equation 11.24 (p. 331)	Along the length: 2 Across the section: see text below.
	Cubic (KEYOPT(3) = 3)	Equation 11.26 (p. 331), Equation 11.27 (p. 331), Equation 11.28 (p. 331), Equation 11.29 (p. 331), Equa-	Along the length: 3 Across the section: see text below.

Matrix or Vector	Option	Shape Functions	Integration Points
		tion 11.30 (p. 331), and Equation 11.31 (p. 331)	
Consistent Mass Matrix and Pressure Load Vector	Linear (KEYOPT(3) = 0)	Equation 11.6 (p. 329), Equation 11.7 (p. 330), Equation 11.8 (p. 330), Equation 11.9 (p. 330), Equation 11.10 (p. 330), and Equation 11.11 (p. 330)	Along the length: 2 Across the section: 1
	Quadratic (KEYOPT(3) = 2)	Equation 11.19 (p. 330), Equation 11.20 (p. 330), Equation 11.21 (p. 330), Equation 11.22 (p. 331), Equation 11.23 (p. 331), and Equation 11.24 (p. 331)	Along the length: 3 Across the section: 1
	Cubic (KEYOPT(3) = 3)	Equation 11.26 (p. 331), Equation 11.27 (p. 331), Equation 11.28 (p. 331), Equation 11.29 (p. 331), Equation 11.30 (p. 331), and Equation 11.31 (p. 331)	Along the length: 4 Across the section: 1
Lumped Mass Matrix	Linear (KEYOPT(3) = 0)	Equation 11.6 (p. 329), Equation 11.7 (p. 330), and Equation 11.8 (p. 330)	Along the length: 2 Across the section: 1
	Quadratic (KEYOPT(3) = 2)	Equation 11.19 (p. 330), Equation 11.20 (p. 330),	Along the length: 3 Across the section: 1

Matrix or Vector	Option	Shape Functions	Integration Points
		and Equation 11.21 (p. 330)	
	Cubic (KEYOPT(3) = 3)	Equation 11.26 (p. 331), Equation 11.27 (p. 331), and Equation 11.28 (p. 331)	Along the length: 4 Across the section: 1

Load Type	Distribution
Element Temperature	Bilinear across cross-section and linear along length
Nodal Temperature	Constant across cross-section, linear along length
Pressure	Linear along length. The pressure is assumed to act along the element x-axis.

References: Simo and Vu-Quoc([237] (p. 934)), Ibrahimbegovic([238] (p. 934)).

13.188.1. Assumptions and Restrictions

The element is based on Timoshenko beam theory; therefore, shear deformation effects are included. The element is well-suited for linear, large rotation, and/or large strain nonlinear applications. If KEYOPT(2) = 0, the cross-sectional dimensions are scaled uniformly as a function of axial strain in nonlinear analysis such that the volume of the element is preserved.

The element includes stress stiffness terms, by default, in any analysis using large deformation (**NLGEOM**,ON). The stress stiffness terms provided enable the elements to analyze flexural, lateral and torsional stability problems (using eigenvalue buckling or collapse studies with arc length methods). Pressure load stiffness (Pressure Load Stiffness (p. 46)) is included.

Transverse-shear strain is constant through cross-section; that is, cross sections remain plane and undistorted after deformation. Higher-order theories are not used to account for variation in distribution of shear stresses. A shear-correction factor is calculated in accordance with in the following references:

- Schramm, U., L. Kitis, W. Kang, and W.D. Pilkey. "On the Shear Deformation Coefficient in Beam Theory." [Finite Elements in Analysis and Design, The International Journal of Applied Finite Elements and Computer Aided Engineering]. 16 (1994): 141-162.

- Pilkey, Walter D. [Formulas for Stress, Strain, and Structural Matrices]. New Jersey: Wiley, 1994.

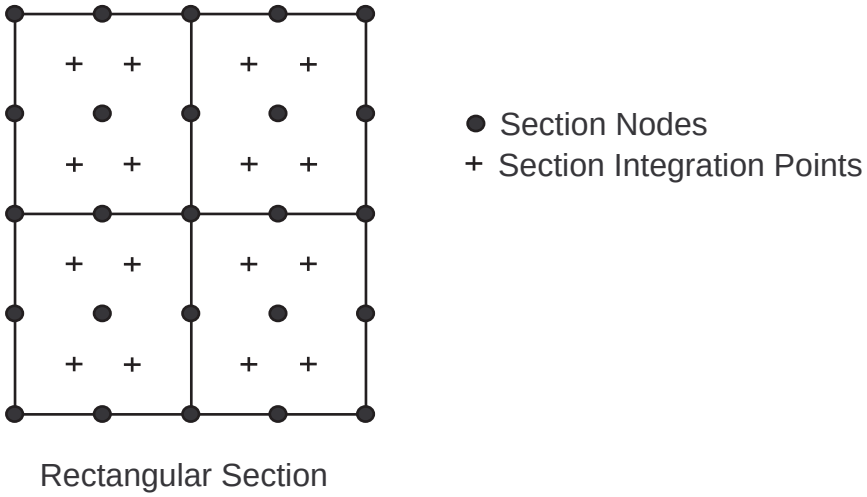
The element can be used for slender or stout beams. Due to the limitations of first order shear deformation theory, only moderately "thick" beams may be analyzed. Slenderness ratio of a beam structure may be used in judging the applicability of the element. It is important to note that this ratio should be calculated using some global distance measures, and not based on individual element dimensions. A slenderness ratio greater than 30 is recommended.

These elements support only elastic relationships between transverse-shear forces and transverse-shear strains. Orthotropic elastic material properties with bilinear and multilinear isotropic hardening plasticity options (BISO, MISO) may be used. Transverse-shear stiffnesses can be specified using real constants.

The St. Venant warping functions for torsional behavior is determined in the undeformed state, and is used to define shear strain even after yielding. The element does not provide options to recalculate the torsional shear distribution on cross sections during the analysis and possible partial plastic yielding of cross section. As such, large inelastic deformation due to torsional loading should be treated with caution and carefully verified.

The elements are provided with section relevant quantities (area of integration, position, Poisson function, function derivatives, etc.) automatically at a number of section points by the use of section commands. Each section is assumed to be an assembly of predetermined number of nine-node cells which illustrates a section model of a rectangular section. Each cell has four integration points.

Figure 13.39: Section Model



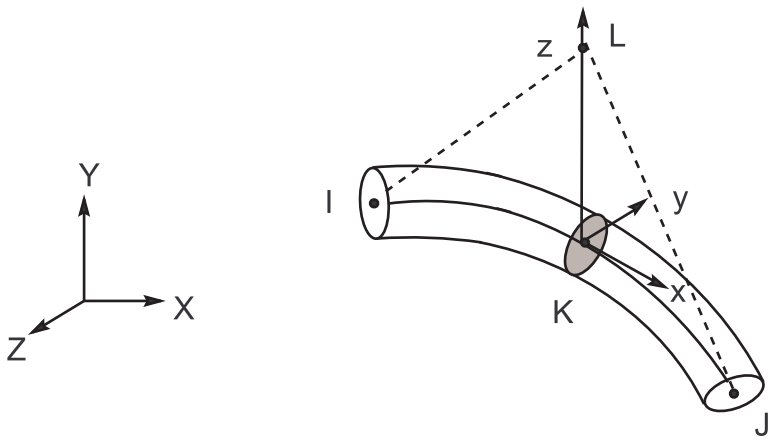
When the material has inelastic behavior or the temperature varies across the section, constitutive calculations are performed at each of the section integration points. For all other cases, the element uses the precalculated properties of the section at each element integration point along the length. The restrained warping formulation used may be found in Timoshenko and Gere([246] (p. 934)) and Schulz and Fillippou([247] (p. 934)).

13.188.2. Stress Evaluation

Several stress-evaluation options exist. The section strains and generalized stresses are evaluated at element integration points and then linearly extrapolated to the nodes of the element.

If the material is elastic, stresses and strains are available after extrapolation in cross-section at the nodes of section mesh. If the material is plastic, stresses and strains are moved without extrapolation to the section nodes (from section integration points).

13.189. BEAM189 - 3-D 3-Node Beam



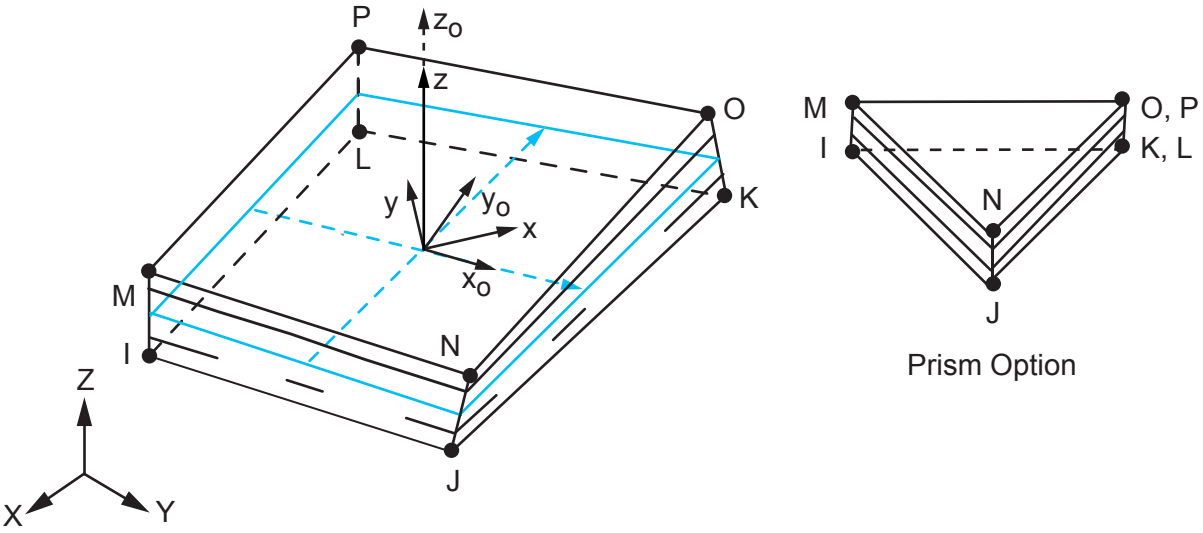
Matrix or Vector	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal and Newton-Raphson Load Vectors	Equation 11.19 (p. 330), Equation 11.20 (p. 330), Equation 11.21 (p. 330), Equation 11.22 (p. 331), Equation 11.23 (p. 331), and Equation 11.24 (p. 331)	Along the length: 2 Across the section: see BEAM188 - 3-D 2-Node Beam (p. 598)
Consistent Mass Matrix and Pressure Load Vector	Same as stiffness matrix	Along the length: 3 Across the section: 1
Lumped Mass Matrix	Equation 11.19 (p. 330), Equation 11.20 (p. 330), and Equation 11.21 (p. 330)	Along the length: 3 Across the section: 1

Load Type	Distribution
Element Temperature	Bilinear across cross-section and linear along length
Nodal Temperature	Constant across cross-section, linear along length
Pressure	Linear along length. The pressure is assumed to act along the element x-axis.

References: Simo and Vu-Quoc([237] (p. 934)), Ibrahimbegovic([238] (p. 934)).

The theory for this element is identical to that of BEAM188 - 3-D 2-Node Beam (p. 598), except that it is a nonlinear, 3-node beam element.

13.190. SOLSH190 - 3-D 8-Node Layered Solid Shell



Matrix or Vector	Shape Functions		Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Equation 11.212 (p. 353), Equation 11.213 (p. 353), and Equation 11.214 (p. 353)		In-plane: 2 x 2 Thru-the-thickness: 2 if no shell section defined 1, 3, 5, 7, or 9 per layer if a shell section is defined
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Quad	Equation 11.69 (p. 337) and Equation 11.70 (p. 337)	2 x 2
	Triangle	Equation 11.49 (p. 336) and Equation 11.50 (p. 336)	3

Load Type	Distribution
Element Temperature	Bilinear in-plane of element, linear thru each layer
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

13.190.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element.

13.190.2. Theory

SOLSH190 is a 3-D solid element free of locking in bending-dominant situations. Unlike shell elements, SOLSH190 is compatible with general 3-D constitutive relations and can be connected directly with other continuum elements.

SOLSH190 utilizes a suite of special kinematic formulations, including assumed strain method (Bathe and Dvorkin([98] (p. 926))) to overcome locking when the shell thickness becomes extremely small.

SOLSH190 employs enhanced strain formulations (Simo and Rifai([318] (p. 938)), Simo et al.([320] (p. 939))) to improve the accuracy in in-plane bending situations. The satisfaction of the in-plane patch test is ensured. Incompatible shape functions are used to overcome the thickness locking.

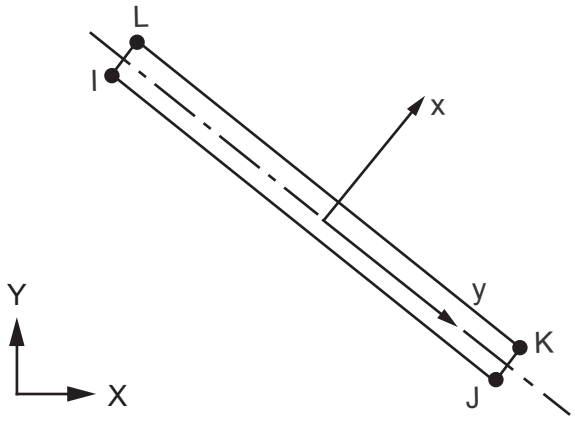
13.190.3. Shear Correction

The element allows for parabolic enhanced transverse shear strains. Stresses are then calculated from enhanced shear strains.

13.191. Reserved for Future Use

This section is reserved for future use.

13.192. INTER192 - 2-D 4-Node Gasket



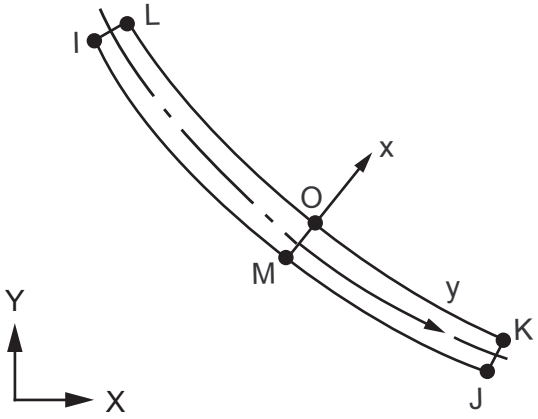
Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Linear in x and y directions	2
Thermal Load Vector	Same as stiffness matrix	Same as stiffness matrix

Load Type	Distribution
Element temperature	Based on element shape function, constant through the direction perpendicular to element plane
Nodal temperature	Same as element temperature distribution

13.192.1. Other Applicable Sections

The theory for this element is described in INTER194 - 3-D 16-Node Gasket (p. 605).

13.193. INTER193 - 2-D 6-Node Gasket



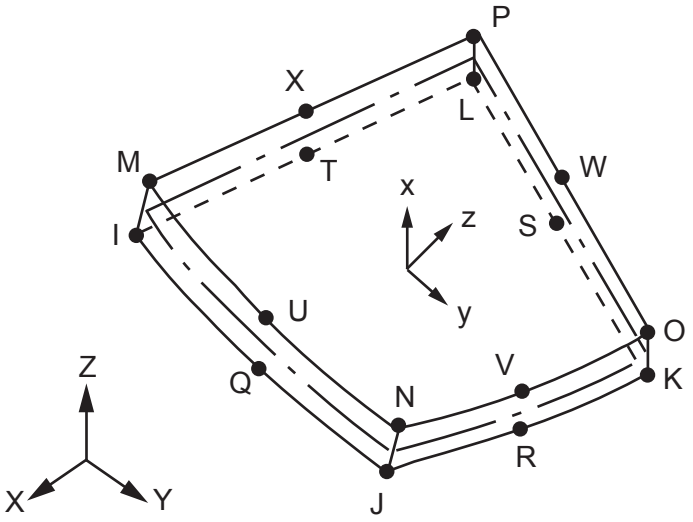
Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Linear in x, quadratic in y direction	2
Thermal Load Vector	Same as stiffness matrix	Same as stiffness matrix

Load Type	Distribution
Element temperature	Based on element shape function, constant through the direction perpendicular to element plane
Nodal temperature	Same as element temperature distribution

13.193.1. Other Applicable Sections

The theory for this element is described in INTER194 - 3-D 16-Node Gasket (p. 605).

13.194. INTER194 - 3-D 16-Node Gasket



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Linear in x, quadratic in y and z directions	2 x 2

Matrix or Vector	Shape Functions	Integration Points
Thermal Load Vector	Same as stiffness matrix	Same as stiffness matrix

Load Type	Distribution
Element temperature	Based on element shape function, constant through the direction perpendicular to element plane
Nodal temperature	Same as element temperature distribution

13.194.1. Element Technology

The element is designed specially for simulation of gasket joints, where the primary deformation is confined to the gasket through-thickness direction. The through-thickness deformation of gasket is decoupled from the other deformations and the membrane (in-plane) stiffness contribution is neglected. The element offers a direct means to quantify the through-thickness behavior of the gasket joints. The pressure-deformation behavior obtained from experimental measurement can be applied to the gasket material. See Gasket Material (p. 121) for detailed description of gasket material options.

The element is composed of bottom and top surfaces. An element midplane is created by averaging the coordinates of node pairs from the bottom and top surfaces of the elements. The numerical integration of interface elements is performed in the element midplane. The element formulation is based on a corotational procedure. The virtual work in an element is written as:

$$\delta W_{int} = \int_{S_{int}} T \delta d dS \tag{13.358}$$

where:

t = traction force across the element
d = closure across the element
S_{int} = midplane of the interface surfaces

The integration is performed in the corotational equilibrium configuration and the Gauss integration procedure is used.

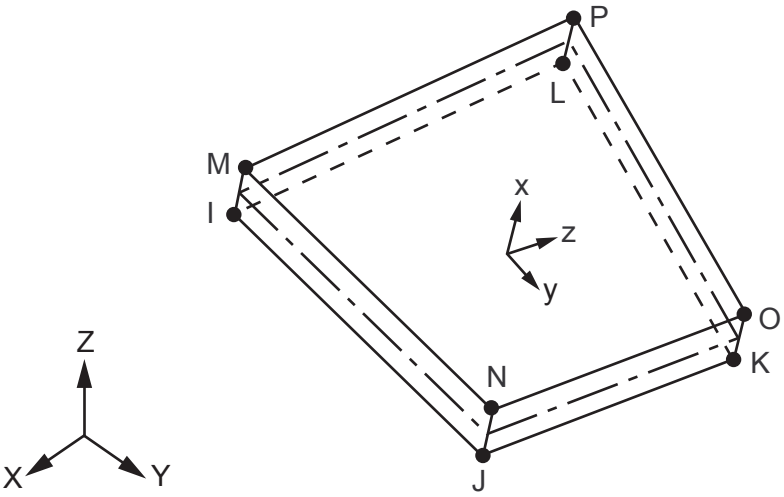
The relative deformation between top and bottom surfaces is defined as:

$$d = u^{TOP} - u^{BOTTOM} \tag{13.359}$$

where, u^{TOP} and u^{BOTTOM} are the displacement of top and bottom surfaces of interface elements in the local element coordinate system based on the midplane of element.

The thickness direction is defined as the normal direction of the mid plane of the element at the integration point.

13.195. INTER195 - 3-D 8-Node Gasket



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Linear in x, bilinear in y and z directions	2 x 2
Thermal Load Vector	Same as stiffness matrix	Same as stiffness matrix

Load Type	Distribution
Element temperature	Based on element shape function, constant through the direction perpendicular to element plane
Nodal temperature	Same as element temperature distribution

13.195.1. Other Applicable Sections

The theory for this element is described in INTER194 - 3-D 16-Node Gasket (p. 605).

13.196. Reserved for Future Use

This section is reserved for future use.

13.197. Reserved for Future Use

This section is reserved for future use.

13.198. Reserved for Future Use

This section is reserved for future use.

13.199. Reserved for Future Use

This section is reserved for future use.

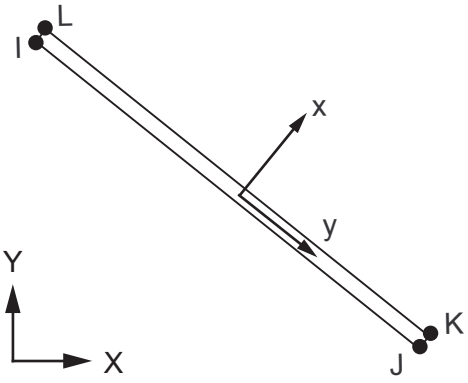
13.200. Reserved for Future Use

This section is reserved for future use.

13.201. Reserved for Future Use

This section is reserved for future use.

13.202. INTER202 - 2-D 4-Node Cohesive



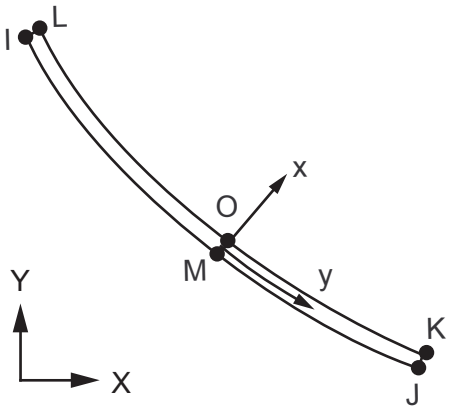
Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Linear in x and y directions	2

Load Type	Distribution
Element temperature	Based on element shape function, constant through the direction perpendicular to element plane
Nodal temperature	Same as element temperature distribution

13.202.1. Other Applicable Sections

The theory for this element is described in INTER204 - 3-D 16-Node Cohesive (p. 609).

13.203. INTER203 - 2-D 6-Node Cohesive



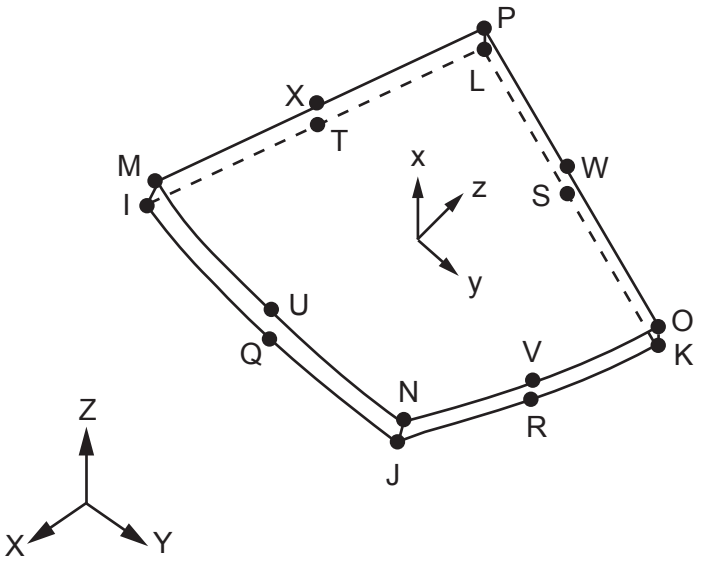
Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Linear in x, quadratic in y direction	2

Load Type	Distribution
Element temperature	Based on element shape function, constant through the direction perpendicular to element plane
Nodal temperature	Same as element temperature distribution

13.203.1. Other Applicable Sections

The theory for this element is described in INTER204 - 3-D 16-Node Cohesive (p. 609).

13.204. INTER204 - 3-D 16-Node Cohesive



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Linear in x, quadratic in y and z directions	2 x 2

Load Type	Distribution
Element temperature	Based on element shape function, constant through the direction perpendicular to element plane
Nodal temperature	Same as element temperature distribution

13.204.1. Element Technology

The element is designed specially for simulation of interface delamination and fracture, where the interface surfaces are represented by a group of interface elements, in which an interfacial constitutive relationship characterizes the traction separation behavior of the interface. The element offers a direct means to quantify the interfacial separation behavior. See Cohesive Zone Material (CZM) Model (p. 162) for detailed description of interface material options.

The virtual work of the element is written as:

$$\delta W_{int} = \int_{S_{int}} T \delta d dS \tag{13.360}$$

where:

t = traction force across the element
d = separation across the element
S_{int} = midplane of the interface surfaces

The integration is performed in the corotational equilibrium configuration and the Gauss integration procedure is used.

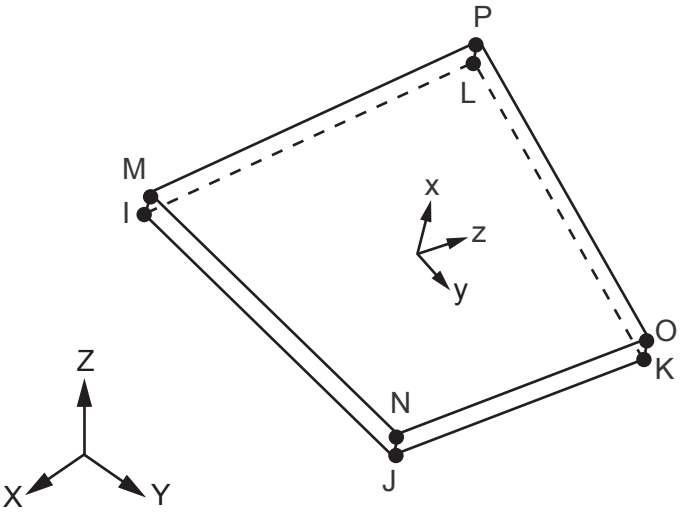
The separation, d, is defined as the relative deformation between top and bottom surfaces as:

$$d = u^{TOP} - u^{BOTTOM} \tag{13.361}$$

where, u^{TOP} and u^{BOTTOM} are the displacement of top and bottom surfaces of interface elements in the local element coordinate system based on the midplane of element.

The thickness direction is defined as the normal direction of the midplane of the element at the integration point.

13.205. INTER205 - 3-D 8-Node Cohesive



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Linear in x, bilinear in y and z directions	2 x 2

Load Type	Distribution
Element temperature	Based on element shape function, constant through the direction perpendicular to element plane
Nodal temperature	Same as element temperature distribution

13.205.1. Other Applicable Sections

The theory for this element is described in INTER204 - 3-D 16-Node Cohesive (p. 609).

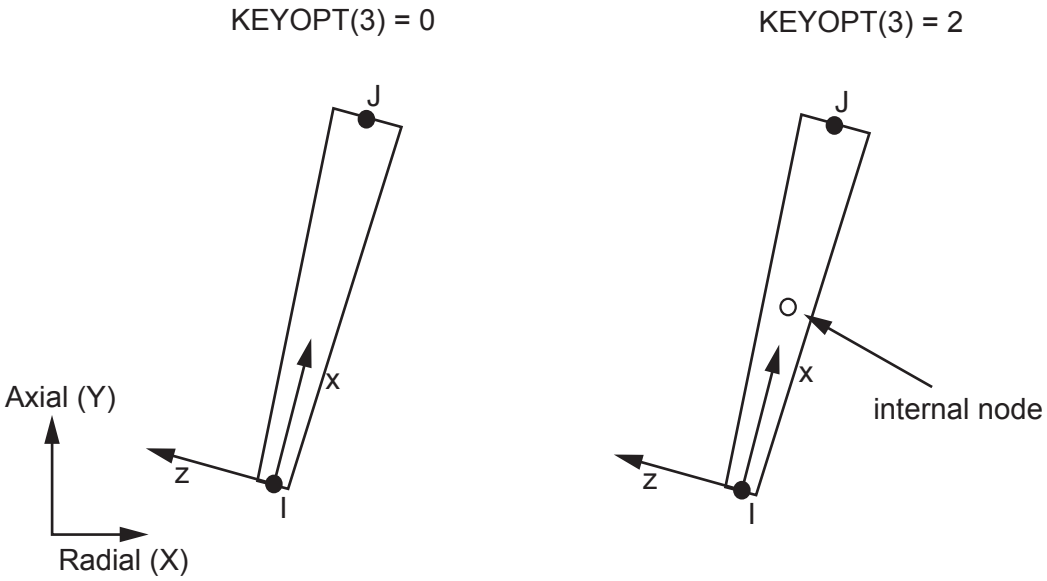
13.206. Reserved for Future Use

This section is reserved for future use.

13.207. Reserved for Future Use

This section is reserved for future use.

13.208. SHELL208 - 2-Node Axisymmetric Shell



Matrix or Vector	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrix; and Thermal and Newton-Raphson Load Vectors	KEYOPT(3) = 0: Equation 11.6 (p. 329), Equation 11.7 (p. 330), and Equation 11.11 (p. 330) KEYOPT(3) = 2: Equation 11.19 (p. 330), Equation 11.20 (p. 330), and Equation 11.24 (p. 331)	Along-the-length: 1 (KEYOPT(3) = 0) 2 (KEYOPT(3) = 2) Thru-the-thickness: 1, 3, 5, 7, or 9 per layer
Mass Matrix and Pressure Load Vector	Same as stiffness matrix	Along-the-length: 2 (KEYOPT(3) = 0) 3 (KEYOPT(3) = 2) Thru-the-thickness: 1, 3, 5, 7, or 9 per layer

Load Type	Distribution
Element Temperature	Linear along length and linear thru thickness
Nodal Temperature	Linear along length and constant thru thickness
Pressure	Linear along length

References: Ahmad([1] (p. 921)), Cook([5] (p. 921))

13.208.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations.

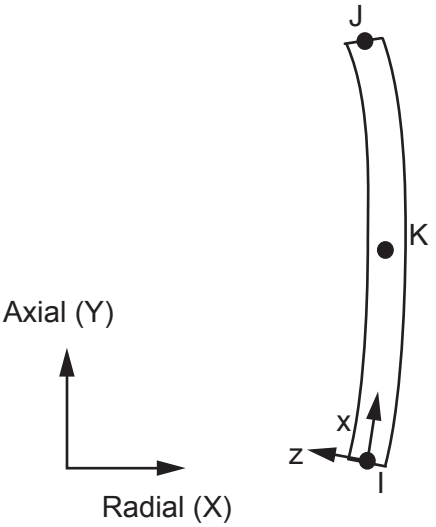
13.208.2. Assumptions and Restrictions

Normals to the centerline are assumed to remain straight after deformation, but not necessarily normal to the centerline.

13.208.3. Shear Correction

The element uses an equivalent energy method to compute shear-correction factors. These factors are predetermined based on the section lay-up at the start of solution.

13.209. SHELL209 - 3-Node Axisymmetric Shell



Matrix or Vector	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrix; and Thermal and Newton-Raphson Load Vectors	Equation 11.19 (p. 330), Equation 11.20 (p. 330), and Equation 11.24 (p. 331)	Along-the-length: 2 Thru-the-thickness: 1, 3, 5, 7, or 9 per layer
Mass Matrix and Pressure Load Vector	Same as stiffness matrix	Along-the-length: 3 Thru-the-thickness: 1, 3, 5, 7, or 9 per layer

Load Type	Distribution
Element Temperature	Linear along length and linear thru thickness
Nodal Temperature	Linear along length and constant thru thickness
Pressure	Linear along length

References: Ahmad([1] (p. 921)), Cook([5] (p. 921))

13.209.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations.

13.209.2. Assumptions and Restrictions

Normals to the centerline are assumed to remain straight after deformation, but not necessarily normal to the centerline.

13.209.3. Shear Correction

The element uses an equivalent energy method to compute shear-correction factors. These factors are predetermined based on the section lay-up at the start of solution.

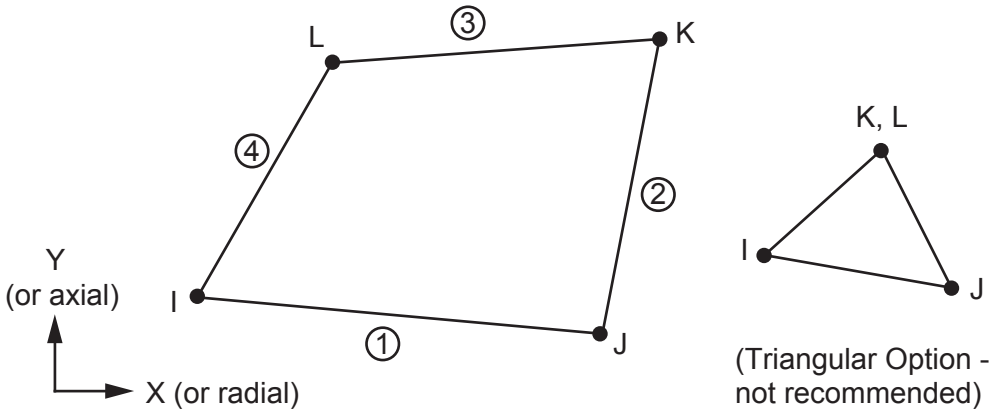
13.210. Reserved for Future Use

This section is reserved for future use.

13.211. Reserved for Future Use

This section is reserved for future use.

13.212. CPT212 - 2-D 4-Node Coupled Pore-Pressure Mechanical Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Quad	Equation 11.120 (p. 341) and Equation 11.121 (p. 341)	2 x 2
	Tri-angle	Equation 11.100 (p. 340) and Equation 11.101 (p. 340)	1
Mass Matrix	Quad	Same as stiffness matrix	2 x 2
	Tri-angle		1
Damping matrices	Same as stiffness matrix		
Pressure Load Vec-tor	Same as stiffness matrix, specialized to edge		2
Flow Load Vector	Linear across each edge		2

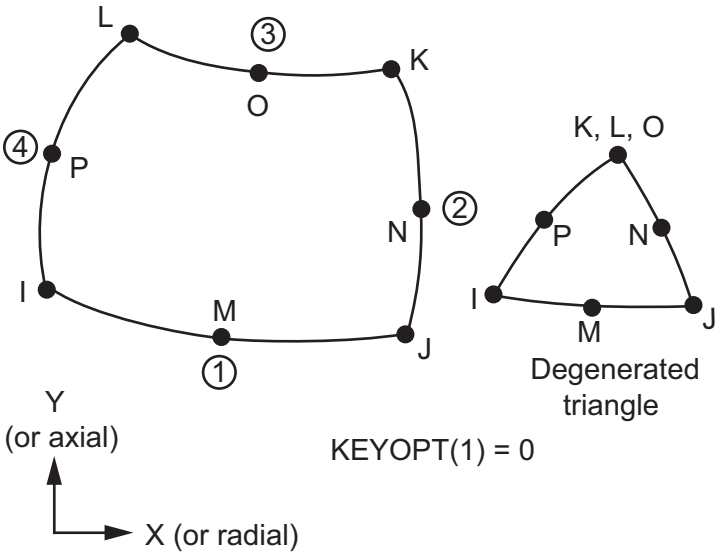
Load Type	Distribution
Element Temperature	Bilinear across element, constant thru thickness or around cir-cumference

Load Type	Distribution
Nodal Temperature	Same as element temperature distribution
Surface Pressure Load	Linear along each edge
Flow flux	Linear along each edge

13.212.1. Other Applicable Sections

General Element Formulations gives the general element formulations used by this element.

13.213. CPT213 - 2-D 8-Node Coupled Pore-Pressure Mechanical Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Quad	Equation 11.134 (p. 342) and Equation 11.135 (p. 342)	2 x 2
	Triangle	Equation 11.112 (p. 340) and Equation 11.113 (p. 340)	3
Mass Matrix	Quad	Same as stiffness matrix	3 x 3
	Triangle		3
Pressure Load Vector	Same as stiffness matrix, specialized to the edge		2
Flow Load Vector	Linear across each edge		2

Load Type	Distribution
Element Temperature	Same as shape functions across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Surface Pressure Load	Linear along each edge
Flow flux	Linear along each edge

Reference: Zienkiewicz(39)

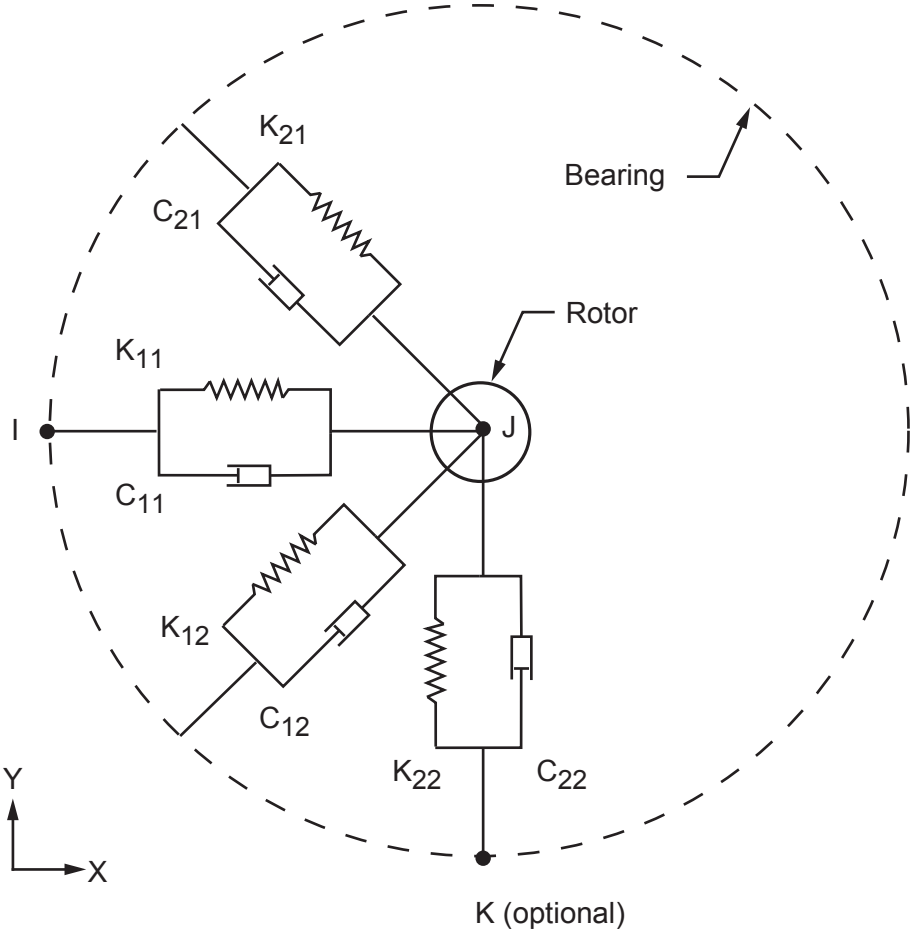
13.213.1. Other Applicable Sections

General Element Formulations gives the general element formulations used by this element.

13.213.2. Assumptions and Restrictions

A dropped midside node implies that the edge is and remains straight.

13.214. COMBI214 - 2-D Spring-Damper Bearing



Matrix or Vector	Shape Functions	Integration Points
Stiffness and Damping Matrices	None	None
Stress Stiffening Matrix	None	None

13.214.1. Matrices

If KEYOPT(2) = 0, the element lies in the (XY) plane and the stiffness, damping and stress-stiffness matrices in nodal coordinates are:

$$[K_e] = \begin{bmatrix} K_{11} & K_{12} & 0 & -K_{11} & -K_{12} & 0 \\ K_{21} & K_{22} & 0 & -K_{21} & -K_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -K_{11} & -K_{12} & 0 & K_{11} & K_{12} & 0 \\ -K_{21} & -K_{22} & 0 & K_{21} & K_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{13.362}$$

$$[C_e] = \begin{bmatrix} C_{11} & C_{12} & 0 & -C_{11} & -C_{12} & 0 \\ C_{21} & C_{22} & 0 & -C_{21} & -C_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -C_{11} & -C_{12} & 0 & C_{11} & C_{12} & 0 \\ -C_{21} & -C_{22} & 0 & C_{21} & C_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{13.363}$$

$$[S_e] = \begin{bmatrix} \frac{K_{11}\varepsilon_0^1}{L_1} & \frac{K_{12}\varepsilon_0^2}{L_2} & 0 & -\frac{K_{11}\varepsilon_0^1}{L_1} & -\frac{K_{12}\varepsilon_0^2}{L_2} & 0 \\ \frac{K_{21}\varepsilon_0^1}{L_1} & \frac{K_{22}\varepsilon_0^2}{L_2} & 0 & -\frac{K_{21}\varepsilon_0^1}{L_1} & -\frac{K_{22}\varepsilon_0^2}{L_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{K_{11}\varepsilon_0^1}{L_1} & -\frac{K_{12}\varepsilon_0^2}{L_2} & 0 & \frac{K_{11}\varepsilon_0^1}{L_1} & \frac{K_{12}\varepsilon_0^2}{L_2} & 0 \\ -\frac{K_{21}\varepsilon_0^1}{L_1} & -\frac{K_{22}\varepsilon_0^2}{L_2} & 0 & \frac{K_{21}\varepsilon_0^1}{L_1} & \frac{K_{22}\varepsilon_0^2}{L_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{13.364}$$

where:

K_{11}, K_{12}, K_{21}, K_{22} = stiffness coefficients (input as K_{11}, etc. on **R** command)
C_{11}, C_{12}, C_{21}, C_{22} = damping coefficients (input as C_{11}, etc. on **R** command)
$\varepsilon_0^1, \varepsilon_0^2$ = stretches in element from previous iteration
L_1 = distance between the two nodes I and J
L_2 = distance between the two nodes K and J

The matrices for KEYOPT(2) equals 1 or 2 are developed analogously.

Stiffness and/or damping matrices may depend upon the rotational velocity (input through OMEGA or CMOMEGA) if real constants are defined as table parameters.

13.214.2. Output Quantities

The stretch is computed as:

$$\varepsilon_0^1 = \begin{Bmatrix} u'_J - u'_I & \text{if KEYOPT(2) = 0} \\ v'_J - v'_I & \text{if KEYOPT(2) = 1} \\ u'_J - u'_I & \text{if KEYOPT(2) = 2} \end{Bmatrix} \text{(output as STRETCH1)} \tag{13.365}$$

$$\varepsilon_0^2 = \begin{Bmatrix} v'_J - v'_I & \text{if KEYOPT(2) = 0} \\ w'_J - w'_I & \text{if KEYOPT(2) = 1} \\ w'_J - w'_I & \text{if KEYOPT(2) = 2} \end{Bmatrix} \text{(output as STRETCH2)} \tag{13.366}$$

where:

u', v', w' = displacements in nodal Cartesian coordinates (UX, UY, UZ)

The static forces are computed as:

$$F_s^1 = K_{11}\varepsilon_0^1 + K_{12}\varepsilon_0^2 \quad \text{(output as FORC1)} \tag{13.367}$$

$$F_s^2 = K_{21}\varepsilon_0^1 + K_{22}\varepsilon_0^2 \quad \text{(output as FORC2)} \tag{13.368}$$

Finally, if a nonlinear transient dynamic (**ANTYPE**,TRANS, with **TIMINT**,ON) analysis is performed, a damping force is computed:

The damping forces are computed as:

$$F_D^1 = C_{11}v^1 + C_{12}v^2 \quad \text{(output as DAMPING FORCE1)} \tag{13.369}$$

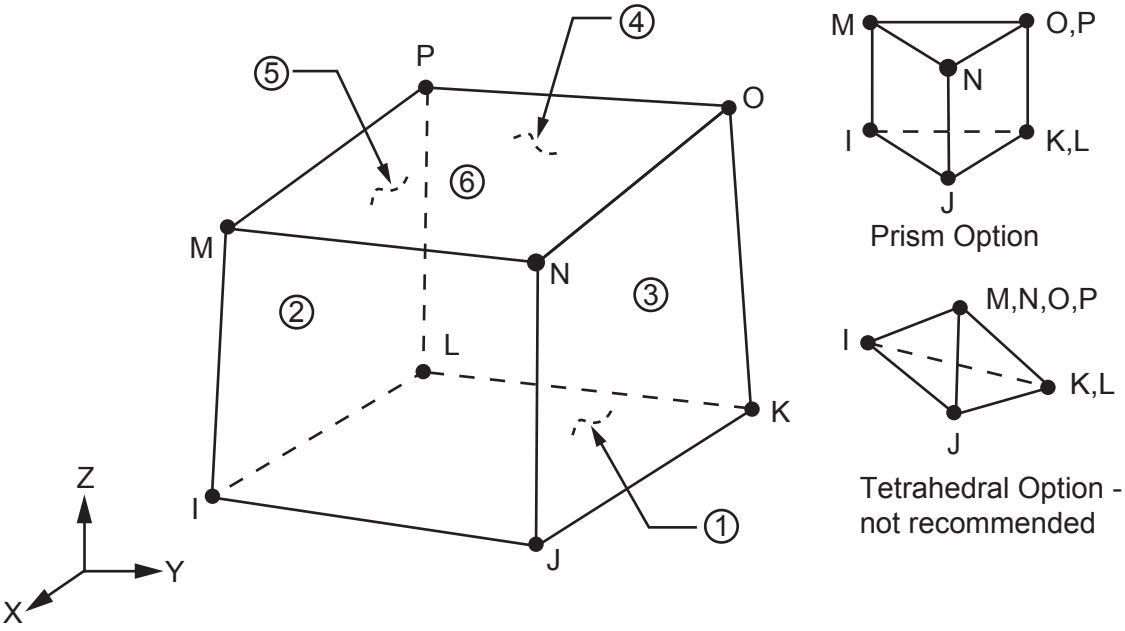
$$F_D^2 = C_{21}v^1 + C_{22}v^2 \quad \text{(output as DAMPING FORCE2)} \tag{13.370}$$

where:

v^1, v^2 = relative velocities

Relative velocities are computed using Equation 13.365 (p. 617) and Equation 13.366 (p. 617), where the nodal displacements u', v', and w' are replaced with the nodal Newmark velocities.

13.215. CPT215 - 3-D 8-Node Coupled Pore-Pressure Mechanical Solid



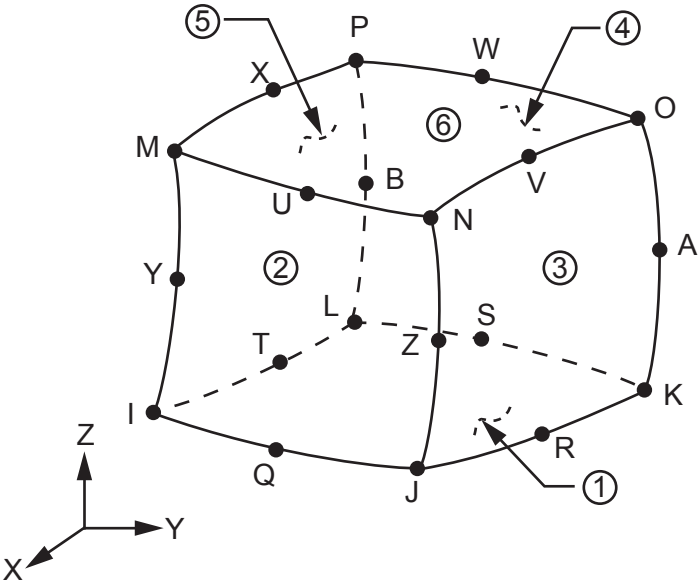
Matrix or Vector	Shape Functions		Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Equation 11.212 (p. 353), Equation 11.213 (p. 353), and Equation 11.214 (p. 353)		2 x 2 x 2
Mass Matrix	Same as stiffness matrix		2 x 2 x 2
Pressure Load Vector Mass Matrix	Quad	Equation 11.69 (p. 337) and Equation 11.70 (p. 337)	2 x 2
	Triangle	Equation 11.49 (p. 336) and Equation 11.50 (p. 336)	3
Damping matrices	Same as stiffness		
Flow Load Vector	Same as Pressure Load Vector		

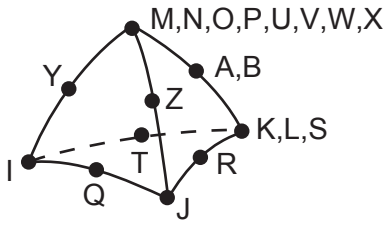
Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Surface Pressure Load	Bilinear across each face
Flow flux	Bilinear across each face

13.215.1. Other Applicable Sections

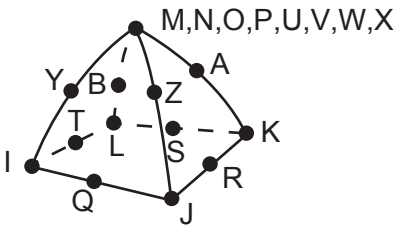
General Element Formulations gives the general element formulations used by this element.

13.216. CPT216 - 3-D 20-Node Coupled Pore-Pressure Mechanical Solid

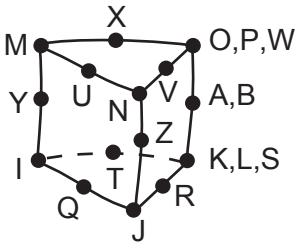




Tetrahedral Option



Pyramid Option



Prism Option

Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Brick	Equation 11.230 (p. 354), Equation 11.231 (p. 354), and Equation 11.232 (p. 355)	14
	Wedge	Equation 11.206 (p. 352), Equation 11.207 (p. 352), and Equation 11.208 (p. 352)	3 x 3
	Pyramid	Equation 11.190 (p. 350), Equation 11.191 (p. 350), and Equation 11.192 (p. 350)	2 x 2 x 2
	Tet	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348)	4
Mass Matrix	Same as stiffness matrix.		3 x 3 x 3 if brick. If other shapes, same as stiffness matrix
Pressure Load Vector	Quad	Equation 11.84 (p. 338) and Equation 11.85 (p. 338)	3 x 3
	Triangle	Equation 11.57 (p. 336) and Equation 11.58 (p. 336)	6
Flow Load Vector	Quad	Equation 11.84 (p. 338) and Equation 11.85 (p. 338)	3 x 3

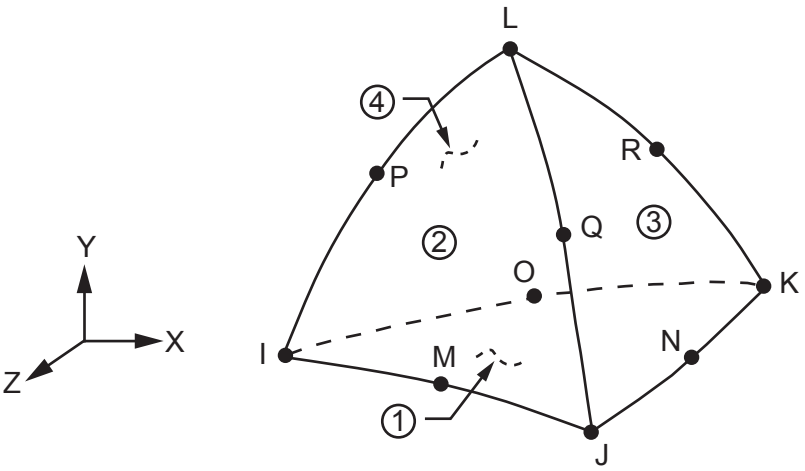
Matrix or Vector	Geometry	Shape Functions	Integration Points
	Triangle	Equation 11.57 (p. 336) and Equation 11.58 (p. 336)	6
damping matrices	Same as stiffness		

Load Type	Distribution
Element Temperature	Bilinear in plane of element
Nodal Temperature	Same as shape functions thru element
Surface Pressure Load	Bilinear across each face
Flow flux	Bilinear across each face

13.216.1. Other Applicable Sections

General Element Formulations gives the general element formulations used by this element.

13.217. CPT217 - 3-D 10-Node Coupled Pore-Pressure Mechanical Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness, Mass, and Stress Stiffness Matrices; and Thermal Load Vector	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348)	4
Pressure Load Vector	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348) specialized to the face	6
Flow Load Vector	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348) specialized to the face	6
Mass Matrix	Same as stiffness matrix	
Damping Matrices	Same as stiffness matrix	

Load Type	Distribution
Element Temperature	Same as shape functions
Nodal Temperature	Same as shape functions
Surface Pressure Load	Linear over each face
Flow flux	Linear across each face

13.217.1. Other Applicable Sections

General Element Formulations gives the general element formulations used by this element.

13.218. Reserved for Future Use

This section is reserved for future use.

13.219. Reserved for Future Use

This section is reserved for future use.

13.220. Reserved for Future Use

This section is reserved for future use.

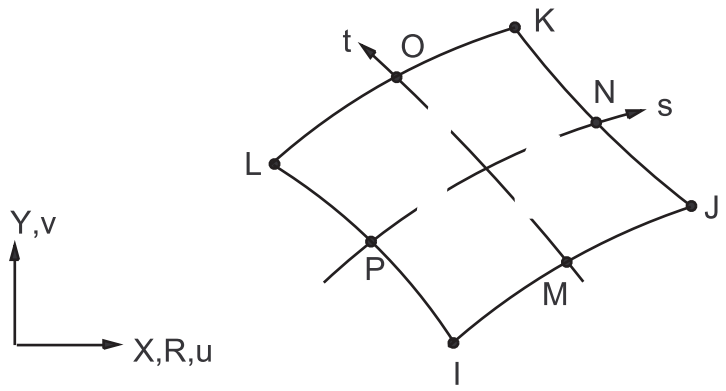
13.221. Reserved for Future Use

This section is reserved for future use.

13.222. Reserved for Future Use

This section is reserved for future use.

13.223. PLANE223 - 2-D 8-Node Coupled-Field Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices;	Quad	Equation 11.134 (p. 342) and Equation 11.135 (p. 342)	2 x 2

Matrix or Vector	Geometry	Shape Functions	Integration Points
Thermal Expansion, Diffusion Expansion, and Electrostatic Force Load Vectors	Triangle	Equation 11.112 (p. 340) and Equation 11.113 (p. 340)	3
Mass Matrix	Quad	Same as stiffness matrix	3 x 3
	Triangle		3
Pressure Load Vector	Same as stiffness matrix, specialized to the face		2
Thermal Conductivity Matrix and Heat Generation Load Vector	Quad	Equation 11.138 (p. 342)	2 x 2
	Triangle	Equation 11.117 (p. 341)	3
Specific Heat Matrix	Same as thermal conductivity matrix		
Convection Surface Matrix and Load Vector	Same as thermal conductivity matrix, specialized to the face		2
Dielectric Permittivity and Electrical Conductivity Matrices; Charge Density, Joule Heating, and Peltier Heat Flux Load Vectors	Quad	Equation 11.139 (p. 342)	2 x 2
	Triangle	Equation 11.118 (p. 341)	3
Diffusivity Matrix and Diffusing Substance Generation Load Vector	Quad	Equation 11.140 (p. 342)	2 x 2
	Triangle	Equation 11.119 (p. 341)	3
Diffusion Damping Matrix	Same as diffusivity matrix. If KEYOPT(10)=1, matrix is diagonalized as described in Lumped Matrices (p. 391)		
Diffusion Flux Load Vector	Same as diffusivity matrix, specialized to the face		2
Thermoelastic stiffness and Damping Matrices	Same as combination of stiffness and thermal conductivity matrices		
Piezoelectric Coupling Matrix	Same as combination of stiffness matrix and dielectric matrix		
Seebeck Coefficient Coupling Matrix	Same as combination of electrical conductivity and thermal conductivity matrices		
Diffusion-Elastic Stiffness Matrix	Same as combination of stiffness and diffusivity matrices		
Surface Charge Density Load Vector	Same as dielectric matrix, specialized to the face		2

13.223.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element. Electromagnetics (p. 177) describes the derivation of dielectric and electric conduction matrices. Piezoelectrics (p. 313) discusses the piezoelectric capability used by the element. Piezoresistivity (p. 318) discusses the piezoresistive effect. Thermoelectrics (p. 319) discusses the thermoelectric effects. Thermoelasticity (p. 309) discusses the thermoelastic effects. Electroelasticity (p. 317) discusses the Maxwell

stress electroelastic coupling. Thermoplasticity (p. 312) discusses the thermoplastic effect. Structural-Diffusion Coupling (p. 323) discusses diffusion strain coupling.

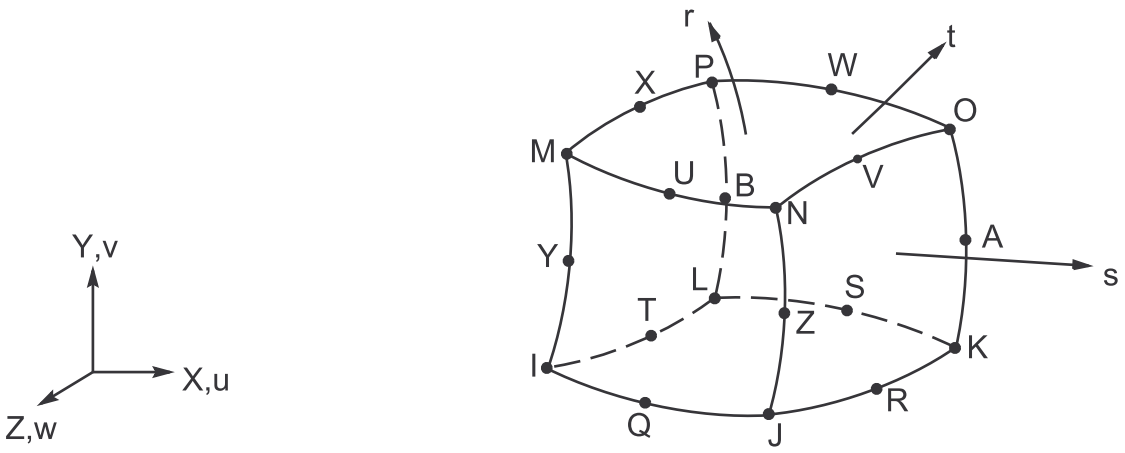
13.224. Reserved for Future Use

This section is reserved for future use.

13.225. Reserved for Future Use

This section is reserved for future use.

13.226. SOLID226 - 3-D 20-Node Coupled-Field Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal Expansion, Diffusion Expansion, and Electrostatic Force Load Vector	Brick	Equation 11.230 (p. 354), Equation 11.231 (p. 354), and Equation 11.232 (p. 355)	14 if KEYOPT(6) = 0 2 x 2 x 2 if KEYOPT(6) = 1
	Wedge	Equation 11.206 (p. 352), Equation 11.207 (p. 352), and Equation 11.208 (p. 352)	3 x 3
	Pyramid	Equation 11.190 (p. 350), Equation 11.191 (p. 350), and Equation 11.192 (p. 350)	2 x 2 x 2
	Tet	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348)	4
Mass Matrix	Same as stiffness matrix.		3 x 3 x 3 if brick. If other shapes, same as
Pressure Load Vector	Quad	Equation 11.84 (p. 338) and Equation 11.85 (p. 338)	3 x 3
	Triangle	Equation 11.57 (p. 336) and Equation 11.58 (p. 336)	6

Matrix or Vector	Geo-metry	Shape Functions	Integration Points
Thermal Conductivity Matrix and Heat Generation Load Vector	Brick	Equation 11.233 (p. 355)	14 [1]
	Wedge	Equation 11.209 (p. 352)	3 x 3
	Pyramid	Equation 11.193 (p. 350)	2 x 2 x 2
	Tet	Equation 11.185 (p. 348)	4
Specific Heat Matrix	Same as thermal conductivity matrix		
Convection Surface Matrix and Load Vector	Quad	Equation 11.91 (p. 338)	3 x 3
	Triangle	Equation 11.63 (p. 337)	6
Dielectric Permittivity and Electrical Conductivity Matrices; Charge Density, Joule Heating, and Peltier Heat Flux Load Vectors	Brick	Equation 11.234 (p. 355)	14 [1]
	Wedge	Equation 11.210 (p. 352)	3 x 3
	Pyramid	Equation 11.194 (p. 350)	2 x 2 x 2
	Tet	Equation 11.186 (p. 348)	4
Diffusivity Matrix and Diffusing Substance Generation Load Vector	Brick	Equation 11.236 (p. 355)	14 [1]
	Wedge	Equation 11.211 (p. 352)	3 x 3
	Pyramid	Equation 11.195 (p. 350)	2 x 2 x 2
	Tet	Equation 11.188 (p. 348)	4
Diffusion Damping Matrix	Same as diffusivity matrix. If KEYOPT(10)=1, matrix is diagonalized as described in Lumped Matr		
Diffusion Flux Load Vector	Quad	Equation 11.93 (p. 339)	3 x 3
	Triangle	Equation 11.65 (p. 337)	6
Thermoelastic stiffness and Damping Matrices	Same as combination of stiffness and thermal conductivity matrices		
Piezoelectric Coupling Matrix	Same as combination of stiffness matrix and dielectric matrix		
Seebeck Coefficient Coupling Matrix	Same as combination of electrical conductivity and thermal conductivity matrices		
Diffusion-elastic stiffness and damping matrices	Same as combination of stiffness and diffusivity matrices		
Surface Charge Density Load Vector	Quad	Equation 11.194 (p. 350)	3 x 3
	Triangle	Equation 11.64 (p. 337)	6

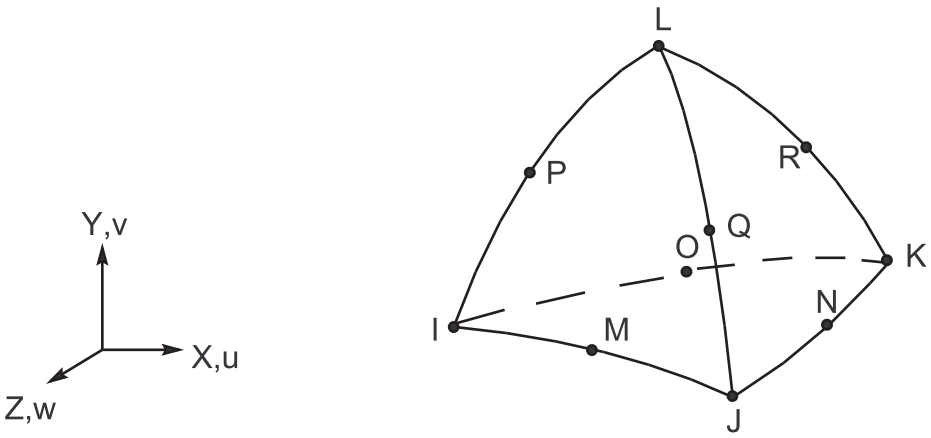
1. When KEYOPT(6) = 1 in a coupled-field analysis with structural DOFs, the unified reduced integration scheme with 2 x 2 x 2 integration points is used to form the thermal, electric, electrostatic matrices and load vectors.

13.226.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element. Electromagnetics (p. 177) describes the derivation of dielectric and electric conduction matrices. Piezoelectrics (p. 313) discusses the piezoelectric capability used by the element. Piezoresistivity (p. 318) discusses the piezoresistive effect. Thermoelectrics (p. 319) discusses the thermoelectric effects.

Thermoelasticity (p. 309) discusses the thermoelastic effects. Electroelasticity (p. 317) discusses the Maxwell stress electroelastic coupling. Thermoplasticity (p. 312) discusses the thermoplastic effect. Structural-Diffusion Coupling (p. 323) discusses diffusion strain coupling.

13.227. SOLID227 - 3-D 10-Node Coupled-Field Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness, Mass, and Stress Stiffness Matrices; and Thermal Expansion and Electrostatic Force Load Vectors	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348)	4
Pressure Load Vector	Equation 11.182 (p. 348), Equation 11.183 (p. 348), and Equation 11.184 (p. 348) specialized to the face	6
Thermal Conductivity Matrix and Heat Generation Load Vector	Equation 11.185 (p. 348)	2 x 2
Specific Heat Matrix	Same as thermal conductivity matrix	11
Convection Surface Matrix and Load Vector	Equation 11.185 (p. 348) specialized to the face. Consistent surface matrix.	6
Dielectric Permittivity and Electrical Conductivity Matrices; Charge Density, Joule Heating, and Peltier Heat Flux Load Vectors	Equation 11.186 (p. 348)	2 x 2
Diffusivity Matrix and Diffusing Substance Generation Load Vector	Equation 11.188 (p. 348)	2 x 2
Damping Matrix	Same as diffusivity matrix. If KEYOPT(10) = 1, the matrix is diagonalized as described in Lumped Matrices (p. 391)	4

Matrix or Vector	Shape Functions	Integration Points
Diffusion Flux Load Vector	Equation 11.188 (p. 348) specialized to the face	6
Thermoelastic Stiffness and Damping Matrices	Same as combination of stiffness and thermal conductivity matrices	
Piezoelectric Coupling Matrix	Same as combination of stiffness matrix and dielectric matrix	
Seebeck Coefficient Coupling Matrix	Same as combination of electrical conductivity and thermal conductivity matrices	
Diffusion-Elastic Stiffness and Damping Matrices	Same as combination of stiffness and diffusivity matrices	
Surface Charge Density Load Vector	Equation 11.186 (p. 348) specialized to the face	6

13.227.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element. Electromagnetics (p. 177) describes the derivation of dielectric and electric conduction matrices. Piezoelectrics (p. 313) discusses the piezoelectric capability used by the element. Piezoresistivity (p. 318) discusses the piezoresistive effect. Thermoelectrics (p. 319) discusses the thermoelectric effects. Thermoelasticity (p. 309) discusses the thermoelastic effects. Electroelasticity (p. 317) discusses the Maxwell stress electroelastic coupling. Thermoplasticity (p. 312) discusses the thermoplastic effect. Structural-Diffusion Coupling (p. 323) discusses diffusion strain coupling.

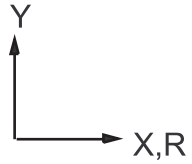
13.228. Reserved for Future Use

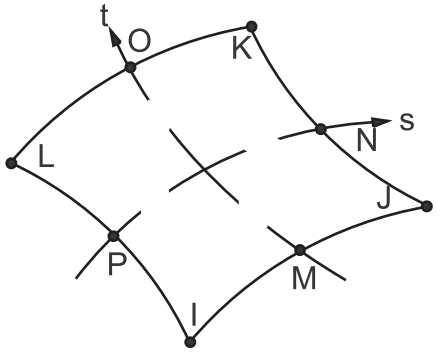
This section is reserved for future use.

13.229. Reserved for Future Use

This section is reserved for future use.

13.230. PLANE230 - 2-D 8-Node Electric Solid





Matrix or Vector	Geometry	Shape Functions	Integration Points
Electrical Conductivity and Dielectric Permittivity Coefficient Matrices	Quad	Equation 11.139 (p. 342)	3 x 3
	Triangle	Equation 11.118 (p. 341)	3

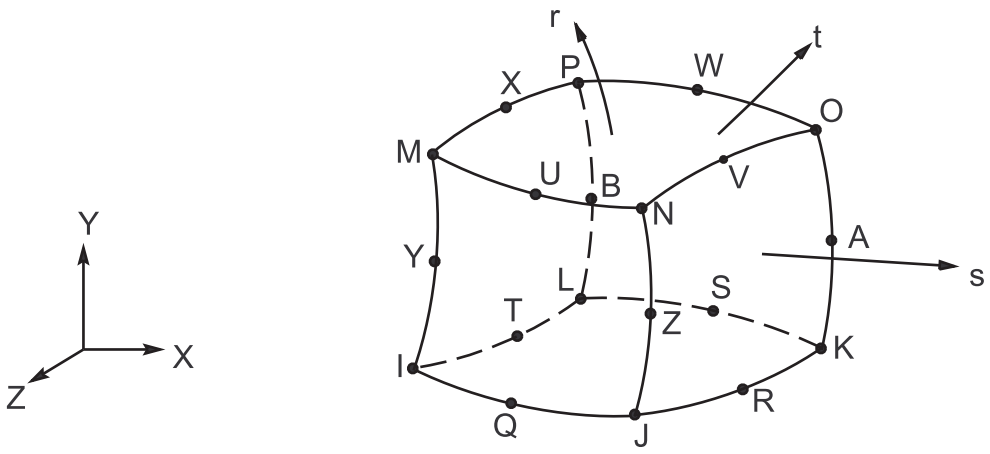
13.230.1. Other Applicable Sections

Electromagnetics (p. 177) describes the derivation of the electric element matrices and load vectors as well as electric field evaluations.

13.230.2. Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the potential varies linearly along that edge.

13.231. SOLID231 - 3-D 20-Node Electric Solid

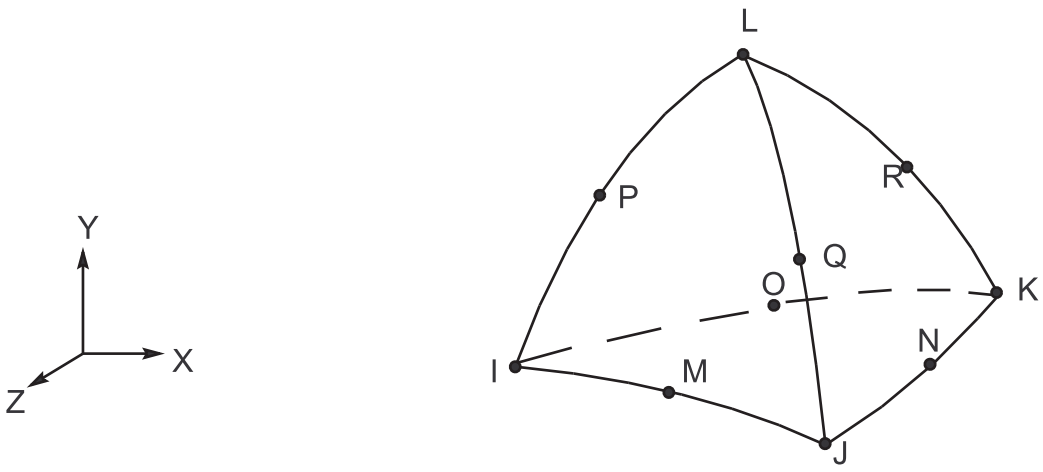


Matrix or Vector	Geometry	Shape Functions	Integration Points
Electrical Conductivity and Dielectric Permittivity Coefficient Matrices	Brick	Equation 11.234 (p. 355)	14
	Wedge	Equation 11.210 (p. 352)	3 x 3
	Pyramid	Equation 11.194 (p. 350)	8
	Tet	Equation 11.186 (p. 348)	4

13.231.1. Other Applicable Sections

Electromagnetics (p. 177) describes the derivation of electric element matrices and load vectors as well as electric field evaluations.

13.232. SOLID232 - 3-D 10-Node Tetrahedral Electric Solid

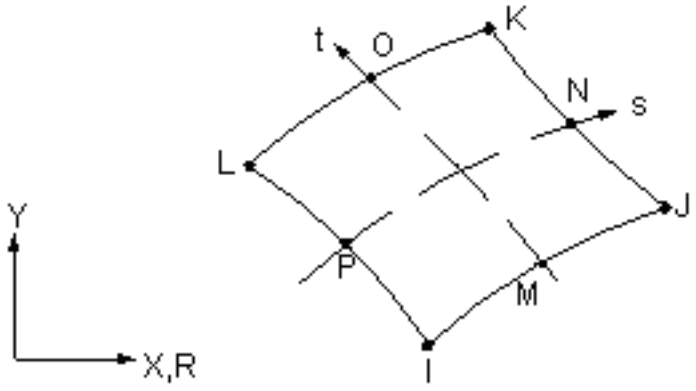


Matrix or Vector	Shape Functions	Integration Points
Dielectric Permittivity and Electrical Conductivity Coefficient Matrices, Charge Density Load Vector	Equation 11.186 (p. 348)	4

13.232.1. Other Applicable Sections

Electromagnetics (p. 177) describes the derivation of electric element matrices and load vectors as well as electric field evaluations.

13.233. PLANE233 - 2-D 8-Node Electromagnetic Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Magnetic Potential Coefficient, Eddy Current Damping or Displacement Current Mass Matrices; and	Quad	Equation 11.137 (p. 342)	3 x 3
	Quad - Axisymmetric Massive Conductor	Equation 11.137 (p. 342)	2 x 2

Matrix or Vector	Geometry	Shape Functions	Integration Points
Permanent Magnet and Applied Current Load Vectors	(KEYOPT(1)=1 and KEYOPT(3)=1)		
	Triangle	Equation 11.115 (p. 341)	3
Electrical Conductivity and Displacement Current Damping Matrices	Quad	Equation 11.137 (p. 342) and Equation 11.139 (p. 342)	Same as coefficient matrix
	Triangle	Equation 11.115 (p. 341) and Equation 11.139 (p. 342)	Same as coefficient matrix
Load Type		**Distribution**	
Current Density and Phase Angle Distribution		Bilinear across element	

13.233.1. Other Applicable Sections

Electromagnetics (p. 177) describes the derivation of the electric element matrices and load vectors as well as electric field evaluations.

13.233.2. Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the potential varies linearly along that edge.

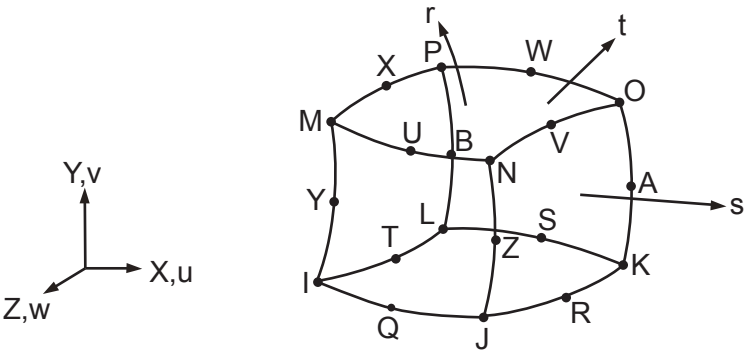
13.234. Reserved for Future Use

This section is reserved for future use.

13.235. Reserved for Future Use

This section is reserved for future use.

13.236. SOLID236 - 3-D 20-Node Electromagnetic Solid

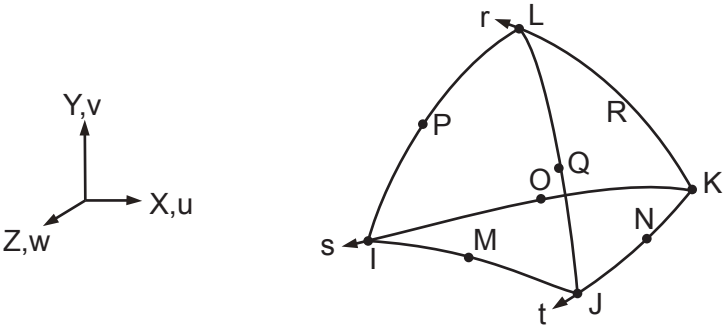


Matrix or Vector	Geometry	Shape Functions	Integration Points
Magnetic Reluctivity, Eddy Current Damping, Displacement Current Mass Matrices, Source Current Density and Remnant Magnetization Load Vectors	Brick	Equation 11.276 (p. 363), Equation 11.276 (p. 363), and Equation 11.276 (p. 363)	14
	Wedge		3 x 3
	Pyramid		3 x 3 x 3
	Tet	Equation 11.270 (p. 362), Equation 11.271 (p. 362), Equation 11.272 (p. 362), Equation 11.273 (p. 362), Equation 11.274 (p. 362), and Equation 11.275 (p. 362),	4
Electrical Conductivity and Displacement Current Damping Matrices	Brick	Equation 11.234 (p. 355)	14
	Wedge	Equation 11.210 (p. 352)	3 x 3
	Pyramid	Equation 11.194 (p. 350)	3 x 3 x 3
	Tet	Equation 11.186 (p. 348)	4
Magneto-Electric Coupling Matrix	Combination of eddy current and electrical conductivity matrices		
Magneto-Dielectric Coupling Matrix	Combination displacement current mass and damping matrices		

13.236.1. Other Applicable Sections

Electromagnetics (p. 177) describes the derivation of element matrices and load vectors as well as electromagnetic field evaluations.

13.237. SOLID237 - 3-D 10-Node Electromagnetic Solid



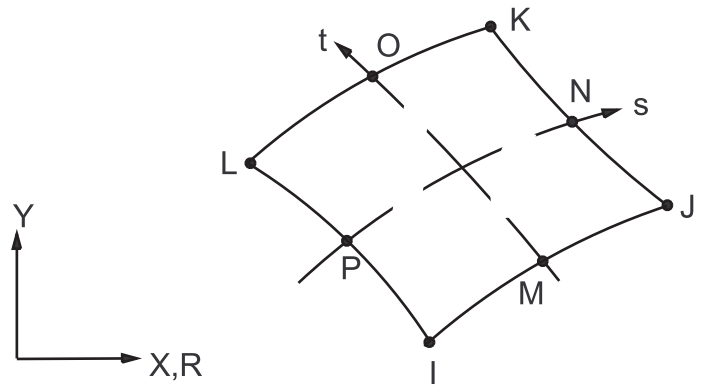
Matrix or Vector	Shape Functions	Integration Points
Magnetic Reluctivity, Eddy Current Damping, Displacement Current Mass Matrices, Source Current Density and Remnant Magnetization Load Vectors	Equation 11.270 (p. 362), Equation 11.271 (p. 362), Equation 11.272 (p. 362), Equation 11.273 (p. 362), Equation 11.274 (p. 362), andEquation 11.275 (p. 362),	4
Electrical Conductivity and Displacement Current Damping Matrices	Equation 11.186 (p. 348)	4

Matrix or Vector	Shape Functions	Integration Points
Magneto-Electric Coupling Matrix	Combination of eddy current and electrical conductivity matrices	
Magneto-Dielectric Coupling Matrix	Combination displacement current mass and damping matrices	

13.237.1. Other Applicable Sections

Electromagnetics (p. 177) describes the derivation of element matrices and load vectors as well as electromagnetic field evaluations.

13.238. PLANE238 - 2-D 8-Node Diffusion Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Diffusivity Matrix and Diffusing Substance Generation Load Vector	Quad	Equation 11.140 (p. 342)	3 x 3
	Triangle	Equation 11.119 (p. 341)	6
Damping Matrix	Same as diffusivity matrix. If KEYOPT(1) = 1, matrix is diagonalized as described in Lumped Matrices (p. 391)		
Diffusion Flux Load Vector	Same as coefficient matrix, specialized to the face		2

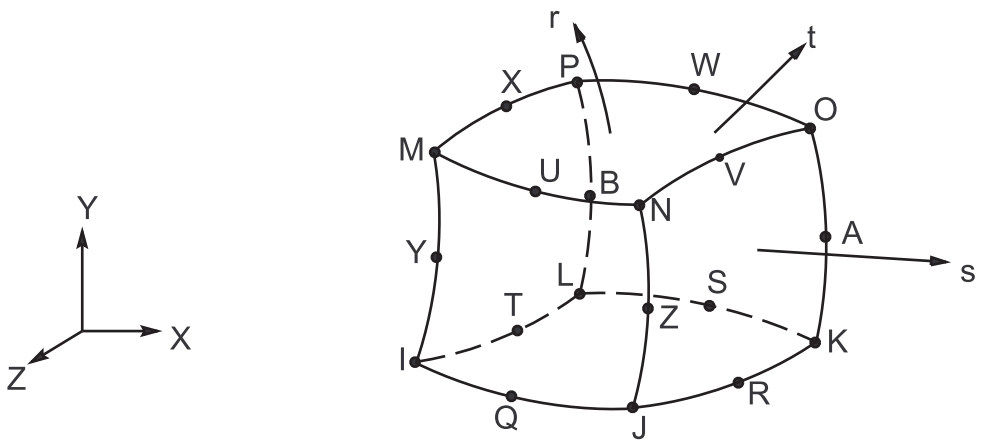
13.238.1. Other Applicable Sections

Diffusion in the *Mechanical APDL Theory Reference* describes the derivation of the diffusion element matrices and load vectors as well as diffusion flux evaluation. If KEYOPT(1) = 1, the damping matrix is diagonalized as described in Lumped Matrices (p. 391).

13.238.2. Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the concentration varies linearly along that edge.

13.239. SOLID239 - 3-D 20-Node Diffusion Solid

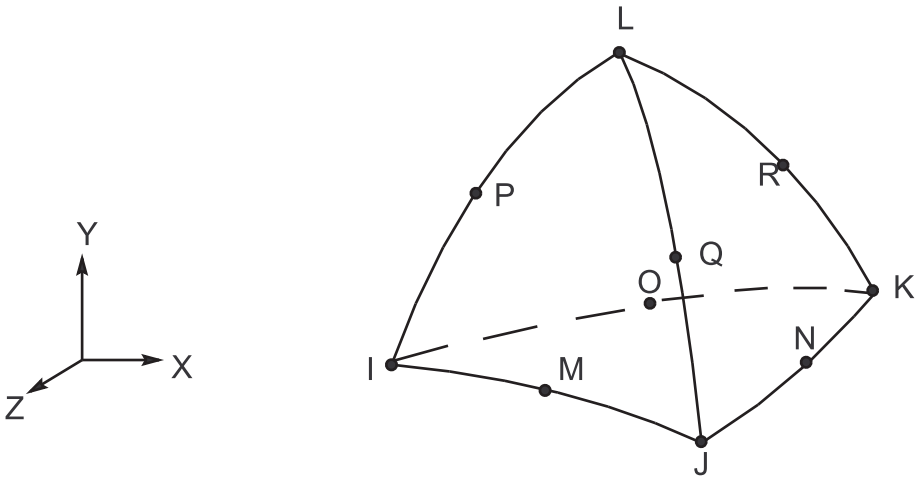


Matrix or Vector	Geometry	Shape Functions	Integration Points
Diffusivity Matrix and Diffusing Substance Generation Load Vector	Brick	Equation 11.236 (p. 355)	14
	Wedge	Equation 11.211 (p. 352)	3 x 3
	Pyramid	Equation 11.195 (p. 350)	2 x 2 x 2
	Tet	Equation 11.188 (p. 348)	4
Damping Matrix	Same as diffusivity matrix. If KEYOPT(1) = 1, the damping matrix is diagonalized as described in Lumped Matrices (p. 391).		
Diffusion Flux Load Vector	Quad	Equation 11.93 (p. 339)	3 x 3
	Triangle	Equation 11.65 (p. 337)	6

13.239.1. Other Applicable Sections

Diffusion in the *Mechanical APDL Theory Reference* describes the derivation of diffusion element matrices and load vectors as well as diffusion flux evaluation.

13.240. SOLID240 - 3-D 10-Node Tetrahedral Diffusion Solid

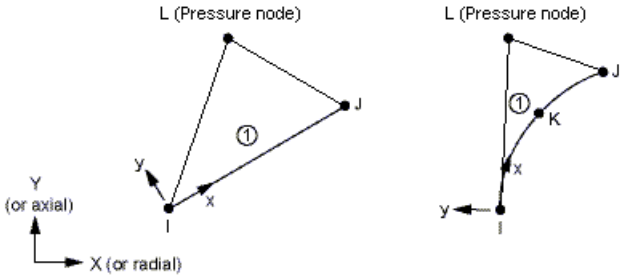


Matrix or Vector	Shape Functions	Integration Points
Diffusivity Matrix and Diffusing Substance Generation Load Vector	Equation 11.188 (p. 348)	4
Damping Matrix	Same as diffusivity matrix. If KEYOPT(1) = 1, the matrix is diagonalized as described in Lumped Matrices (p. 391)	11
Diffusion Flux Load Vector	Equation 11.188 (p. 348) specialized to the face	6

13.240.1. Other Applicable Sections

Diffusion in the *Mechanical APDL Theory Reference* describes the derivation of diffusion element matrices and load vectors as well as diffusion flux evaluations. If KEYOPT(1) = 1, the damping matrix is diagonalized as described in Lumped Matrices (p. 391).

13.241. HSFLD241 - 2-D Hydrostatic Fluid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Same as attached 2-D solid element	2 or 3

Load Type	Distribution
Fluid Mass Flow Rate	Fluid mass flow rate at pressure node
Nodal Temperature	Temperature at the pressure node

HSFLD241 is used to model fluids that are fully enclosed by solids (containing vessels). The logic for HSFLD241 is very similar to that given for HSFLD242 - 3-D Hydrostatic Fluid (p. 634), with the differences noted below:

1. The element thickness must be input on **R** command.
2. If KEYOPT(3) = 0, the element behaves as 2-D. The determination for plane stress or plane strain behavior is done based on the attached 2-D solid element.
3. If KEYOPT(3) = 1, the element behaves as axisymmetric.
4. The set of orthonormal vectors on the fluid surface for defining fluid volume is given as:

$$\mathbf{n} = \mathbf{t}_1 \times \mathbf{t}_2 \tag{13.371}$$

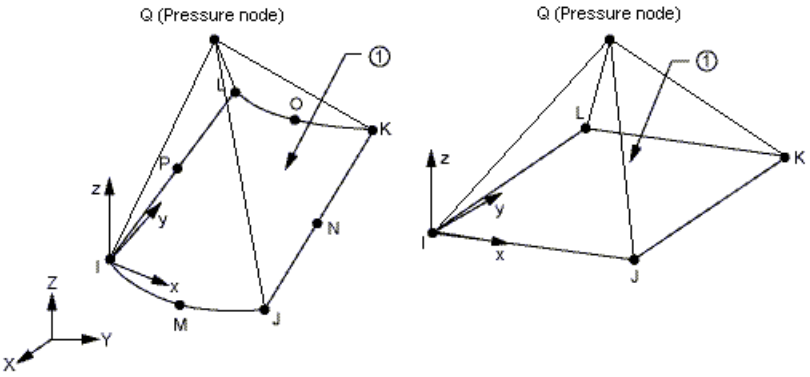
where:

$\mathbf{n}$ = outward normal vector (in X-Y plane) to the fluid surface enclosing the fluid volume
$\mathbf{t}_1 = \mathbf{k}$ = first tangent vector (normal to X-Y plane) to the fluid surface
$\mathbf{t}_2$ = second tangent vector (in X-Y plane) to the fluid surface

Since the model stays in the X-Y plane, the variation of the first tangent vector is always zero. As a result, no geometric stiffness contributions are made from the rate of change of volume enclosed by the solid surface:

$$d\delta\dot{V}_s = \int_{S_s} v_s \cdot \left[(\delta\mathbf{t}_1 \times d\mathbf{t}_2) + (d\mathbf{t}_1 \times \delta\mathbf{t}_2) \right] dS = 0 \tag{13.372}$$

13.242. HSFLD242 - 3-D Hydrostatic Fluid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Same as attached 3-D solid or shell element	2 x 2 or 3 x 3

Load Type	Distribution
Fluid Mass Flow Rate	Fluid mass flow rate at pressure node
Nodal Temperature	Temperature at the pressure node

13.242.1. Introduction

HSFLD242 is used to model fluids that are fully enclosed by solids (containing vessels). The fluid is assumed to have no viscosity, rotation or inertial effects (including acceleration loads) such as sloshing. The fluid fills the containing vessel completely so it has no free surface. Under these conditions, the fluid can be assumed to be hydrostatic; that is, the fluid pressure is uniform throughout the fluid volume. The hydrostatic fluid element is pyramid shaped with the base overlaying a 3-D solid or shell element face and the vertex at a pressure node. The element nodes shared with the underlying 3-D solid or shell element have only translation degrees of freedom. The pressure node has a single hydrostatic pressure degree of freedom (HDSP) and is shared by all hydrostatic fluid elements defining the fluid volume. The coupling between the fluid volume and the containing vessel is modeled by applying the hydrostatic fluid pressure as a surface load on the containing vessel. No fluid is assumed to flow in or out of the containing vessel unless a hydrostatic pressure value or fluid mass flow rate is prescribed at the pressure node.

13.242.2. Element Matrices and Load Vectors

Formulating a coupled system of a solid (containing vessel) enclosing a hydrostatic fluid requires augmenting the internal virtual work for the solid with contributions from the fluid. Starting with the rate of internal energy expression:

$$W' = W + \int_{S_s} t_{si} v_{si} dS + \int_{S_f} t_{fi} v_{fi} dS \tag{13.373}$$

where:

W = internal energy for the solid
S_s = current solid surface enclosing the fluid volume
S_f = current fluid surface enclosing the fluid volume
t_{si} = component i of surface traction at a point on S_s
t_{fi} = component i of surface traction at a point on S_f
v_{si} = component i of velocity at a point on S_s
v_{fi} = component i of velocity at a point on S_f

Since the fluid fills the containing vessel completely and the system is in equilibrium, the surface traction at a point on the solid surface should be equal and opposite to the surface traction at a point on the fluid surface that it is in contact with. The surface tractions represent the loads exerted by the fluid and the solid on each other. Assuming no viscosity for the fluid, the surface tractions have only normal non-zero components, so:

$$t_{si} = -t_{fi} = -P n_i \tag{13.374}$$

where:

P = fluid pressure
n_i = component i of outward normal to the fluid surface enclosing the fluid volume

Further, assuming that the fluid has no inertial effects or acceleration loads, Equation 8.2 (p. 253), Equation 8.3 (p. 253), and Equation 8.4 (p. 254) for fluid momentum show that the fluid pressure should be uniform. Using this fact with Equation 13.374 (p. 635) in Equation 13.373 (p. 635) gives:

$$W' = W - P \int_{S_s} n_i v_{si} dS + P \int_{S_f} n_i v_{fi} dS \tag{13.375}$$

By taking a variation of Equation 13.375 (p. 635) and then using Gauss' theorem, we get the internal virtual work expression:

$$\delta W' = \delta W - P \int_{S_s} n_i \delta v_{si} dS - \delta P \left(\int_{S_s} n_i v_{si} dS - \int_{V_f} v_{fi,i} dV \right) \tag{13.376}$$

where:

V_f = current fluid volume
$v_{fi,i}$ = div $\mathbf{v}_f$ = divergence of velocity vector at a point in V_f

The second term on the right hand side in Equation 13.376 (p. 636) is the virtual work done by the hydrostatic fluid pressure on the solid that contains the fluid and represents the coupling between the fluid volume and the solid (containing vessel).

Since the fluid volume has uniform pressure, the volumetric strain is also uniform. This allows Equation 13.376 (p. 636) to be simplified further.

Using the definitions of deformation gradient and its determinant as given in Equation 3.2 (p. 30), Equation 3.3 (p. 30), and Equation 3.4 (p. 30), and the derivative of the determinant (Gurtin [381] (p. 942)), it can be seen that:

$$\dot{V}_f \overset{\text{def}}{=\!=} \frac{dV_f}{dt} = \int_{V_{0f}} \frac{d(\det[F])}{dt} dV = \int_{V_f} v_{fi,i} dV \tag{13.377}$$

Similarly, for the solid surface enclosing the fluid volume:

$$\dot{V}_s \overset{\text{def}}{=\!=} \frac{dV_s}{dt} = \int_{S_s} n_i v_{si} dS \tag{13.378}$$

where:

V_s = current volume enclosed by solid surface S_s

Using Equation 13.377 (p. 636) and Equation 13.378 (p. 636) (and its variation) in Equation 13.376 (p. 636) gives:

$$\delta W' = \delta W - P \delta \dot{V}_s - \delta P \left(\dot{V}_s - \dot{V}_f \right) \tag{13.379}$$

The internal virtual work in Equation 13.379 (p. 636) is valid for large as well as small strains. Since the fluid is assumed to have no acceleration loads or free surface, there are no contributions to external virtual work from loads. However, if fluid mass flow rate is prescribed at the pressure node, then the mass, and hence the volume, of the fluid will change and the virtual work expression will be:

$$\delta W' = \delta W - P \delta \dot{V}_s - \delta P \left(\dot{V}_s - \dot{V}_f - \frac{w}{\rho_f} \right) \tag{13.380}$$

where:

w = fluid mass flow rate (use **F** command)
ρ_f = current fluid density

The stiffness matrix for the coupled system can be obtained by taking the derivative of the virtual work expression as given in Equation 13.380 (p. 636):

$$D\delta W' = D\delta W - DP\delta \dot{V}_s - PD\delta \dot{V}_s - \delta P\left(D\dot{V}_s - D\dot{V}_f + \frac{w}{(\rho_f)^2} D\rho_f \right) \tag{13.381}$$

For completeness, the variational quantities in Equation 13.381 (p. 637) need to be expressed in terms of the independent variables. To compute $\delta \dot{V}_s$, we start by taking a variation of Equation 13.378 (p. 636):

$$\delta \dot{V}_s = \delta \int_{S_s} n_i v_{si} dS = \int_{S_s} n_i \delta v_{si} dS \tag{13.382}$$

Since the fluid is assumed to be irrotational, there is no contribution from the term containing the variation of the normal, δn_i.

For $D\delta \dot{V}_s$, we start with Equation 13.382 (p. 637), including the term containing the variation of the normal:

$$D\delta \dot{V}_s = D\left(\int_{S_s} n_i \delta v_{si} dS + \int_{S_s} \delta n_i v_{si} dS \right) = \int_{S_s} D\delta n_i v_{si} dS \tag{13.383}$$

Again, irrotational fluid assumption eliminates terms with variation of the normal. If $\mathbf{t}_1$ and $\mathbf{t}_2$ define orthonormal tangent vectors on the surface such that:

$$\mathbf{n} = \mathbf{t}_1 \times \mathbf{t}_2 \tag{13.384}$$

then, Equation 13.383 (p. 637) can be expanded as:

$$D\delta \dot{V}_s = \int_{S_s} \mathbf{v}_s \cdot \left[(\delta \mathbf{t}_1 \times D\mathbf{t}_2) + (D\mathbf{t}_1 \times \delta \mathbf{t}_2) \right] dS \tag{13.385}$$

The terms $D\dot{V}_f$ and $D\rho_f$ in Equation 13.381 (p. 637) define material stiffness for the fluid volume. This requires the definition of the pressure-volume relationship as described below.

13.242.2.1. Incompressible Fluid

The incompressibility constraint for the fluid volume can be given by using Equation 13.377 (p. 636) as:

$$\dot{V}_f = \int_{V_f} v_{fi,i} dV = 0 \tag{13.386}$$

The augmented virtual work expression incorporating the incompressibility constraint can be written as:

$$\delta W' = \delta W - \lambda \delta \dot{V}_s - \delta\lambda \left(\dot{V}_s - \frac{w}{\rho_f} \right) \tag{13.387}$$

where:

λ = Lagrange multiplier

Comparing Equation 13.387 (p. 637) to Equation 13.380 (p. 636) shows that they are equivalent with the hydrostatic fluid pressure P acting as a Lagrange multiplier. The stiffness contribution can be obtained by taking the derivative of the augmented virtual work:

$$D\delta W' = D\delta W - D\lambda\delta\dot{V}_s - \lambda D\delta\dot{V}_s - \delta\lambda D\dot{V}_s \tag{13.388}$$

The incompressible fluid may have a change in volume if it has non-zero thermal expansion. Since temperature is not defined as a degree of freedom, changes in volume due to temperature changes are computed by specifying temperature loading (use the **BF** command) and coefficient of thermal expansion (use the **MP** command):

$$\dot{V}_f = 3V_{0f}\alpha\dot{T} \tag{13.389}$$

where:

α = coefficient of thermal expansion (use **MP** command)

$\dot{T}$ = rate of change of temperature

V_{0f} = initial fluid volume

13.242.2.2. Liquid

Using the constitutive equation for liquid (see Fluid Material Models (p. 174)), the rate of change of volume and its variation can be given as:

$$\dot{V}_f = -\frac{V_{0f}}{K}\dot{P} + 3V_{0f}\alpha\dot{T} \tag{13.390}$$

$$D\dot{V}_f = -\frac{V_{0f}}{K}D\dot{P} \tag{13.391}$$

where:

K = bulk modulus (use **TB** command with *Lab* = FLUID and *TB0PT* = LIQUID)

The variation of current density can be defined as:

$$D\rho_f = \frac{\rho_f}{\rho_{0f}}\frac{\rho_f}{K}DP \tag{13.392}$$

13.242.2.3. Gas

To define the rate of change of volume, we need to look at incremental change in volume:

$$\dot{V}_f = \frac{1}{\Delta t}(V_f - V_{nf}) \tag{13.393}$$

where:

Δt = time increment for current substep

V_{nf} = fluid volume at the end of previous substep

Using the Ideal Gas Law, the previous expression can be expanded as:

$$\dot{V}_f = \frac{1}{\Delta t}\left(V_f - V_f\frac{P_t}{P_{nt}}\frac{T_{nt}}{T_t}\right) = -\frac{V_f}{P_{nt}}\dot{P} + \frac{V_{nf}}{T_{nt}}\dot{T} \tag{13.394}$$

where:

$P_t = P_{ref} + P$ = total fluid pressure
P_{ref} = reference pressure (specified as a real constant with **R** command)
$T_t = T_{off} + T$ = total temperature
T_{off} = temperature offset from absolute zero to zero (use **TOFFST** command)
P_{nt} = total fluid pressure at the end of previous substep
T_{nt} = total temperature at the end of previous substep

The variation of the rate of change of volume can be easily obtained as:

$$D\dot{V}_f = -\frac{V_f}{P_{nt}} D\dot{P} \tag{13.395}$$

and the variation of current density can be given as:

$$D\rho_f = \frac{\rho_f}{P_t} DP \tag{13.396}$$

13.242.2.4. Pressure-Volume Data

For compressible fluids that do not follow the constitutive equation for liquid or the Ideal Gas Law, the rate of change of volume can be expressed as:

$$\dot{V}_f = \frac{\Delta V_f}{\Delta P}\dot{P} + \frac{\Delta V_f}{\Delta T}\dot{T} \tag{13.397}$$

where:

$\frac{\Delta V_f}{\Delta P}$ = slope (interpolated) of the pressure-volume data curve
$\frac{\Delta V_f}{\Delta T}$ = slope (interpolated) of the temperature-volume data curve

and the variation of the rate of change of volume can be expressed as:

$$D\dot{V}_f = \frac{\Delta V_f}{\Delta P} D\dot{P} \tag{13.398}$$

The variation of current density is defined as:

$$D\rho_f = -\frac{\rho_f}{V_f}\frac{\Delta V_f}{\Delta P} DP \tag{13.399}$$

See Fluid Material Models (p. 174) for more details.

13.243. Reserved for Future Use

This section is reserved for future use.

13.244. Reserved for Future Use

This section is reserved for future use.

13.245. Reserved for Future Use

This section is reserved for future use.

13.246. Reserved for Future Use

This section is reserved for future use.

13.247. Reserved for Future Use

This section is reserved for future use.

13.248. Reserved for Future Use

This section is reserved for future use.

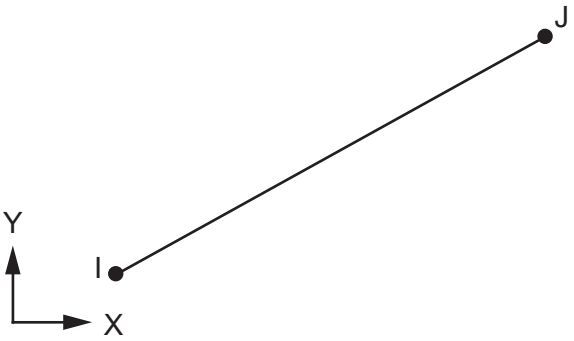
13.249. Reserved for Future Use

This section is reserved for future use.

13.250. Reserved for Future Use

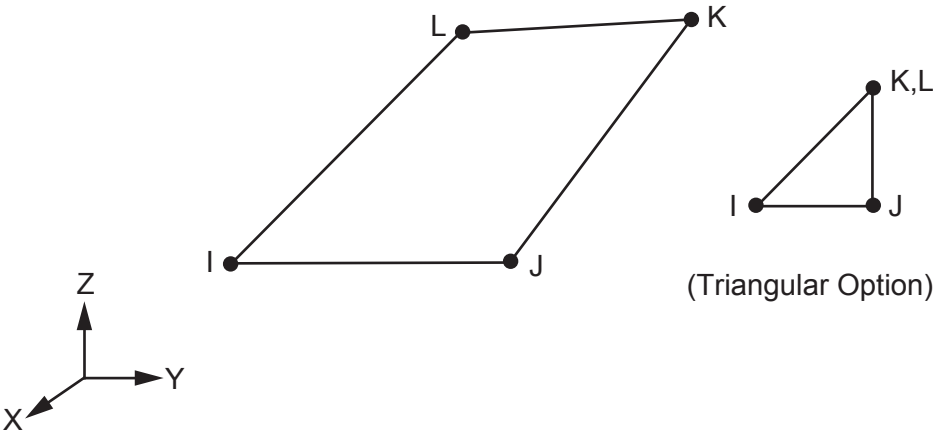
This section is reserved for future use.

13.251. SURF251 - 2-D Radiosity Surface



SURF251 is used only for postprocessing of radiation quantities, such as radiation heat flux. See SURF251 in the *Element Reference* for details.

13.252. SURF252 - 3-D Thermal Radiosity Surface



SURF252 is used only for postprocessing of radiation quantities, such as radiation heat flux. See SURF252 in the *Element Reference* for details.

13.253. Reserved for Future Use

This section is reserved for future use.

13.254. Reserved for Future Use

This section is reserved for future use.

13.255. Reserved for Future Use

This section is reserved for future use.

13.256. Reserved for Future Use

This section is reserved for future use.

13.257. Reserved for Future Use

This section is reserved for future use.

13.258. Reserved for Future Use

This section is reserved for future use.

13.259. Reserved for Future Use

This section is reserved for future use.

13.260. Reserved for Future Use

This section is reserved for future use.

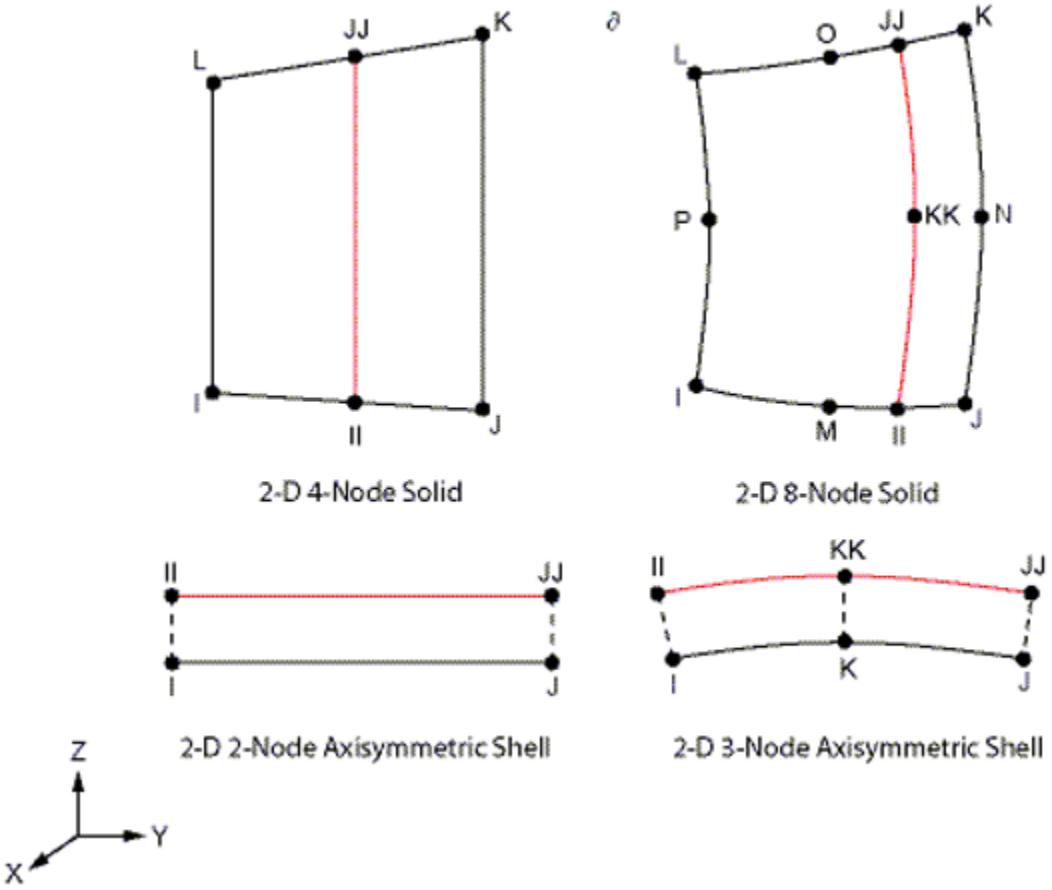
13.261. Reserved for Future Use

This section is reserved for future use.

13.262. Reserved for Future Use

This section is reserved for future use.

13.263. REINF263 - 2-D Smeared Reinforcing



Matrix or Vector	Base Element	Shape Functions (for each layer)	Integration Points
Stiffness and Stress Stiffness Matrices and Thermal Load Vector	Linear 2-D solid or shell	Equation 11.6 (p. 329), Equation 11.7 (p. 330), and Equation 11.8 (p. 330)	Across the length: 1 Across the section: 1
	Quadratic 3-D solid or shell	Equation 11.19 (p. 330), Equation 11.20 (p. 330), and Equation 11.21 (p. 330)	Across the length: 2 Across the section: 1

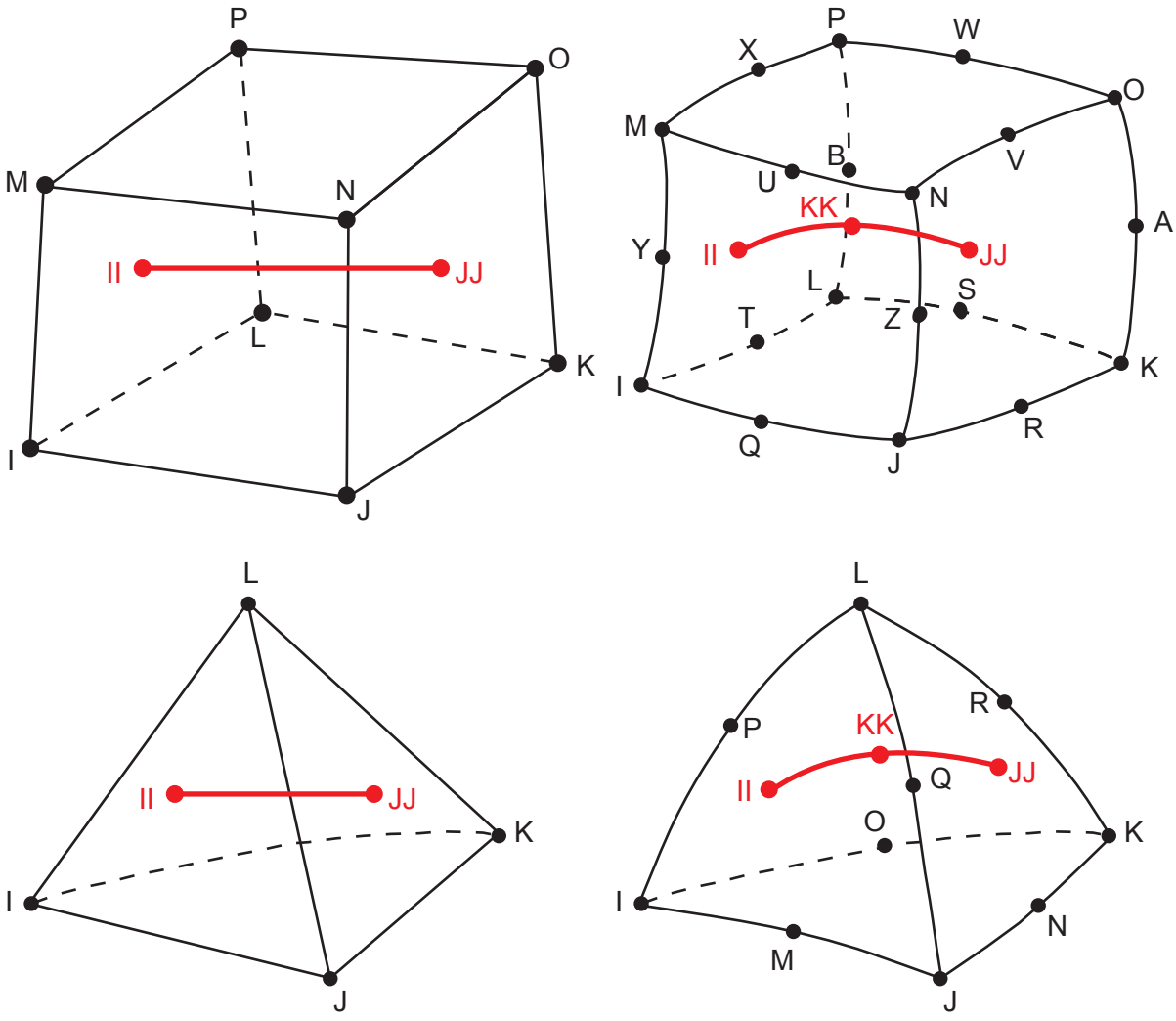
Matrix or Vector	Base Element	Shape Functions (for each layer)	Integration Points
Mass Matrix	Same as stiffness matrix.		Same as stiffness matrix.
Pressure Load Vector	N/A		N/A

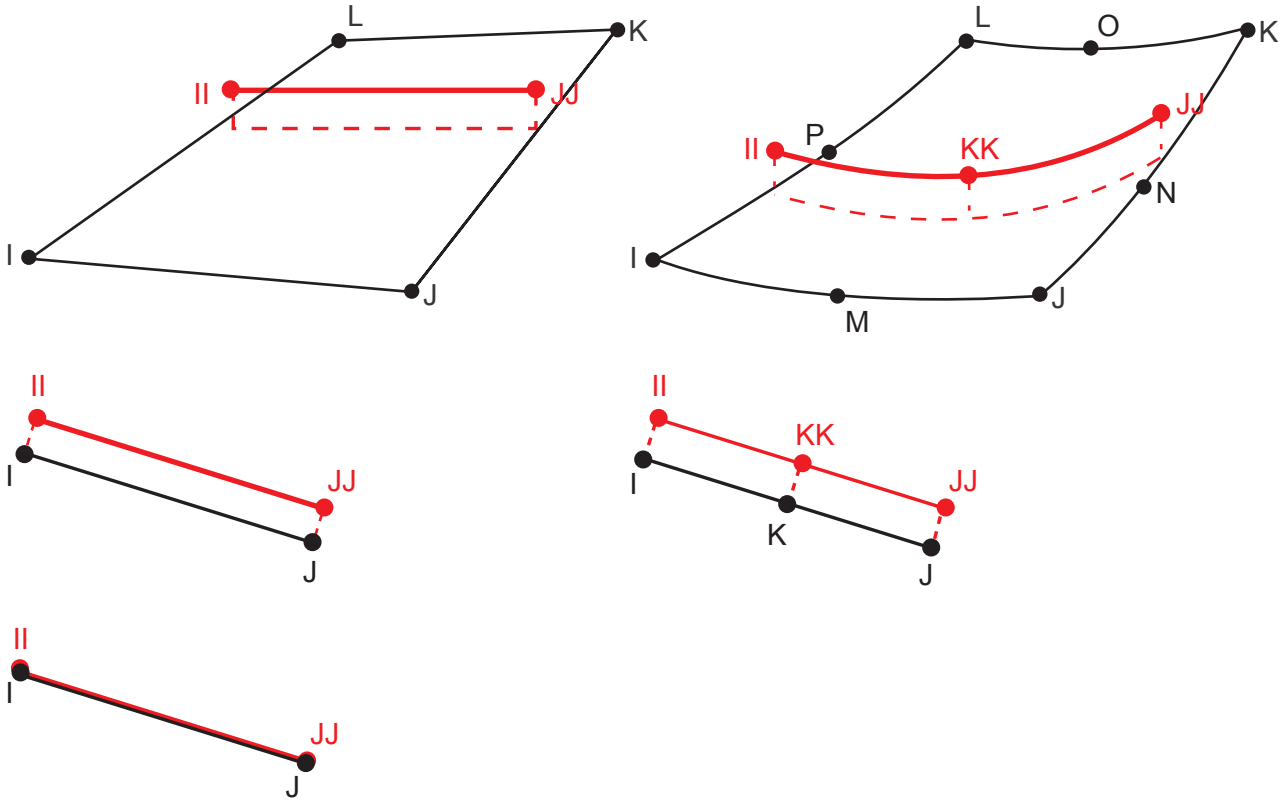
Load Type	Distribution
Element Temperature	Linear along the length, constant across the section.
Nodal Temperature	N/A
Pressure	N/A

13.263.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element. See Stiffness and Mass Matrices of a Reinforcing Layer (p. 646) for the general formulation of the reinforcing stiffness and mass matrices.

13.264. REINF264 - 3-D Discrete Reinforcing





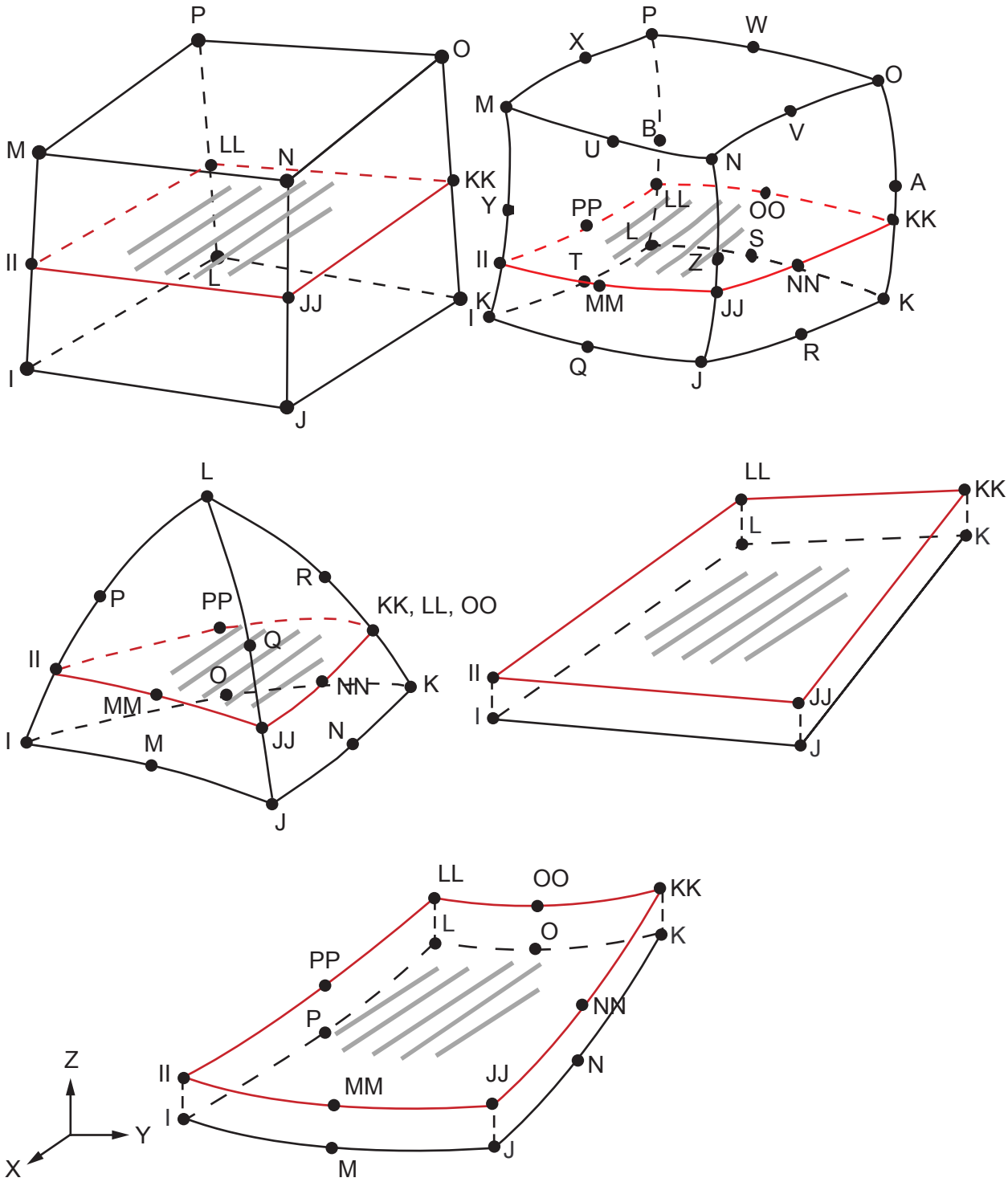
Matrix or Vector	Base Element	Shape Functions (for each layer)	Integration Points
Stiffness and Stress Stiffness Matrices and Thermal Load Vector	Linear 3-D spar, beam, solid, or shell	Equation 11.6 (p. 329), Equation 11.7 (p. 330), and Equation 11.8 (p. 330)	Across the length: 1 Across the section: 1
	Quadratic 3-D beam, solid, or shell	Equation 11.19 (p. 330), Equation 11.20 (p. 330), and Equation 11.21 (p. 330)	Across the length: 2 Across the section: 1
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	N/A		N/A

Load Type	Distribution
Element Temperature	Linear along the length, constant across the section.
Nodal Temperature	N/A
Pressure	N/A

13.264.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element. See Stiffness and Mass Matrices of a Reinforcing Layer (p. 646) for the general formulation of the reinforcing stiffness and mass matrices.

13.265. REINF265 - 3-D Smeared Reinforcing



Matrix or Vector	Base Element	Shape Functions (for each layer)	Integration Points
Stiffness and Stress Stiffness Matrices and Thermal Load Vector	Linear 3-D solid or shell	Equation 11.69 (p. 337), Equation 11.70 (p. 337), and Equation 11.71 (p. 337)	In-plane: 1 x 1 Thru-the-thickness: 1

Matrix or Vector	Base Element	Shape Functions (for each layer)	Integration Points
	Quadratic 3-D solid or shell	Equation 11.84 (p. 338), Equation 11.85 (p. 338), and Equation 11.86 (p. 338)	In-plane: 2 x 2 Thru-the-thickness: 1
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	N/A		N/A

Load Type	Distribution
Element Temperature	Bilinear in plane of each reinforcing layer, constant thru-the-thickness of each layer.
Nodal Temperature	N/A
Pressure	N/A

13.265.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element.

13.265.2. Stiffness and Mass Matrices of a Reinforcing Layer

Each layer of reinforcing fibers is simplified as a membrane with unidirectional stiffness. The equivalent membrane thickness h is given by:

$$h = A / S \tag{13.400}$$

where:

A = cross-section area of each fiber (input on **SECDATA** command)
S = distance between two adjacent fibers (input on **SECDATA** command)

We assume that the reinforcing fibers are firmly attached to the base element (that is, no relative movement between the base element and the fibers is allowed). Therefore, the degrees of freedom (DOF) of internal layer nodes (II, JJ, KK, LL, etc.) can be expressed in terms of DOFs of the external element nodes (I, J, K, L, etc.). Taking a linear 3-D solid base element as the example, the DOFs of an internal layer node II can be shown as:

$$\begin{Bmatrix} u_{II} \\ v_{II} \\ w_{II} \end{Bmatrix} = \sum_{i=1}^{8} N_i(\xi_{II}, \eta_{II}, \zeta_{II}) \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \tag{13.401}$$

where:

$\{u_{II}, v_{II}, w_{II}\}$ = displacements of internal layer node II
$\{u_i, v_i, w_i\}$ = displacements of base element node i
$N_i(\xi_{II}, \eta_{II}, \zeta_{II})$ = value of trilinear shape function of node i at the location of internal node II

Similar relationships can be established for other type of base elements. The stiffness and mass matrices of each reinforcing layer are first evaluated with respect to internal layer DOFs. The equivalent stiffness and mass contributions of this layer to the element is then determined through relationship (Equation 13.401 (p. 646)).

13.266. Reserved for Future Use

This section is reserved for future use.

13.267. Reserved for Future Use

This section is reserved for future use.

13.268. Reserved for Future Use

This section is reserved for future use.

13.269. Reserved for Future Use

This section is reserved for future use.

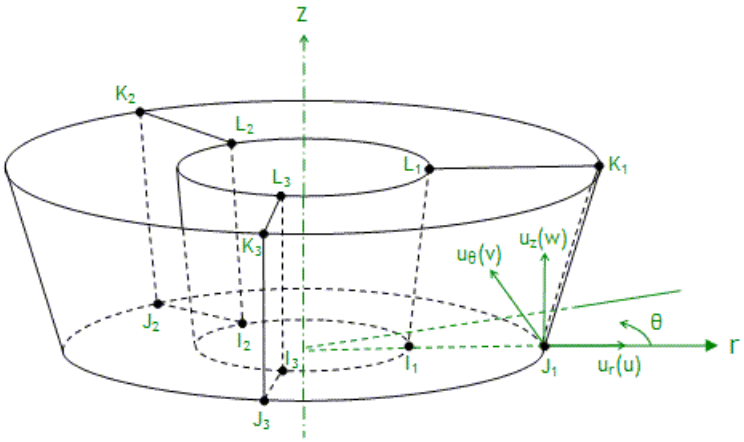
13.270. Reserved for Future Use

This section is reserved for future use.

13.271. Reserved for Future Use

This section is reserved for future use.

13.272. SOLID272 - General Axisymmetric Solid with 4 Base Nodes



Matrix or Vector	Geometry	Shape Functions	Integration Points*
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Quad	Equation 11.255 (p. 359), Equation 11.256 (p. 359), Equation 11.258 (p. 360),	2 x 2 x N_c

Matrix or Vector	Geo-metry	Shape Functions	Integration Points*
		Equation 11.259 (p. 360), and Equation 11.260 (p. 360)	
	Triangle	Equation 11.255 (p. 359), Equation 11.256 (p. 359), Equation 11.261 (p. 360), Equation 11.262 (p. 360), and Equation 11.263 (p. 360)	1 x N_c
Mass Matrix	Quad	Same as stiffness matrix	2 x 2 x N_c
	Triangle		1 x N_c
Pressure Load Vector	Same as stiffness matrix, specialized to face		2 x N_c

* N_c = the number of integration points in the circumferential direction. The N_c integration points are circumferentially located at:

- the nodal planes, and
- midway between the nodal planes (that is, at the integration planes)

so that $N_c = (2 * N_{np})$, where N_{np} = number of nodal planes (KEYOPT(2)). Exception: If KEYOPT(2) = 1, then $N_c = 1$.

Load Type	Distribution
Element Temperature	Bilinear across element on rz plane, linear in circumferential direction
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

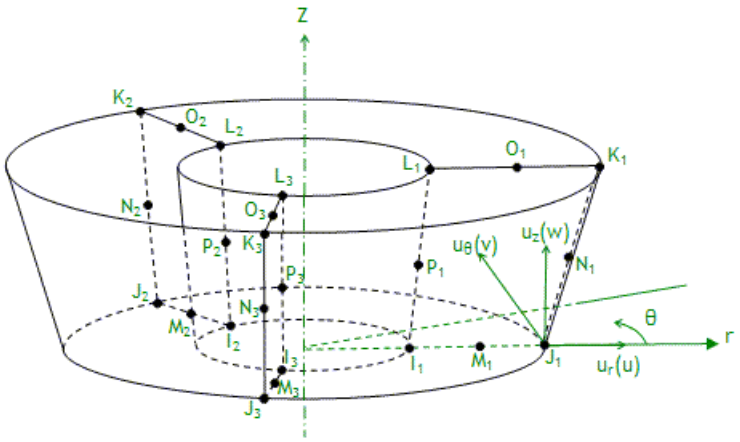
13.272.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations.

13.272.2. Assumptions and Restrictions

Although the elements are initially axisymmetric, the loads and deformation can be general in non-axisymmetric 3-D. The displacements are interpolated in elemental coordinate system by interpolation functions, but the user can define the nodal displacements in any direction.

13.273. SOLID273 - General Axisymmetric Solid with 8 Base Nodes



Matrix or Vector	Geo-metry	Shape Functions	Integration Points*
Stiffness and Stress Stiffness Matrices; and Thermal Load Vector	Quad	Equation 11.255 (p. 359), Equation 11.256 (p. 359), Equation 11.264 (p. 361), Equation 11.265 (p. 361), and Equation 11.266 (p. 361)	2 x 2 x N_c
	Triangle	Equation 11.255 (p. 359), Equation 11.256 (p. 359), Equation 11.267 (p. 361), Equation 11.268 (p. 361), and Equation 11.269 (p. 361)	3 x N_c
Mass Matrix	Quad	Same as stiffness matrix	3 x 3 x N_c
	Triangle		3 x N_c
Pressure Load Vector	Same as stiffness matrix, specialized to face		2 x N_c

* N_c = the number of integration points in the circumferential direction. The N_c integration points are circumferentially located at:

- the nodal planes, and
- midway between the nodal planes (that is, at the integration planes)

so that $N_c = (2 * N_{np})$, where N_{np} = number of nodal planes (KEYOPT(2)). Exception: If KEYOPT(2) = 1, then $N_c = 1$.

Load Type	Distribution
Element Temperature	Biquadratic across element on rz plane and linear between nodal planes in the circumferential direction
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

13.273.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element.

13.273.2. Assumptions and Restrictions

Although the elements are initially axisymmetric, the loads and deformation can be general in non-axisymmetric 3-D. The displacements are interpolated in elemental coordinate system by interpolation functions, but the user can define the nodal displacements in any direction.

13.274. Reserved for Future Use

This section is reserved for future use.

13.275. Reserved for Future Use

This section is reserved for future use.

13.276. Reserved for Future Use

This section is reserved for future use.

13.277. Reserved for Future Use

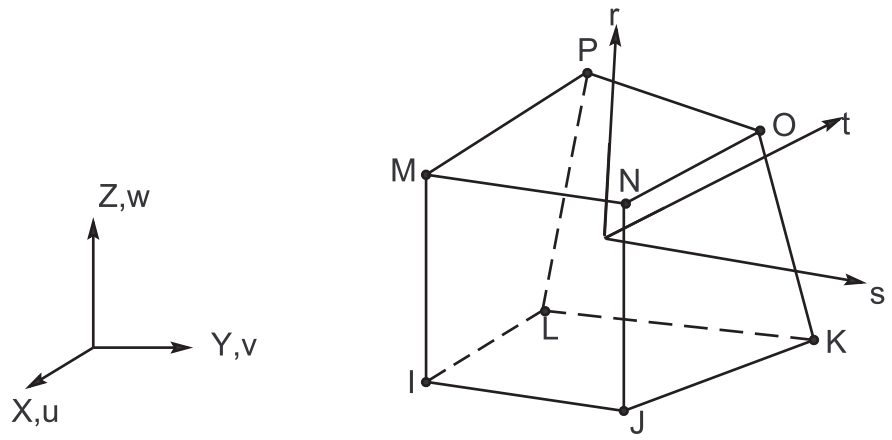
This section is reserved for future use.

13.278. SOLID278 - 3-D 8-Node homogeneous/Layered Thermal Solid

SOLID278 is available in two forms:

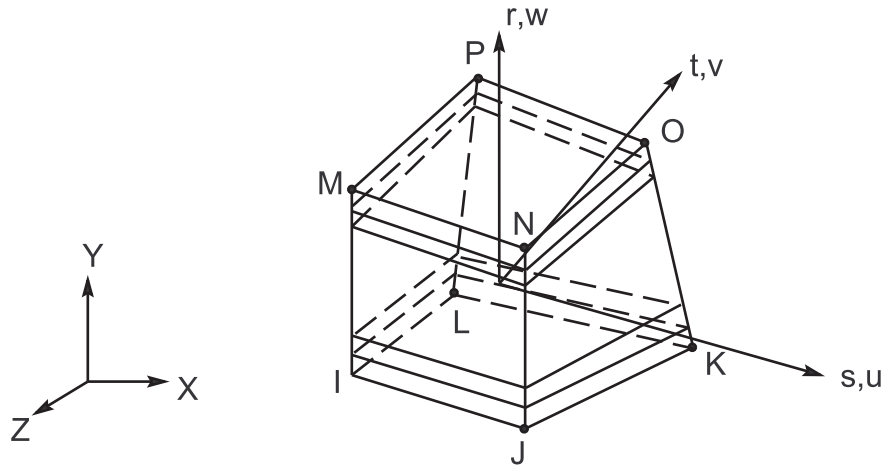
- homogeneous (nonlayered) thermal solid (KEYOPT(3) = 0, the default) - see SOLID278 - 3-D 8-Node homogeneous Thermal Solid (p. 651).
- Layered thermal solid (KEYOPT(3) = 1) - see SOLID278 - 3-D 8-Node Layered Thermal Solid (p. 651).

13.278.1. SOLID278 - 3-D 8-Node homogeneous Thermal Solid



Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Equation 11.222 (p. 353)	2 x 2 x 2
Specific Heat Matrix	Equation 11.222 (p. 353). Matrix is diagonalized as described in Lumped Matrices (p. 391)	Same as conductivity matrix
Convection Surface Matrix and Load Vector	Equation 11.222 (p. 353) specialized to the face	2 x 2

13.278.2. SOLID278 - 3-D 8-Node Layered Thermal Solid



Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Equation 11.222 (p. 353)	In-plane: 2 x 2 Thru-the-thickness:

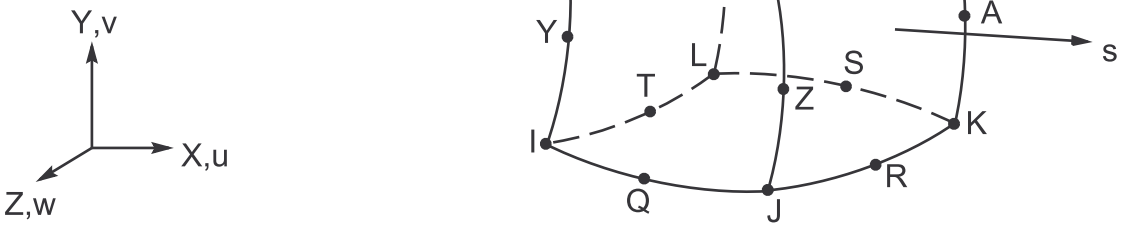
Matrix or Vector	Shape Functions	Integration Points
		1, 3, 5, 7, or 9 per layer Defaults to 3
Specific Heat Matrix	Equation 11.222 (p. 353). Matrix is diagonalized as described in Lumped Matrices (p. 391)	Same as conductivity matrix
Convection Surface Matrix and Load Vector	Equation 11.222 (p. 353) specialized to the face	2 x 2

13.279. SOLID279 - 3-D 20-Node homogeneous/Layered Thermal Solid

SOLID279 is available in two forms:

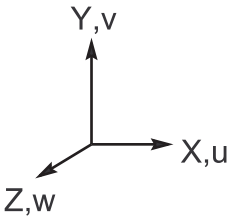
- homogeneous (nonlayered) thermal solid (KEYOPT(3) = 0, the default) - see SOLID279 - 3-D 20-Node homogeneous Thermal Solid (p. 652).
- Layered thermal solid (KEYOPT(3) = 1) - see SOLID279 - 3-D 20-Node Layered Thermal Solid (p. 653).

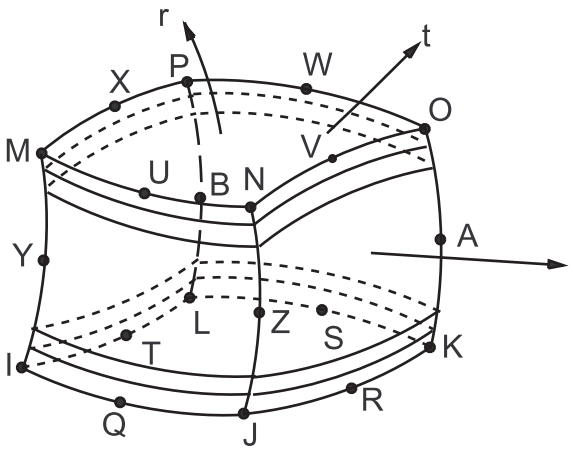
13.279.1. SOLID279 - 3-D 20-Node homogeneous Thermal Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Brick	Equation 11.233 (p. 355)	14
	Wedge	Equation 11.209 (p. 352)	3 x 3
	Pyramid	Equation 11.193 (p. 350)	2 x 2 x 2
	Tet	Equation 11.185 (p. 348)	4
Specific Heat Matrix	Same as conductivity matrix.		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Quad	Equation 11.91 (p. 338)	3 x 3
	Triangle	Equation 11.63 (p. 337)	6

13.279.2. SOLID279 - 3-D 20-Node Layered Thermal Solid



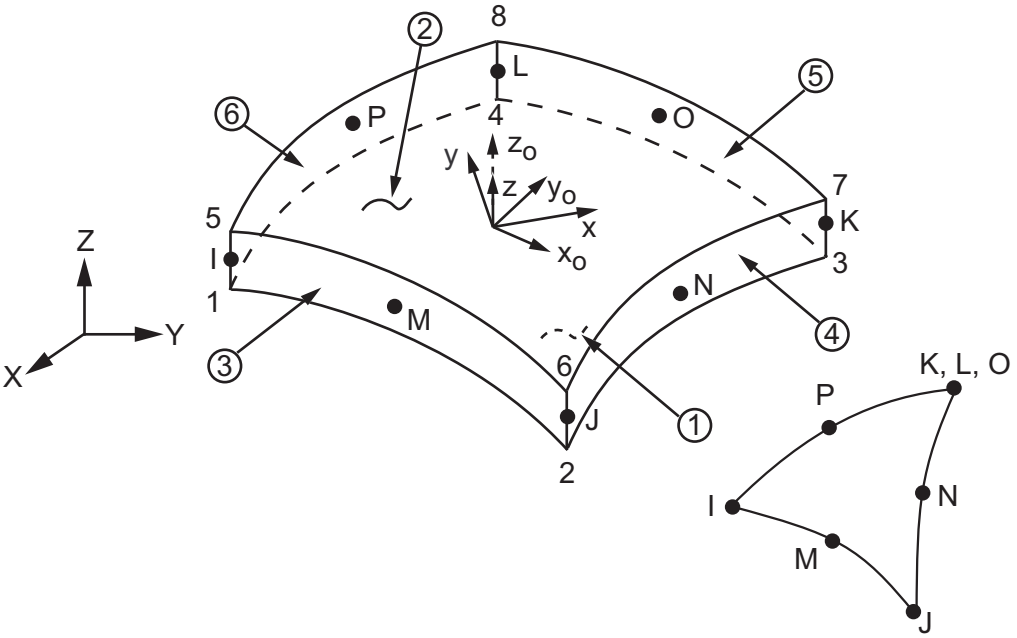


Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix and Heat Generation Load Vector	Brick	Equation 11.233 (p. 355)	In-plane: 2 x 2 Thru-the-thickness: 1, 3, 5, 7, or 9 per layer Defaults to 3
	Wedge	Equation 11.209 (p. 352)	3 x 3
	Pyramid	Equation 11.193 (p. 350)	2 x 2 x 2
	Tet	Equation 11.185 (p. 348)	4
Specific Heat Matrix	Same as conductivity matrix.		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Quad	Equation 11.91 (p. 338)	3 x 3
	Triangle	Equation 11.63 (p. 337)	6

13.280. Reserved for Future Use

This section is reserved for future use.

13.281. SHELL281 - 8-Node Shell



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix and Thermal Load Vector	Quad	Equation 11.84 (p. 338), Equation 11.85 (p. 338), Equation 11.86 (p. 338), Equation 11.87 (p. 338), Equation 11.88 (p. 338), and Equation 11.89 (p. 338)	In-plane: 2 x 2 Thru-the-thickness: 1, 3, 5, 7, or 9 per layer for section data input for general shell option (KEYOPT(1) = 0) Thru-the-thickness: 1 per layer for section data input for membrane shell option (KEYOPT(1) = 1)
	Triangle	Equation 11.57 (p. 336), Equation 11.58 (p. 336), Equation 11.59 (p. 336), Equation 11.60 (p. 336), Equation 11.61 (p. 336), and Equation 11.62 (p. 337)	In-plane: 3 1, 3, 5, 7, or 9 per layer for section data input for general shell option (KEYOPT(1) = 0) 1 per layer for section data input for membrane shell option (KEYOPT(1) = 1)
Consistent Mass and Stress Stiffness Matrices	Quad	Equation 11.84 (p. 338), Equation 11.85 (p. 338), Equation 11.86 (p. 338), Equa-	Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
		tion 11.87 (p. 338), Equation 11.88 (p. 338), and Equation 11.89 (p. 338)	
	Triangle	Equation 11.57 (p. 336), Equation 11.58 (p. 336), Equation 11.59 (p. 336), Equation 11.60 (p. 336), Equation 11.61 (p. 336), and Equation 11.62 (p. 337)	Same as stiffness matrix
Lumped Mass Matrix	Quad	Equation 11.84 (p. 338), Equation 11.85 (p. 338), and Equation 11.86 (p. 338)	Same as stiffness matrix
	Triangle	Equation 11.57 (p. 336), Equation 11.58 (p. 336), and Equation 11.59 (p. 336)	Same as stiffness matrix
Transverse Pressure Load Vector	Quad	Equation 11.86 (p. 338)	2 x 2
	Triangle	Equation 11.59 (p. 336)	3
Edge Pressure Load Vector	Same as in-plane mass matrix, specialized to the edge		2

Load Type	Distribution
Element Temperature	Linear thru thickness, bilinear in plane of element
Nodal Temperature	Constant thru thickness, bilinear in plane of element
Pressure	Bilinear in plane of element, linear along each edge

References: Ahmad([1] (p. 921)), Cook([5] (p. 921)), Dvorkin([96] (p. 926)), Dvorkin([97] (p. 926)), Bathe and Dvorkin([98] (p. 926)), Allman([113] (p. 927)), Cook([114] (p. 927)), MacNeal and Harder([115] (p. 927))

13.281.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations.

13.281.2. Assumptions and Restrictions

Normals to the centerplane are assumed to remain straight after deformation, but not necessarily normal to the centerplane.

Each set of integration points thru a layer (in the r direction) is assumed to have the same element (material) orientation.

13.281.3. Membrane Option

A membrane option is available for SHELL281 if KEYOPT(1) = 1. For this option, there is no bending stiffness or rotational degrees of freedom. There is only one integration point per layer, regardless of other input.

13.281.4. Shear Correction

The element uses an equivalent energy method to compute shear-correction factors. These factors are predetermined based on the section lay-up at the start of solution.

13.282. Reserved for Future Use

This section is reserved for future use.

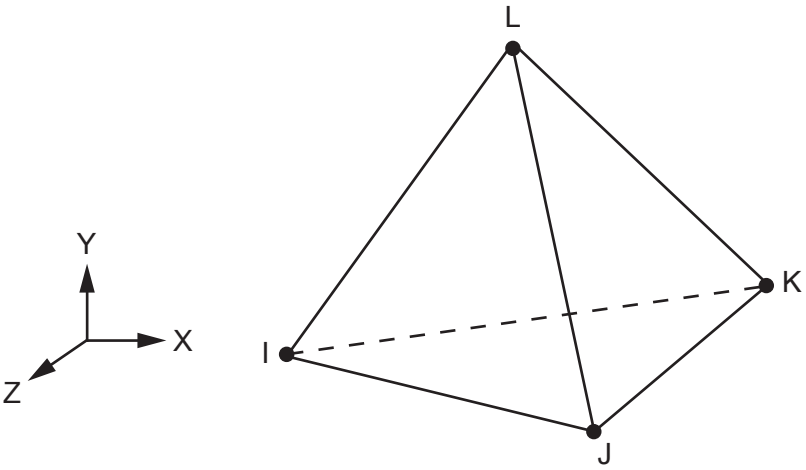
13.283. Reserved for Future Use

This section is reserved for future use.

13.284. Reserved for Future Use

This section is reserved for future use.

13.285. SOLID285 - 3-D 4-Node Tetrahedral Structural Solid with Nodal Pressures



Matrix or Vector	Shape Functions	Integration Points
Stiffness, Mass, and Stress Stiffness Matrices; and Thermal Load Vector	Equation 11.170 (p. 347) through Equation 11.181 (p. 348)	4
Pressure Load Vector	Equation 11.49 (p. 336) and Equation 11.50 (p. 336)	3

Load Type	Distribution
Element Temperature	Same as shape functions
Nodal Temperature	Same as shape functions
Pressure	Linear over each face

13.285.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations. General Element Formulations (p. 50) gives the general element formulations used by this element.

13.285.2. Theory

Stabilization terms are introduced and condensed out at element as enhanced term (Onate et al.([91] (p. 925))).

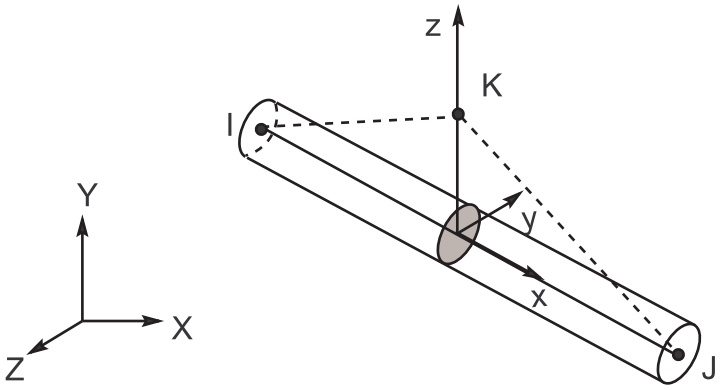
13.286. Reserved for Future Use

This section is reserved for future use.

13.287. Reserved for Future Use

This section is reserved for future use.

13.288. PIPE288 - 3-D 2-Node Pipe



Matrix or Vector	Option	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal and Newton-Raphson Load Vectors	Linear (KEYOPT(3)=0)	Equation 11.6 (p. 329), Equation 11.7 (p. 330), Equation 11.8 (p. 330), Equation 11.9 (p. 330), Equation 11.10 (p. 330), and Equation 11.11 (p. 330)	Along the length: 1 Across the section: see text below
	Quadratic (KEYOPT(3)=2)	Equation 11.19 (p. 330), Equation 11.20 (p. 330), Equation 11.21 (p. 330), Equation 11.22 (p. 331), Equation 11.23 (p. 331), and Equation 11.24 (p. 331)	Along the length: 2 Across the section: see text below

Matrix or Vector	Option	Shape Functions	Integration Points
	Cubic (KEYOPT(3)=3)	Equation 11.26 (p. 331), Equation 11.27 (p. 331), Equation 11.28 (p. 331), Equation 11.29 (p. 331), Equation 11.30 (p. 331), and Equation 11.31 (p. 331)	Along the length: 3 Across the section: see text below
Consistent Mass Matrix and Pressure Load Vector	Linear (KEYOPT(3)=0)	Equation 11.6 (p. 329), Equation 11.7 (p. 330), Equation 11.8 (p. 330), Equation 11.9 (p. 330), Equation 11.10 (p. 330), and Equation 11.11 (p. 330)	Along the length: 2 Across the section: 1
	Quadratic (KEYOPT(3)=2)	Equation 11.19 (p. 330), Equation 11.20 (p. 330), Equation 11.21 (p. 330), Equation 11.22 (p. 331), Equation 11.23 (p. 331), and Equation 11.24 (p. 331)	Along the length : 3 Across the section: 1
	Cubic (KEYOPT(3)=3)	Equation 11.26 (p. 331), Equation 11.27 (p. 331), Equation 11.28 (p. 331), Equation 11.29 (p. 331), Equation 11.30 (p. 331), and Equation 11.31 (p. 331)	Along the length: 4 Across the section: 1
Lumped Mass Matrix	Linear (KEYOPT(3) = 0)	Equation 11.6 (p. 329), Equation 11.7 (p. 330), and Equation 11.8 (p. 330)	Along the length: 2 Across the section: 1
	Quadratic (KEYOPT(3) = 2)	Equation 11.19 (p. 330), Equation 11.20 (p. 330), and Equation 11.21 (p. 330)	Along the length : 3 Across the section: 1
	Cubic (KEYOPT(3) = 3)	Equation 11.26 (p. 331), Equation 11.27 (p. 331), and Equation 11.28 (p. 331)`	Along the length: 4 Across the section: 1

Load Type	Distribution
Element Temperature	KEYOPT(1) = 0 Linear through wall and linear along length KEYOPT(1) = 1 Bilinear across cross-section and linear along length
Nodal Temperature	Constant across cross-section, linear along length

Load Type	Distribution
Internal and External Pressures	Constant, except as adjusted by the affect of internal and external fluids.

References: Simo and Vu-Quoc([237] (p. 934)), Ibrahimbegovic([238] (p. 934)).

13.288.1. Assumptions and Restrictions

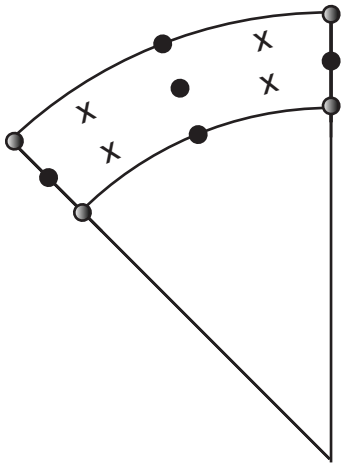
The element is based on Timoshenko beam theory; therefore, shear deformation effects are included. The element is well-suited for linear, large rotation, and/or large strain nonlinear applications.

The element includes stress stiffness terms, by default, in any analysis using large deformation (**NLGEOM**,ON). The stress stiffness terms provided enable the elements to analyze flexural, lateral and torsional stability problems (using eigenvalue buckling or collapse studies with arc length methods).

Transverse shear strain is constant through cross-section (that is, cross sections remain plane and undistorted after deformation). The element can be used for slender or stout beams. Due to the limitations of first-order shear deformation theory, slender to moderately thick beams can be analyzed. Slenderness ratio of a beam structure may be used in judging the applicability of the element. It is important to note that this ratio should be calculated using some global distance measures, and not based on individual element dimensions. A slenderness ratio greater than 30 is recommended.

The elements are provided with section relevant quantities (area of integration, position, Poisson function, function derivatives, etc.) automatically at a number of section points by the use of section commands. Each section is assumed to be an assembly of number of nine-node cells. The number of cells in both the circumferential and thickness directions is controlled by the **SECDATA** command. Each cell has four integration points, as shown in the following figure:

Figure 13.40: Section Model



The section includes internal fluid which contributes only mass and applied pressure, and insulation which contributes only mass, and adds to the effective (hydraulic) diameter.

13.288.2. Ocean Effects

The following ocean effects apply to this element:

13.288.2.1. Location of the Element

The origin for any problem containing PIPE288 using ocean effects must be at the free surface (mean sea level). Further, the Z axis is always the vertical axis, pointing away from the center of the earth.

The element may be located in the fluid, above the fluid, or in both regimes simultaneously. There is a tolerance of only $\frac{D_e}{8}$ below the mud line, for which

$$D_e = D_o + 2t_i \tag{13.402}$$

where:

D_e = effective diameter of the pipe
D_o = outside diameter of pipe (input as D_o on the **SECDATA** command)
t_i = thickness of external insulation (input as T_{ins} on the **SECDATA** command)

The mud line is located at distance d below the origin (input as *DEPTH* on the **OCDATA** command). This condition is checked with:

$$Z(N) > -\left(d + \frac{D_e}{8}\right) \leftarrow \text{no error message} \tag{13.403}$$

$$Z(N) \leq -\left(d + \frac{D_e}{8}\right) \leftarrow \text{fatal error message} \tag{13.404}$$

where Z(N) = vertical location of node N.

If it is necessary to generate a structure below the mud line, a second material property can be set up for those elements using a greater d and omitting hydrodynamic effects.

For a large-deflection problem, greater tolerances apply for second and subsequent iterations:

$$Z(N) > -(d + 10D_e) \leftarrow \text{no error message} \tag{13.405}$$

$$-(d + 10D_e) \geq Z(N) > (2d) \leftarrow \text{warning message} \tag{13.406}$$

$$-(2d) \geq Z(N) \leftarrow \text{fatal error message} \tag{13.407}$$

The element is allowed to sink into the mud for 10 diameters before generating a warning message. If a node sinks into the mud a distance equal to the water depth, the analysis terminates. If the element is intended to lie on the ocean floor, gap elements must be provided.

13.288.2.2. Mass Matrix

The element mass matrix is comprised of several components, each organized on a mass-per-unit-length basis:

$$m_w = \rho \frac{\pi}{4}(D_o^2 - D_i^2) = \text{mass of pipe wall per unit length} \quad (13.408)$$

where:

ρ = density of the pipe wall (input as DENS on the **MP** command)

$$m_{int} = \rho_{int} \frac{\pi}{4} D_i^2 = \text{mass of internal fluid per unit length} \quad (13.409)$$

where:

ρ_{int} = density of the internal fluid (input as DENS on the **MP** command, with the material number specified via M_{int} on the **SECDATA** command)

$$m_{ins} = \rho_i \frac{\pi}{4}(D_e^2 - D_o^2) = \text{mass of insulation per unit length} \quad (13.410)$$

where:

ρ_i = density of external insulation (input as DENS on the **MP** command, with the material number specified via M_{ins} on the **SECDATA** command)

$$m_{ah} = \text{mass of additional hardware per unit length} \quad (13.411)$$

where:

m_{ah} is input as *ADDMAS* on the **SECCONTROL** command

$$m_{add} = C_a \rho_w \frac{\pi}{4} D_e^2 = \text{mass of external fluid per unit length} \quad (13.412)$$

where:

C_a = coefficient of added mass of the external fluid (input as *Ca* on the **OCDATA** command)
ρ_w = fluid density (input as DENS on the **MP** command, with the material number specified via *MATOC* on the **OCDATA** command)

m_{add} only resists submerged lateral acceleration. If the element is above the instantaneous free surface, m_{add} = 0.0 . For modal (**ANTYPE**,MODAL) and harmonic (**ANTYPE**,HARMIC) analyses, the surface location does not account for ocean waves.

13.288.2.3. Load Vector

The element load vector consists of two parts:

- Distributed force per unit length to account for hydrostatic (buoyancy) effects ($\{F/L\}_b$) as well as axial nodal forces due to internal pressure and temperature effects $\{F_x\}$.

- Distributed force per unit length to account for hydrodynamic effects (current and waves) ($\{F/L\}_d$).

The hydrostatic and hydrodynamic effects work with the original diameter and length, i.e., initial strain and large deflection effects are not considered.

13.288.2.4. Hydrostatic Effects

See Hydrostatic Loads (p. 393) in the Element Tools section of this document.

13.288.2.5. Hydrodynamic Effects

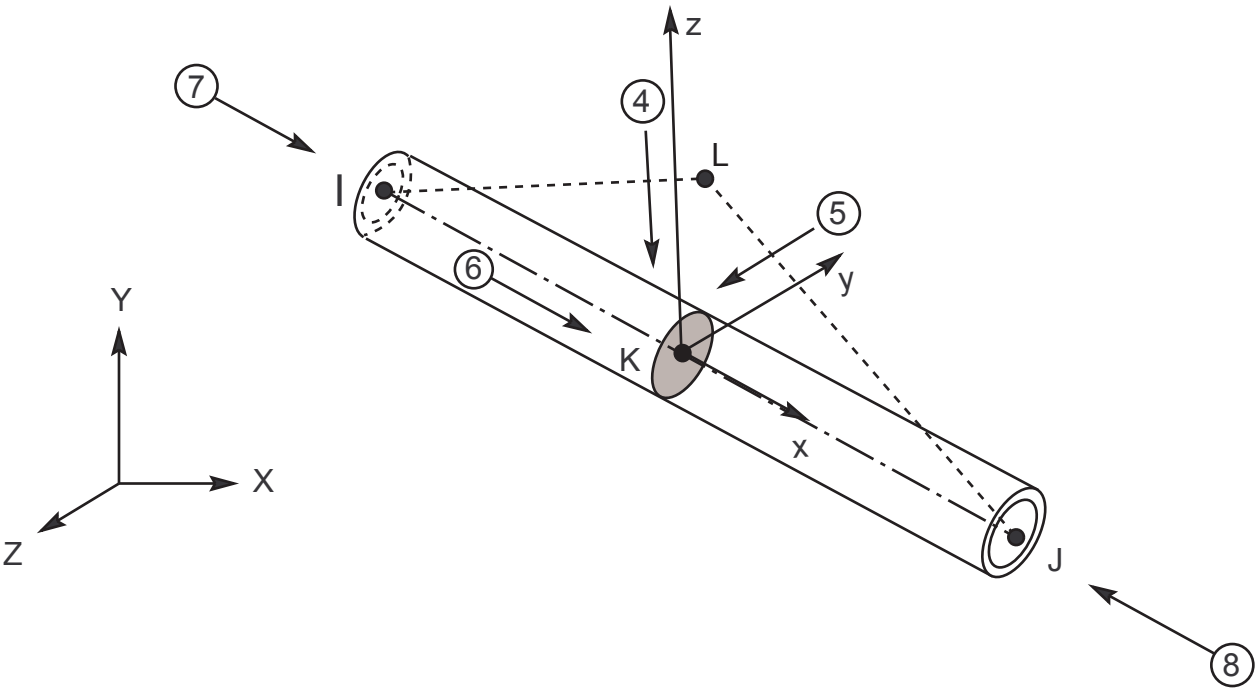
See Hydrodynamic Loads (p. 395) in the Element Tools section of this document.

13.288.3. Stress Evaluation

Several stress evaluation options exist. The section strains and generalized stresses are evaluated at element integration points and then linearly extrapolated to the nodes of the element.

If the material is elastic, stresses and strains are available after extrapolation in cross-section at the nodes of section mesh. If the material is plastic, stresses and strains are moved without extrapolation to the section nodes (from section integration points).

13.289. PIPE289 - 3-D 3-Node Pipe



Matrix or Vector	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal and Newton-Raphson Load Vectors	Equation 11.19 (p. 330), Equation 11.20 (p. 330), Equation 11.21 (p. 330), Equation 11.22 (p. 331), Equation 11.23 (p. 331), and Equation 11.24 (p. 331)	Along the length: 2 Across the section: see PIPE288 - 3-D 2-Node Pipe (p. 657)

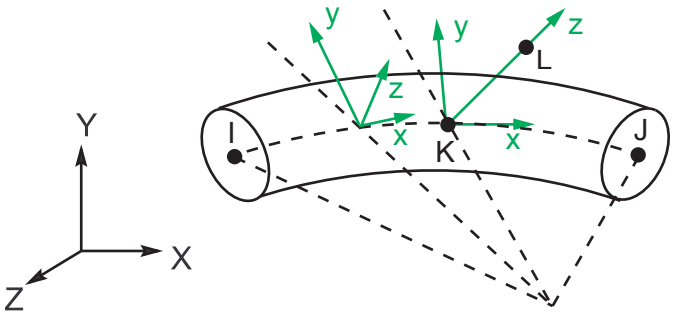
Matrix or Vector	Shape Functions	Integration Points
Consistent Mass Matrix and Pressure Load Vector	Same as stiffness matrix	Along the length: 3 Across the section: 1
Lumped Mass Matrix	Equation 11.19 (p. 330), Equation 11.20 (p. 330), and Equation 11.21 (p. 330)	Along the length: 3 Across the section: 1

Load Type	Distribution
Element Temperature	KEYOPT(1) = 0 Linear thru wall and linear along length KEYOPT(1) = 1 Bilinear across cross-section and linear along length
Nodal Temperature	Constant across cross-section, linear along length
Internal and External Pressures	Constant, except as adjusted by the affect of internal and external fluids.

References: Simo and Vu-Quoc([237] (p. 934)), Ibrahimbegovic([238] (p. 934)).

The theory for this element is identical to that of PIPE288 - 3-D 2-Node Pipe (p. 657), except that this element is a nonlinear, 3-node pipe.

13.290. ELBOW290 - 3-D 3-Node Elbow



Matrix or Vector	Shape Functions	Integration Points
Stiffness and Stress Stiffness Matrices; and Thermal and Newton-Raphson Load Vectors	Equation 11.19 (p. 330), Equation 11.20 (p. 330), Equation 11.21 (p. 330), Equation 11.22 (p. 331), Equation 11.23 (p. 331), and Equation 11.24 (p. 331) Around the circumference: Fourier Series	Along the length: 2 Around the circumference: 8 or higher Thru-the-thickness: 1, 3, 5, 7, or 9 per layer
Mass Matrix and Pressure Load Vector	Same as stiffness matrix	Along the length: 3 Around the circumference: 8 or higher Thru-the-thickness: 1, 3, 5, 7, or 9 per layer

Load Type	Distribution
Element Temperature	KEYOPT(1) = 0 Linear thru wall and linear along length KEYOPT(1) = 1 Bilinear across cross-section and linear along length
Nodal Temperature	Constant across cross-section, linear along length
Internal and External Pressures	Constant

References:

Bathe and Almeida ([369] (p. 941))

Yan, Jospin, and Nguyen ([370] (p. 941))

13.290.1. Other Applicable Sections

Structures (p. 5) describes the derivation of structural element matrices and load vectors as well as stress evaluations.

13.290.2. Assumptions and Restrictions

Pipe cross-sectional motions (i.e., radial expansion, ovalization, and warping) are modeled with Fourier series. The corresponding unknowns (Fourier magnitudes) are treated as internal degrees of freedom. A higher number of Fourier modes may be required to achieve an adequate level of accuracy in cross-sectional motions. Also, a higher number of integration points around the circumference may be needed for capturing nonlinear material behaviors or ensuring sufficient numerical integration accuracy.

No slippage is assumed between the element layers. Shear deflections are included in the element; however, normals to the center wall surface are assumed to remain straight after deformation, but not necessarily normal to the center surface. Therefore, constant transverse shears through the pipe wall are allowed.

13.290.3. Shear Correction

The element uses an equivalent energy method to compute shear correction factors. These factors are predetermined based on the section lay-up at the start of solution

Chapter 14: Analysis Tools

The following analysis tools are available:

14.1. Acceleration Effect

Accelerations are applicable only to elements with displacement degrees of freedom (DOFs).

The acceleration vector $\{a_c\}$ which causes applied loads consists of a vector with a term for every degree of freedom in the model. In the description below, a typical node having a specific location and accelerations associated with the three translations and three rotations will be considered:

$$\{a_c\} = \begin{Bmatrix} \{a_t\} \\ \{a_r\} \end{Bmatrix} \tag{14.1}$$

where:

$\{a_t\} = \{a_t^d\} + \{a_t^I\} + \{a_t^r\}$ = translational acceleration vector

$\{a_r\} = \{a_r^I\} + \{a_r^r\}$ = rotational acceleration vector

where:

$\{a_t^d\}$ = accelerations in global Cartesian coordinates (input on **ACEL** command or **CMACEL** command for component base acceleration)

$\{a_t^I\}$ = translational acceleration vector due to inertia relief (see Inertia Relief (p. 669))

$\{a_r^I\}$ = rotational acceleration vector due to inertia relief (see Inertia Relief (p. 669))

$\{a_t^r\}$ = translational acceleration vector due to rotations (defined below)

$\{a_r^r\}$ = angular acceleration vector due to input rotational accelerations (defined below)

ANSYS defines three types of rotations:

Rotation 1: The whole structure rotates about each of the global Cartesian axes (input on **OMEGA** and **DOMEGA** commands)
Rotation 2: The element component rotates about an axis defined by user (input on **CMOMEGA** and **CMDOMEGA** commands).
Rotation 3: The global origin rotates about the axis by user if Rotation 1 appears or the rotational axis rotates about the axis defined by user if Rotation 2 appears (input on **CGOMGA**, **DCGOMG**, and **CGLOC** commands)

Up to two out of the three types of rotations may be applied on a structure at the same time.

The angular acceleration vector due to rotations is:

$$\{a_r^r\} = \{\dot{\omega}\} + \{\dot{\Omega}\} + \{\Omega\} \times \{\omega\} \tag{14.2}$$

The translational acceleration vector due to rotations is:

$$\{a_t^r\} = \{\omega\} \times (\{\omega\} \times \{r\}) + \{\dot{\omega}\} \times \{r\} + 2 \cdot \{\Omega\} \times (\{\omega\} \times \{r\}) + \Omega \times (\{\Omega\} \times (\{R\} + \{r\})) + \{\dot{\Omega}\} \times (\{R\} + \{r\}) \tag{14.3}$$

where:

x = vector cross product

In the case where the rotations are the combination of **Rotation 1** and **Rotation 3**:

$\{\omega\}$ = angular velocity vector defined about the global Cartesian origin (input on **OMEGA** command)
$\{\Omega\}$ = angular velocity vector of the overall structure about the point CG (input on **CGOMGA** command)
$\{\dot{\omega}\}$ = angular acceleration vector defined about the global Cartesian origin (input on **DOMEGA** command)
$\{\dot{\Omega}\}$ = angular acceleration vector of the overall structure about the point CG (input on **DCGOMG** command)
{r} = position vector (see Figure 14.1: Rotational Coordinate System (Rotations 1 and 3) (p. 667))
{R} = vector from CG to the global Cartesian origin (computed from input on **CGLOC** command, with direction opposite as shown in Figure 14.1: Rotational Coordinate System (Rotations 1 and 3) (p. 667).

In the case where the rotations are **Rotation 1** and **Rotation 2**:

$\{\omega\}$ = angular velocity vector defined about the rotational axis of the element component (input on **CMOMEGA** command)
$\{\Omega\}$ = angular velocity vector defined about the global Cartesian origin (input on **OMEGA** command)

$\{\dot{\omega}\}$ = angular acceleration vector defined about the rotational axis of the element component (input on **CMDOMEGA** command)

$\{\dot{\Omega}\}$ = angular acceleration vector defined about the global Cartesian origin (input on **DOMEGA** command)
{r} = position vector (see Figure 14.2: Rotational Coordinate System (Rotations 1 and 2) (p. 668))
{R} = vector from about the global Cartesian origin to the point on the rotational axis of the component (see Figure 14.2: Rotational Coordinate System (Rotations 1 and 2) (p. 668)).

In the case where the rotations are **Rotation 2** and **Rotation 3**:

{ω} = angular velocity vector defined about the rotational axis of the element component (input on **CMOMEGA** command)
{Ω} = angular velocity vector of the overall structure about the point CG (input on **CGOMGA** command)

$\{\dot{\omega}\}$ = angular acceleration vector defined about the rotational axis of the element component (input on **CMDOMEGA** command)

$\{\dot{\Omega}\}$ = angular acceleration vector of the overall structure about the point CG (input on **DCGOMG** command)
{r} = position vector (see Figure 14.3: Rotational Coordinate System (Rotations 2 and 3) (p. 669))
{R} = vector from CG to the point on the rotational axis of the component (see Figure 14.3: Rotational Coordinate System (Rotations 2 and 3) (p. 669))

Figure 14.1: Rotational Coordinate System (Rotations 1 and 3)

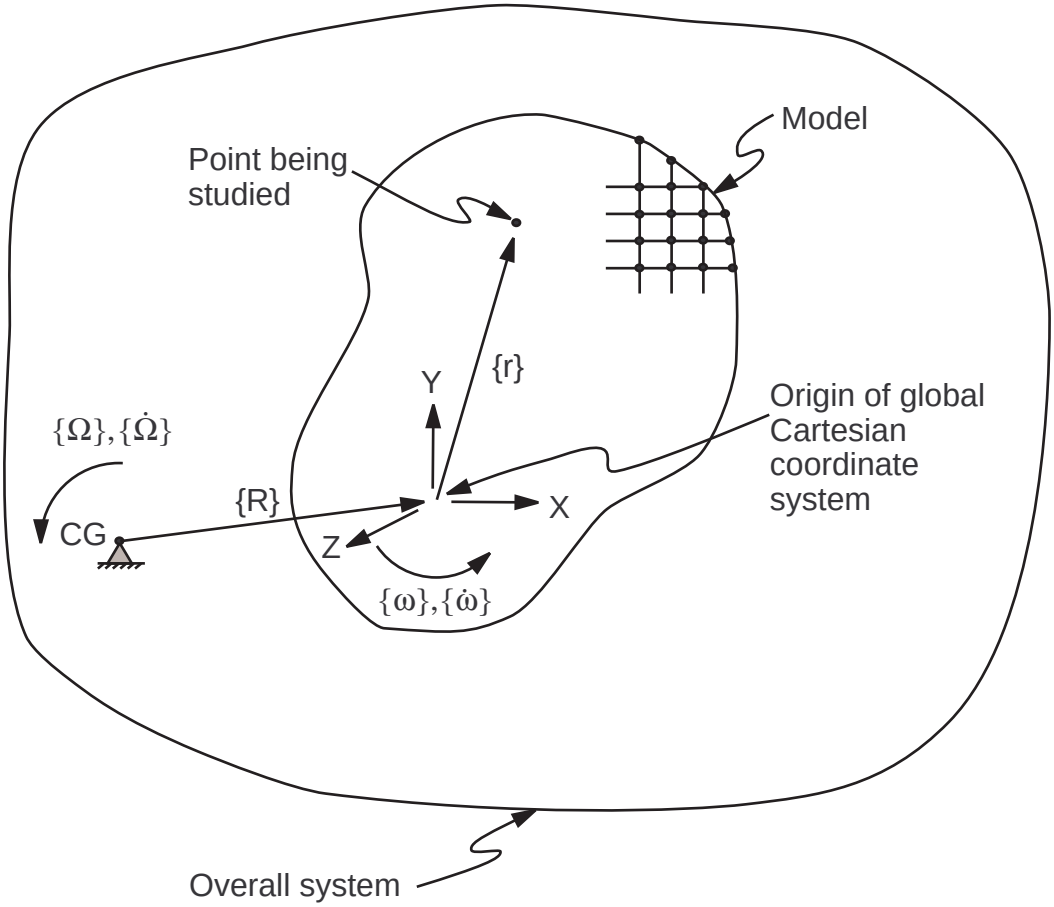


Figure 14.2: Rotational Coordinate System (Rotations 1 and 2)

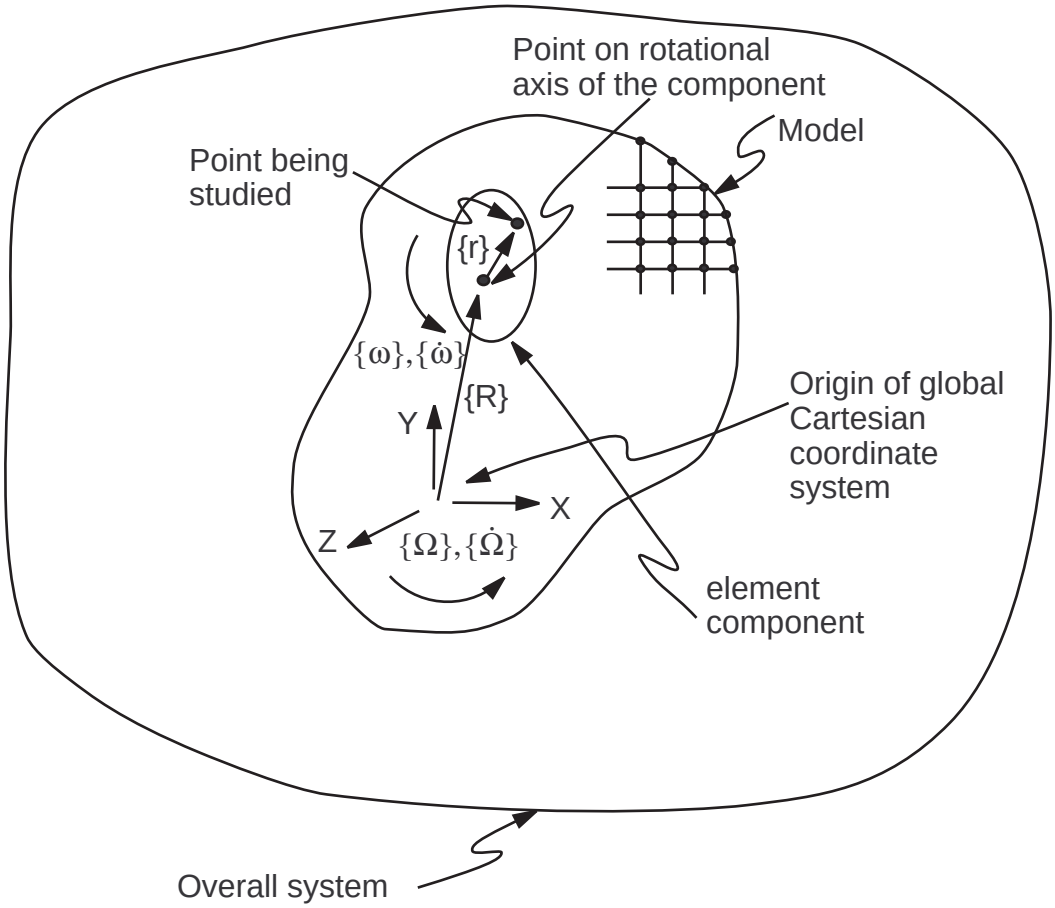
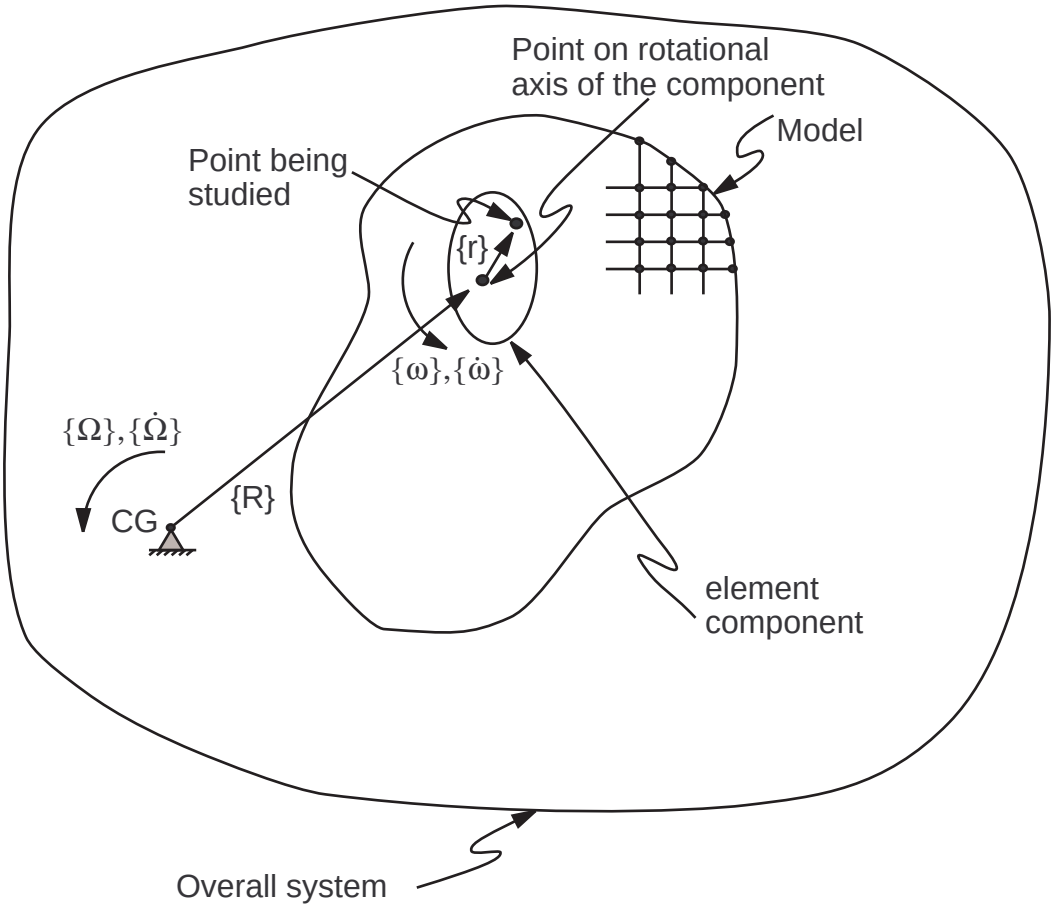


Figure 14.3: Rotational Coordinate System (Rotations 2 and 3)



For MASS21 with KEYOPT(3) = 0 and MATRIX27 with KEYOPT(3) = 2, additional Euler's equation terms are considered:

$$\{M\} = \{\omega_T\} \times [I]\{\omega_T\} \tag{14.4}$$

where:

{M} = additional moments generated by the angular velocity
[I] = matrix of input moments of inertia
$\{\omega_T\}$ = total applied angular velocities: = $\{\omega\} + \{\Omega\}$

14.2. Inertia Relief

Inertia relief is applicable only to the structural parts of linear analyses.

Symmetry models are not valid for inertia relief analysis.

An equivalent free-body analysis is performed if a static analysis (**ANTYPE**,STATIC) and inertia relief (**IRLF**,1) are used. This is a technique in which the applied forces and torques are balanced by inertial forces induced by an acceleration field. Consider the application of an acceleration field (to be determined) that precisely cancels or balances the applied loads:

$$\{F_t^a\} + \int_{vol} \{a_t^I\} \rho \, d(vol) = \{0\}$$
$$\{F_r^a\} + \int_{vol} \{r\} \times (\{a_r^I\} \times \{r\}) \rho \, d(vol) = \{0\} \quad (14.5)$$

where:

$\{F_t^a\}$ = force components of the applied load vector

$\{a_t^I\}$ = translational acceleration vector due to inertia relief (to be determined)

ρ = density

vol = volume of model

$\{F_r^a\}$ = moment components of the applied load vector

$\{r\} = \text{position vector} = \lfloor X \; Y \; Z \rfloor^T$

$\{a_r^I\}$ = rotational acceleration vector due to inertia relief (to be determined)

x = vector cross product

In the finite element implementation, the position vector {r} and the moment in the applied load vector $\{F_r^a\}$ are taken with respect to the origin. Considering further specialization for finite elements, Equation 14.5 (p. 670) is rewritten in equivalent form as:

$$[M_t]\{a_t^I\} = \{F_t^a\}$$
$$[M_t]\{a_r^I\} = \{F_r^a\} \quad (14.6)$$

where:

$[M_t]$ = mass tensor for the entire finite element model (developed below)

$[M_r]$ = mass moments and mass products of the inertia tensor for the entire finite element model (developed below)

Once $[M_t]$ and $[M_r]$ are developed, then $\{a_t^I\}$ and $\{a_r^I\}$ in Equation 14.6 (p. 670) can be solved. The output inertia relief summary includes $\{a_t^I\}$ (output as TRANSLATIONAL ACCELERATIONS) and $\{a_r^I\}$ (output as ROTATIONAL ACCELERATIONS).

The computation for $[M_t]$ and $[M_r]$ proceeds on an element-by-element basis:

$$[M_t] = \sum [m_e] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \int_{vol} \rho \, d(vol) \quad (14.7)$$

$$[M_r] = \sum [I_e] = \int_{vol} \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{bmatrix} \rho \, d(vol) \quad (14.8)$$

in which $[m_e]$ and $[I_e]$ relate to individual elements, and the summations are for all elements in the model. The output `precision mass summary' includes components of $[M_t]$ (labeled as TOTAL MASS) and $[M_r]$ (MOMENTS AND PRODUCTS OF INERTIA TENSOR ABOUT ORIGIN).

The evaluation for components of $[m_e]$ are simply obtained from a row-by-row summation applied to the elemental mass matrix over translational (x, y, z) degrees of freedom. It should be noted that $[m_e]$ is a diagonal matrix ($m_{xy} = 0$, $m_{xz} = 0$, etc.). The computation for $[I_e]$ is somewhat more involved, but can be summarized in the following form:

$$[I_e] = [b]^T [M_e][b] \tag{14.9}$$

where:

$[M_e]$ = elemental mass matrix (which may be either lumped or consistent)
$[b]$ = matrix which consists of nodal positions and unity components

The forms of $[b]$ and, of course, $[M_e]$ are dependent on the type of element under consideration. The description of element mass matrices $[M_e]$ is given in Derivation of Structural Matrices (p. 12). The derivation for $[b]$ comes about by comparing Equation 14.5 (p. 670) and Equation 14.6 (p. 670) on a per element basis, and eliminating $\{F_r^a\}$ to yield

$$[M_r]\{a_r^I\} = \int_{vol} \{r\} \times \{a_r^I\} \times \{r\} \rho \, d(vol) \tag{14.10}$$

where:

vol = element volume

After a little manipulation, the acceleration field in Equation 14.10 (p. 671) can be dropped, leaving the definition of $[I_e]$ in Equation 14.9 (p. 671).

It can be shown that if the mass matrix in Equation 14.9 (p. 671) is derived in a consistent manner, then the components in $[I_e]$ are quite precise. This is demonstrated as follows. Consider the inertia tensor in standard form:

$$[I_e] = \int_{vol} \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{bmatrix} \rho \, d(vol) \tag{14.11}$$

which can be rewritten in product form:

$$[I_e] = \int_{vol} [Q]^T [Q] \rho \, d(vol) \tag{14.12}$$

The matrix $[Q]$ is a skew-symmetric matrix.

$$[Q] = \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix} \tag{14.13}$$

Next, shape functions are introduced by way of their basic form,

$$\{r\} = \lfloor XYZ \rfloor^T = [N] \lfloor x_1\, y_1\, z_1\, x_2\, y_2\, z_2 \ldots \rfloor^T \tag{14.14}$$

where:

[N] = usual matrix containing individual shape functions

Omitting the tedious algebra, Equation 14.13 (p. 671) and Equation 14.14 (p. 672) are combined to obtain

$$[Q] = [N][b] \tag{14.15}$$

where:

$$[b]^T = \begin{bmatrix} 0 & -z_2 & y_1 & 0 & -z_2 & y_2 \ldots \\ z_1 & 0 & -x_1 & z_2 & 0 & -x_2 \ldots \\ -y_1 & x_1 & 0 & -y_2 & x_2 & 0 \ldots \end{bmatrix} \tag{14.16}$$

Inserting Equation 14.16 (p. 672) into Equation 14.12 (p. 671) leads to

$$[I_e] = [b]^T \int_{vol} [N]^T \rho [N] d(vol) [b] \tag{14.17}$$

Noting that the integral in Equation 14.17 (p. 672) is the consistent mass matrix for a solid element,

$$[M_e] = \int_{vol} [N]^T \rho [N] d(vol) \tag{14.18}$$

So it follows that Equation 14.9 (p. 671) is recovered from the combination of Equation 14.17 (p. 672) and Equation 14.18 (p. 672).

As stated above, the exact form of [b] and $[M_e]$ used in Equation 14.9 (p. 671) varies depending on the type of element under consideration. Equation 14.16 (p. 672) and Equation 14.18 (p. 672) apply to all solid elements (in 2-D, z = 0). For discrete elements, such as beams and shells, certain adjustments are made to [b] in order to account for moments of inertia corresponding to individual rotational degrees of freedom. For 3-D beams, for example, [b] takes the form:

$$[b]^T = \begin{bmatrix} 0 & -z_2 & y_1 & 1 & 0 & 0 & 0 & -z_2 & y_2 & 1 & 0 & 0 \ldots \\ z_1 & 0 & -x_1 & 0 & 1 & 0 & z_2 & 0 & -x_2 & 0 & 1 & 0 \ldots \\ -y_1 & x_1 & 0 & 0 & 0 & 1 & -y_2 & x_2 & 0 & 0 & 0 & 1 \ldots \end{bmatrix} \tag{14.19}$$

In any case, it is worth repeating that precise $[I_e]$ and $[M_r]$ matrices result when consistent mass matrices are used in Equation 14.9 (p. 671).

If inertia relief is requested (**IRLF**,1), then the m_x, m_y, and m_z diagonal components in $[M_t]$ as well as all tensor components in $[M_r]$ are calculated. Then the acceleration fields $\{a_t^I\}$ and $\{a_r^I\}$ are computed by the inversion of Equation 14.6 (p. 670). The body forces that correspond to these accelerations are added to the user-imposed load vector, thereby making the net or resultant reaction forces null. The user may request only a mass summary for $[M_t]$ and $[M_r]$ (**IRLF**,-1).

The calculations for $[M_t]$, $[M_r]$, $\{a_t^I\}$ and $\{a_r^I\}$ are made at every substep of every load step where they are requested, reflecting changes in material density and applied loads.

Several limitations apply:

- Element mass and/or density must be defined in the model.
- In a model containing both 2-D and 3-D elements, only $M_t(1,1)$ and $M_t(2,2)$ in $[M_t]$ and $M_r(3,3)$ in $[M_r]$ are correct in the precise mass summary. All other terms in $[M_t]$ and $[M_r]$ should be ignored. The acceleration balance is, however, correct.
- Axisymmetric and generalized plane strain elements are not allowed.
- If grounded gap elements are in the model, their status should not change from their original status. Otherwise the exact kinematic constraints stated above might be violated.
- The "CENTER OF MASS" output does not include the effects of offsets or tapering on current-technology beam, pipe or elbow elements (such as BEAM188, BEAM189, PIPE288, PIPE289, and ELBOW290). Breaking up each tapered element into several elements will give a more accurate solution.

14.3. Damping Matrices

Damping may be introduced into a transient, harmonic, or damped modal analysis as well as a response spectrum or PSD analysis. The type of damping allowed depends on the analysis as described in the subsequent sections.

14.3.1. Transient (FULL) Analysis and Damped Modal Analysis

The damping matrix, [C], may be used in transient and damped modal analyses as well as substructure generation:

- **ANTYPE**,TRANS with **TRNOPT**,FULL
- **ANTYPE**,MODAL with **MODOPT**,QRDAMP or **MODOPT**,DAMP
- **ANTYPE**,SUBSTR with **SEOPT**,,,3

In its most general form, the damping matrix is composed of the following components:

$$[C] = \alpha[M] + \beta[K] + \sum_{i=1}^{N_{ma}} \alpha_i^m [M_i] + \sum_{j=1}^{N_{mb}} \beta_j^m [K_j] + \sum_{k=1}^{N_e} [C_k] + \sum_{l=1}^{N_g} [G_l] \qquad (14.20)$$

where:

[C] = structural damping matrix
α = mass matrix multiplier (input on **ALPHAD** command)
[M] = structural mass matrix
β = stiffness matrix multiplier (input on **BETAD** command)
[K] = structural stiffness matrix
N_{ma} = number of materials with **MP**,ALPD input
α_i^m = mass matrix multiplier for material i (input as ALPD on **MP** command)
$[M_i]$ = portion of structural mass matrix based on material i
N_{mb} = number of materials with **MP**,BETD input
β_j^m = stiffness matrix multiplier for material j (input as BETD on **MP** command)
$[K_j]$ = portion of structural stiffness matrix based on material j
N_e = number of elements with specified damping

$[C_k]$ = element damping matrix
N_g = number of elements with Coriolis or gyroscopic damping
$[G_l]$ = element Coriolis or gyroscopic damping matrix; see Rotating Structures (p. 678)

Element damping matrices are available for:

LINK11	Linear Actuator	MATRIX50	Superelement
COMBIN14	Spring-Damper	SURF153	2-D Structural Surface Effect
MATRIX27	Stiffness, Damping, or Mass Matrix	SURF154	3-D Structural Surface Effect
COMBIN37	Control	SURF159	General Axisymmetric Surface
FLUID38	Dynamic Fluid Coupling	MPC184	Multipoint Constraint (Joint) Element
COMBIN40	Combination	COMBI214	Bearing
CONTA173	Surface-to-Surface Contact	CONTA174	Surface-to-Surface Contact

Note that [K], the structural stiffness matrix, may include plasticity and/or large-deflection effects (i.e., [K] may be the tangent matrix). In the case of a rotating structure, it may also include spin softening or rotating damping effects. Stiffness matrices generated by contact elements are not included in the damping matrix generation.

For the special case of thin-film fluid behavior, damping parameters may be computed for structures and used in a subsequent structural analysis (see Extraction of Modal Damping Parameter for Squeeze Film Problems (p. 703)).

For constant structural damping in a QRDAMP modal analysis (see QR Damped Method (p. 733) for details), the damping matrix [C] is ignored, and the imaginary part of the stiffness matrix is written as:

$$\left[K_{imag}\right] = g[K] + \sum_{j=1}^{N_m} 2m_j \left[K_j\right] \tag{14.21}$$

where:

g = constant structural damping coefficient (input with **DMPSTR** command)
N_m = number of materials with **MP**,DMPR input
m_j = constant structural damping coefficient for material j (input with **MP**,DMPR)
$[K_j]$ = portion of structural stiffness matrix based on material j

14.3.2. Harmonic (FULL) Analysis

The damping matrix ([C]) used in harmonic analyses (**ANTYPE**,HARM with *Method* = FULL, AUTO, or VT on the **HROPT** command) is composed of the following components:

$$[C] = \alpha[M] + (\beta + \frac{1}{\Omega}g)[K] + \sum_{i=1}^{N_{ma}} \alpha_i^m [M_i] + \sum_{j=1}^{N_m} \left[\left(\beta_j^m + \frac{2}{\Omega}m_j + \frac{1}{\Omega}g_j^E\right)\left[K_j\right]\right] + \sum_{k=1}^{N_e}[C_k] + \sum_{m=1}^{N_v}\frac{1}{\Omega}[C_m] + \sum_{l=1}^{N_g}[G_l] \tag{14.22}$$

where:

[C] = structural damping matrix
α = mass matrix multiplier (input on **ALPHAD** command)
[M] = structural mass matrix
β = stiffness matrix multiplier (input on **BETAD** command)
g = constant structural damping coefficient (input on the **DMPSTR** command)
Ω = excitation circular frequency
[K] = structural stiffness matrix
N_{ma} = number of materials with **MP**,ALPD input
α_i^m = stiffness matrix multiplier for material i (input as ALPD on **MP** command)
$[M_i]$ = portion of structural mass matrix based on material i
N_m = number of materials with **MP**,BETD, DMPR, or SDAMP input
β_j^m = stiffness matrix multiplier for material j (input as BETD on **MP** command)
m_j = constant structural damping coefficient for material j (input as DMPR on the **MP** command)
g_j^E = structural damping coefficient for material j (input as SDAMP on the **TB** command)
$[K_j]$ = portion of structural stiffness matrix based on material j
N_e = number of elements with specified damping
$[C_k]$ = element damping matrix
N_v = number of elements with viscoelastic damping
C_m = element viscoelastic damping matrix
N_g = number of elements with Coriolis or gyroscopic damping
$[G_l]$ = element Coriolis or gyroscopic damping matrix

The input exciting frequency, Ω, is defined in the range between Ω_B and Ω_E via:

$\Omega_B = 2\pi f_B$
$\Omega_E = 2\pi f_E$
f_B = beginning frequency (input as *FREQB* on **HARFRQ** command)
f_E = end frequency (input as *FREQE* on **HARFRQ** command)

Substituting Equation 14.22 (p. 674) into the harmonic response equation of motion (Equation 15.60 (p. 781)) and rearranging terms yields:

$$\left(\begin{array}{c} [K] + i\left(g[K] + \sum\left(2m_j + g_j^E\right)[K_j] + \sum[C_m]\right) + \\ i\Omega\left[\alpha[M] + \sum\alpha_i^m[M_i] + \beta[K] + \sum\beta_j^m[K_j] + \sum[C_k] + \sum[G_l]\right] - \\ \Omega^2[M] \end{array} \right) \left(\{u_1\} + i\{u_2\}\right) = \{F_1\} + i\{F_2\} \quad (14.23)$$

The complex stiffness matrix in the first row of the equation consists of the normal stiffness matrix augmented by the structural damping terms given by g, m_j, g_j^E, and $[C_m]$ which produce an imaginary contribution. Structural damping is independent of the forcing frequency, Ω, and produces a damping force proportional to displacement (or strain). Note that the terms g, g_j, and g_j^E are damping coefficients and not damping ratios.

The second row consists of the usual viscous damping terms and is linearly dependent on the forcing frequency, Ω, and produces forces proportional to velocity.

Viscoelastic damping (see Harmonic Viscoelasticity) also introduces a contribution to the complex stiffness matrix via the loss moduli. Note that the stresses are also computed using the loss moduli, whereas in the case of structural damping, which is a phenomenological model, the stresses are computed only using the real material properties and g is not used in the stress calculations.

14.3.3. Mode-Superposition Analysis

For mode-superposition based analyses:

- **ANTYPE**,HARM with **HROPT**,MSUP
- **ANTYPE**,TRANS with **TRNOPT**,MSUP
- **ANTYPE**,SPECTRUM with *Sptype* = SPRS, MPRS, or PSD on the **SPOPT** command

the damping matrix is not explicitly computed, but rather the damping is defined directly in terms of a damping ratio ξ^d. The damping ratio is the ratio between actual damping and critical damping.

The damping ratio ξ_i^d for mode i is the combination of:

$$\xi_i^d = \xi + \xi_i^m + \frac{\alpha}{2\omega_i} + \frac{\beta}{2}\omega_i \tag{14.24}$$

where:

ξ = constant modal damping ratio (input on **DMPRAT** command)
ξ_i^m = modal damping ratio for mode shape i (see below)
ω_i = circular natural frequency associated with mode shape i = $2\pi f_i$
f_i = natural frequency associated with mode shape i
α = mass matrix multiplier (input on **ALPHAD** command)
β = stiffness matrix multiplier (input on **BETAD** command)

The modal damping ratio ξ_i^m can be defined for each mode directly using the **MDAMP** command (undamped modal analyses only).

Alternatively, for the case where multiple materials are present whose damping ratios are different, an effective mode-dependent damping ratio ξ_i^m can be defined in the modal analysis if material-dependent damping is defined and the element results are calculated during the expansion (**MXPAND**,,,,YES). This effective damping ratio is computed from the ratio of the strain energy in each material in each mode using:

$$\xi_i^m = \frac{\sum_{j=1}^{N_m} \xi_j^m E_j^s}{\sum_{j=1}^{N_m} E_j^s} \tag{14.25}$$

where:

N_m = number of materials

β_j^m = damping ratio for material j (input as DMPR on **MP** command); see note below

$E_j^s = \frac{1}{2}\{\phi_i\}^T[K_j]\{\phi_i\}$ = strain energy contained in mode i for material j

$\{\varphi_i\}$ = displacement vector for mode i

$[K_j]$ = stiffness matrix of part of structure of material j

These mode-dependent (and material-dependent) ratios, ξ_i^m, will be carried over into the subsequent mode-superposition or spectrum analysis. Note that any manually-defined damping ratios (**MDAMP**) will overwrite those computed in the modal analysis via Equation 14.25 (p. 676).

For harmonic analyses (**ANTYPE**,HARM with **HROPT**,MSUP), constant structural damping may also be included. In this case, the harmonic equation of motion in modal coordinates (Equation 15.75 (p. 784)) is:

$$\left(-\Omega^2 + i\left(2\omega_i\Omega\xi_i^d + \omega_i^2 g\right) + \omega_i^2\right) y_{ic} = f_{ic} \tag{14.26}$$

where:

y_{ic} = complex modal coordinate

ω_i = natural circular frequency of mode i

ξ_i^d = fraction of critical damping for mode i as given in Equation 14.24 (p. 676)

g = constant structural damping coefficient (input on **DMPSTR** command)

f_{ic} = complex force in modal coordinates

14.3.3.1. Specialization for Mode-Superposition Analyses Following a QR Damp Modal Analysis

Equation 14.24 (p. 676) presumes Rayleigh damping (see the derivation in Mode Superposition Method (p. 698)). With Rayleigh (or proportional) damping, the damping in the modal basis leads to the diagonal entries of Equation 14.24 (p. 676).

$$\{\Phi_i\}^T[C_\xi]\{\Phi_i\} = 2\xi_i^d\omega_i \tag{14.27}$$

For QR Damp modal analyses (**MODOPT**,QRDAMP), the damping may be from non-Rayleigh sources such as element damping or Coriolis damping. The damping in the modal basis is no longer diagonal, and the full matrix must be retained in the mode-superposition transient or harmonic analysis:

$$[C_m] = [\Phi^T][C][\Phi] + \frac{1}{\Omega}g[\Phi^T][K][\Phi] + \sum_{j=1}^{N_m}\frac{2}{\Omega}m_j[\Phi^T][K_j][\Phi] + [\Xi] \tag{14.28}$$

where:

$[C_m]$ = damping matrix in the modal basis

$[\Phi]$ = matrix of the mode shapes (see QR Damped Method (p. 733))

[C] = damping matrix from Equation 14.20 (p. 673)

g = constant structural damping coefficient (input on **DMPSTR** command)

N_m = number of materials with **MP**,DMPR input

m_j = constant structural damping coefficient for material j (harmonic analyses only, input as DMPR on **MP** command)
N_e = number of elements with specified damping

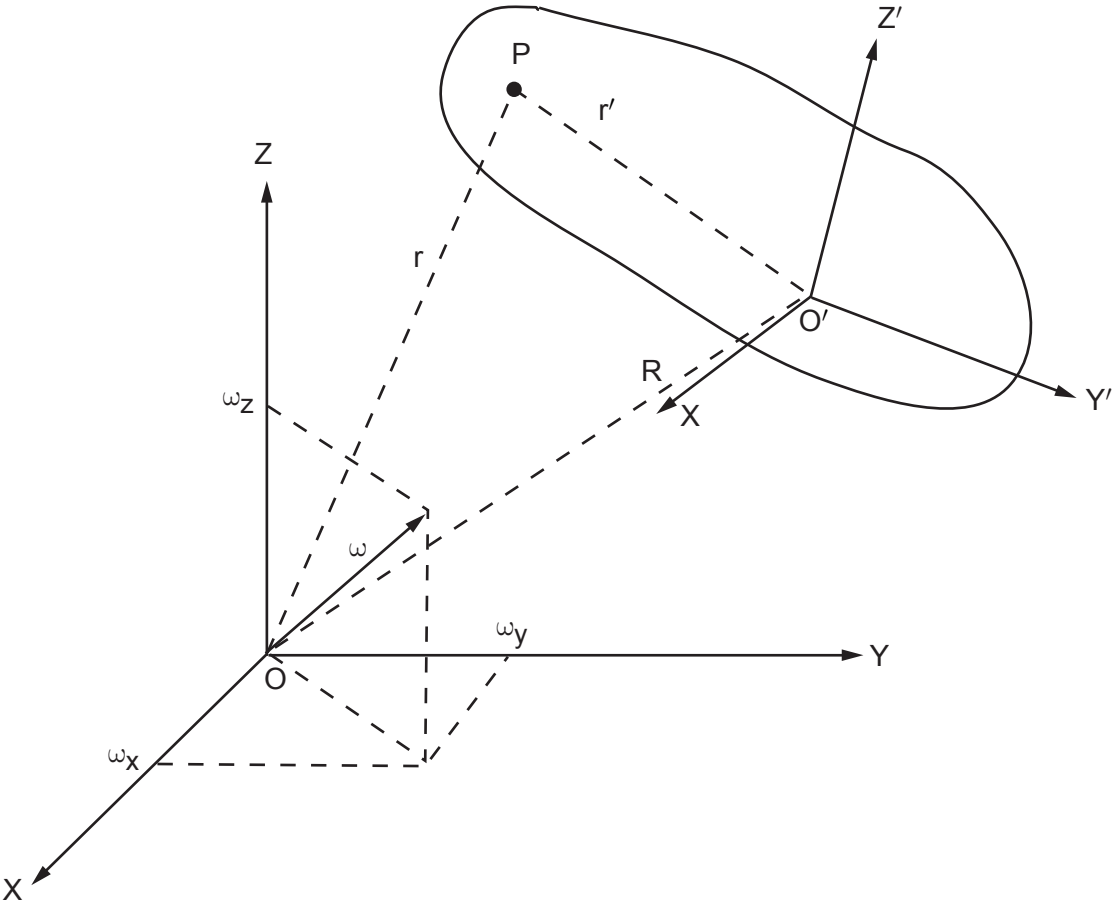
$[\Xi]$ = diagonal matrix with the terms $\xi + {}^{\xi_i^m}$ on the diagonal

The second term (using g_j) is applicable to harmonic analyses only and represents a structural damping ratio (and not a modal damping ratio).

14.4. Rotating Structures

When a structure is rotating, inertial forces and moments are observed. To best express these quantities, you can choose a stationary reference frame: global Cartesian (OXYZ) or a rotating reference frame which is attached to the structure (O'X'Y'Z') (input on **CORIOLIS** command).

Figure 14.4: Reference Frames



The case of a stationary reference frame is developed in Gyroscopic Matrix in a Stationary Reference Frame (p. 681) and leads to the so-called gyroscopic matrix.

The rotating reference frame is addressed below and leads to a Coriolis matrix for dynamic analysis and a Coriolis force for quasi-static analysis. In both types of analyses, the effect of spin softening (Spin Softening (p. 47)) modifies the apparent rigidity of the structure.

Synchronous and asynchronous forces are discussed in Harmonic Analysis with Rotating Forces on Rotating Structures (p. 789).

14.4.1. Coriolis Matrix and Coriolis Force in a Rotating Reference Frame

In Figure 14.4: Reference Frames (p. 678) above, a part or component is rotating at angular velocity $\{\omega\}$, with components ω_x, ω_y, and ω_z defined in the stationary reference frame. The position of a point P with reference to (OXYZ) is $\{r\}$, while its position with reference to the rotating frame of reference $(O'X'Y'Z')$ is $\{r'\}$, and:

$$\{r\} = \{R\} + \{r'\} \tag{14.29}$$

The velocities of point P as observed in the stationary and rotating frames are defined as:

$$\{v_s\} = \left\{\frac{dr}{dt}\right\}_s \tag{14.30}$$

and

$$\{v_r\} = \left\{\frac{dr'}{dt}\right\}_r \tag{14.31}$$

The velocities of point P observed in the stationary frame can be expressed as:

$$\{v_s\} = \left\{\frac{dR}{dt}\right\}_s + \left\{\frac{dr'}{dt}\right\}_s = \{V\} + \{v_r\} + \{\omega\} \times \{r'\} \tag{14.32}$$

where:

$$\{V\} = \text{translational velocity of rotating-frame origin} = \left\{\frac{dR}{dt}\right\}_s$$

The acceleration of point P as observed in the stationary and rotating frames of reference:

$$\{a_s\} = \left\{\frac{dv_s}{dt}\right\}_s = \text{translational acceleration observed in stationary reference frame} \tag{14.33}$$

and

$$\{a_r\} = \left\{\frac{dv_r}{dt}\right\}_r = \text{translational acceleration observed in rotating reference frame} \tag{14.34}$$

By substituting Equation 14.32 (p. 679) into Equation 14.33 (p. 679) and using Equation 14.34 (p. 679), we obtain:

$$\{a_s\} = \{A\} + \{a_r\} + \{\dot{\omega}\} \times \{r'\} + \{\omega\} \times (\{\omega\} \times \{r'\}) + 2\{\omega\} \times \{v_r\} \tag{14.35}$$

where:

$$\{A\} = \text{translational acceleration of rotating-frame origin} = \left\{\frac{dV}{dt}\right\}_s$$

We assume that the origin of the rotating system O' is fixed, so that:

$$\{V\} = \{A\} = \{0\} \tag{14.36}$$

By substituting Equation 14.36 (p. 679) into Equation 14.35 (p. 679),

$$\{a_s\} = \{a_r\} + \{\dot{\omega}\} \times \{r'\} + \{\omega\} \times (\{\omega\} \times \{r'\}) + 2\{\omega\} \times \{v_r\} \tag{14.37}$$

By applying virtual work from the d'Alembert force, the contribution of the first term $\{\mathbf{a}_r\}$ to the virtual work introduces the mass matrix of the element (Guo et al.([364] (p. 941))).

$$[M_e] = \int_v [N]^T [N] \rho \, dv \tag{14.38}$$

where:

$[M_e]$ = element mass matrix
N = shape function matrix
ρ = element density

The second term $\{\dot{\omega}\} \times \{r'\}$, is the rotational acceleration load term (see Acceleration Effect (p. 665)).

The third term $(\omega \times r')^*$) is the centrifugal load term (see Guo et al.([364] (p. 941)), Acceleration Effect (p. 665), and Stress Stiffening (p. 41)).

The last term contributes to the Coriolis force which generates the damping matrix of the element as a skew symmetric matrix (Guo et al.([364] (p. 941))):

$$[G_e] = 2\int_v [N]^T [\omega][N] \rho \, dV \tag{14.39}$$

where:

$[G_e]$ = element Coriolis damping matrix

$$[\omega] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \text{rotational matrix associated with } \{\omega\}$$

The governing equation of motion in dynamic analysis can be written as,

$$[M]\{\ddot{u}\} + ([G] + [C])\{\dot{u}\} + ([K] - [K_c])\{u\} = \{F\} \tag{14.40}$$

where:

$$[M] = \text{global mass matrix} = \sum_i^n [M_e]$$

$$[G] = \text{global Coriolis matrix} = \sum_i^n [G_e]$$

$$[C] = \text{global damping matrix} = \sum_i^n [C_e]$$

n = number of elements
[K] = global stiffness matrix
$[K_c]$ = global stiffness due to centrifugal force Spin Softening (p. 47)
{F} = load vector

In a quasi-static analysis, Coriolis force term will be introduced as a load vector as:

$$\{F_c\} = [G]\{\dot{u}\} \tag{14.41}$$

where:

$\{F_c\}$ = Coriolis force

$\{\dot{u}\}$ = nodal velocity vector (input using the **IC** command).

Coriolis forces and damping matrices are available for the elements listed under ROTATING REFERENCE FRAME in the Notes section of the **CORIOLIS** command.

14.4.2. Gyroscopic Matrix in a Stationary Reference Frame

Suppose a structure is spinning around an axis Δ. If a rotation about an axis perpendicular to Δ is applied to the structure, then a reaction moment appears. It is called the gyroscopic moment. Its axis is perpendicular to both the spinning axis Δ and the applied rotation axis.

The gyroscopic effect is thus coupling rotational degrees of freedom which are perpendicular to the spinning axis.

Let us consider the spinning axis is along X so:

- The spinning velocity (input using the **OMEGA** or **CMOMEGA** commands) is $\omega_x = \dot{\theta}_x$.
- The displacements perpendicular to the spin axis are u_y and u_z.
- The corresponding rotations are θ_y and θ_z, and the angular velocities are $\dot{\theta}_y$ and $\dot{\theta}_z$.

The gyroscopic finite element matrix is calculated from the kinetic energy due to the inertia forces.

The kinetic energy for lumped mass and beam element (Nelson and McVaugh([362] (p. 941))) is detailed in Kinetic Energy for the Gyroscopic Matrix Calculation of Lumped Mass and Legacy Beam Element (p. 681) below.

The general expression of the kinetic energy used for the development of the gyroscopic matrices for all other elements (Geradin and Kill [380] (p. 942)) is presented in General Expression of the Kinetic Energy for the Gyroscopic Matrix Calculation (p. 682)

14.4.2.1. Kinetic Energy for the Gyroscopic Matrix Calculation of Lumped Mass and Legacy Beam Element

Both mass and beam are supposed to be axisymmetric around the spinning axis. The spinning axis is along one of the principal axis of inertia for lumped mass. For the beam, it is along the length.

Two reference frames are used (see Figure 14.4: Reference Frames (p. 678)) (OXYZ) which is stationary and $(O'X'Y'Z')$ which is attached to the cross-section with X' axis normal to it.

$(O'X'Y'Z')$ is defined using 3 successive rotations:

- θ_z around Z axis to give (x'', y'', z'')
- θ_y around y'' axis to give (x', y', z')

- θ_x around x' axis to give (X', Y', Z')

Hence for small rotations θ_y and θ_z, the instantaneous angular velocity is:

$$\{\omega_i\} = \begin{Bmatrix} -\dot{\theta}_z\theta_y + \omega_x \\ \dot{\theta}_z \sin\omega_x t + \dot{\theta}_y \cos\omega_x t \\ \dot{\theta}_z \cos\omega_x t - \dot{\theta}_y \sin\omega_x t \end{Bmatrix} \tag{14.42}$$

1. For a lumped mass, considering only second order terms, kinetic energy is obtained using the instantaneous angular velocity vector in Equation 14.42 (p. 682).

$$E^{ki}_{mass} = \frac{1}{2}\begin{Bmatrix}\dot{u}_y \\ \dot{u}_z\end{Bmatrix}^T \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{Bmatrix}\dot{u}_y \\ \dot{u}_z\end{Bmatrix} + \frac{1}{2}\begin{Bmatrix}\dot{\theta}_y \\ \dot{\theta}_z\end{Bmatrix}^T \begin{bmatrix} I_d & 0 \\ 0 & I_d \end{bmatrix} \begin{Bmatrix}\dot{\theta}_y \\ \dot{\theta}_z\end{Bmatrix} - \omega_x I_p \dot{\theta}_z \theta_y \tag{14.43}$$

   where:

   E^{ki}_{mass} = total kinetic energy of the mass element
   m = mass
   I_d = diametral inertia
   I_p = polar inertia

   The first two terms contribute to the mass matrix of the element and the last term gives the gyroscopic matrix.

2. The beam element is considered as an infinite number of lumped masses. The gyroscopic kinetic energy of the element is obtained by integrating the last term of Equation 14.43 (p. 682) along the length of the beam:

$$E^{Gki}_{beam} = -2\rho I_x \omega_x \int_0^L \dot{\theta}_z \theta_y dx \tag{14.44}$$

   where:

   E^{Gki}_{beam} = gyroscopic kinetic energy of the beam element
   ρ = density
   I_x = moment of inertia normal to x
   L = length of the beam element

   Gyroscopic matrix is deduced based on the element shape functions. (See PIPE288).

Gyroscopic matrices are available for the elements listed under STATIONARY REFERENCE FRAME in the Notes section of the **CORIOLIS** command.

14.4.2.2. General Expression of the Kinetic Energy for the Gyroscopic Matrix Calculation

A point, in element i, with coordinates (x,y,z) in the stationary reference frame is considered. The kinetic energy is

$$E^{Gki} = -\omega_x \int_{V_i} x(\dot{\theta}_z y + \dot{\theta}_y z) dm \tag{14.45}$$

where:

E^{Gki} = gyroscopic kinetic energy of element i
V_i = volume of element i
dm = elementary mass

The gyroscopic matrix is then calculated using the element shape functions.

Gyroscopic matrices are available for the elements listed under STATIONARY REFERENCE FRAME in the Notes section of the **CORIOLIS** command.

14.4.3. Rotating Damping Matrix in a Stationary Reference Frame

In a linear approach, the relation between displacements in the stationary reference frame (0XYZ) and displacements in the rotating reference frame (0X'Y'Z') can be written as:

$$\{r'\} = [R]\{r\} \tag{14.46}$$

where:

r' = the displacement vector in the rotating reference frame
[R] = the transformation matrix
{r} = the displacement vector in the stationary reference frame

Differentiating Equation 14.46 (p. 683) with respect to time, one obtains the expression for the velocity vector:

$$\{\dot{r}'\} = [R]\{\dot{r}\} + [R][\omega]^T\{r\} \tag{14.47}$$

where:

$\{\dot{r}'\}$ = the velocity vector in the rotating reference frame

$\{\dot{r}\}$ = the velocity vector in the stationary reference frame
ω = the rotational matrix, as defined in Equation 14.39 (p. 680)

If structural damping is present in the rotating structure (proportional damping for example) or if there is localized viscous damping (as in a damper), damping forces in the rotating reference frame may be expressed as:

$$\{F'_d\} = [C]\{\dot{r}'\} \tag{14.48}$$

where:

$\{F'_d\}$ = the damping forces in the rotating reference frame
[C] = the damping matrix

To obtain the damping forces in the stationary reference frame, first apply the transformation of Equation 14.46 (p. 683):

$$\{F_d\} = [R]^T\{F'_d\} \tag{14.49}$$

where:

$\{F_d\}$= the damping forces in the stationary reference frame.

Then replace Equation 14.47 (p. 683) in Equation 14.48 (p. 683), the resulting expression in Equation 14.49 (p. 683) yields:

$$\{F_d\} = [R]^T [C][R]\{\dot{r}\} + [R]^T [C][R][\omega]^T \{r\} \quad (14.50)$$

If the damping is isotropic (implementation assumption):

$$\{F_d\} = [C]\{\dot{r}\} + [B]\{r\} \quad (14.51)$$

Where [B] is the rotating damping matrix:

$$[B] = [C][\dot{E}]^T \quad (14.52)$$

It is a non-symmetric matrix which will modify the apparent stiffness of the structure.

The rotating damping matrix is available for elements that generate a gyroscopic matrix. See the Notes section of the **CORIOLIS** command.

14.5. Element Reordering

The ANSYS program provides a capability for reordering the elements. Since the solver processes the elements sequentially, the order of the elements slightly affects the efficiency of element assembly time. Reordering the elements minimizes the number of DOFs that are active at the same time during element assembly.

Each element has a location, or order, number which represents its sequence in the solution process. Initially, this order number is equal to the identification number of the element. Reordering changes the order number for each element. (The element identification numbers are not changed during reordering and are used in preprocessing and postprocessing.) The new order is used only during the solution phase and is transparent to the user, but can be displayed (using the **/PNUM**,LOC command). Reordering can be accomplished in one of three ways:

14.5.1. Reordering Based on Topology with a Program-Defined Starting Surface

This sorting algorithm is used by default, requiring no explicit action by the user. The sorting may also be accessed by initiating the reordering (**WAVES** command), but without a wave starting list (**WSTART** command). The starting surface is defined by the program using a graph theory algorithm (Hoit and Wilson([99] (p. 926)), Cuthill and McKee([100] (p. 926)), Georges and McIntyre([101] (p. 926))). The automatic algorithm defines a set of accumulated nodal and element weights as suggested by Hoit and Wilson([99] (p. 926)). These accumulated nodal and element weights are then used to develop the element ordering scheme.

14.5.2. Reordering Based on Topology with a User- Defined Starting Surface

This sorting algorithm is initiated (using the **WAVES** command) and uses a starting surface (input on the **WSTART** command), and then possibly is guided by other surfaces (also input on the **WSTART** command). These surfaces, as required by the algorithm, consist of lists of nodes (wave lists) which are used to start and stop the ordering process. The steps taken by the program are:

1. Define each coupled node set and constraint equation as an element.

2. Bring in wave list (defined on **WSTART** command).

3. Define candidate elements (elements having nodes in present wave list, but not in any other wave list).
4. If no candidate elements were found, go to step 2 and start again for next wave list. If no more wave lists, then stop.
5. Find the best candidate based on:
    a. element that brings in the least number of new nodes (nodes not in present wave list) - Subset A of candidate elements.
    b. if Subset A has more than one element, then element from Subset A on the surface of the model - Subset B of candidate elements.
    c. if Subset B has more than one element, then element from Subset B with the lowest element number.
6. Remove processed nodes from wave list and include new nodes from best candidate.
7. If best candidate element is not a coupled node set or constraint equation, then save element.
8. Repeat steps 3 to 7 until all elements have been processed.

Restrictions on the use of reordering based on topology are:

1. Master DOFs and imposed displacement conditions are not considered.
2. Any discontinuous models must have at least one node from each part included in a list.

14.5.3. Reordering Based on Geometry

This sorting algorithm (accessed with the **WSORT** command) is performed by a sweep through the element centroids along one of the three global or local axes, either in the positive or negative direction.

14.5.4. Automatic Reordering

If no reordering was explicitly requested (accessed with the **NOORDER** command), models are automatically reordered before solution. Both methods outlined in Reordering Based on Topology with a Program-Defined Starting Surface (p. 684) and Reordering Based on Geometry (p. 685) (in three positive directions) are used, and the optimal ordering is implemented.

14.6. Automatic Time Stepping

The method of automatic time stepping (or automatic loading) is one in which the time step size and/or the applied loads are automatically determined in response to the current state of the analysis under consideration. This method (accessed with **AUTOTS**,ON) may be applied to structural, thermal, electric, and magnetic analyses that are performed in the time domain (using the **TIME** command), and includes static (or steady state) (**ANTYPE**,STATIC) and dynamic (or transient) (**ANTYPE**,TRANS) situations.

An important point to be made here is that automatic loading always works through the adjustment of the time step size; and that the loads that are applied are automatically adjusted if ramped boundary conditions are activated (using **KBC**,0). In other words the time step size is always subjected to possible adjustment when automatic loading is engaged. Applied loads and boundary conditions, however, will vary according to how they are applied and whether the boundary conditions are stepped or ramped. That is why this method may also be thought of as automatic loading.

There are two important features of the automatic time stepping algorithm. The first feature concerns the ability to estimate the next time step size, based on current and past analysis conditions, and make proper load adjustments. In other words, given conditions at the current time, t_n, and the previous time increment, Δt_n, the primary aim is to determine the next time increment, Δt_{n+1}. Since the determination of Δt_{n+1} is largely predictive, this part of the automatic time stepping algorithm is referred to as the *time step prediction* component.

The second feature of automatic time stepping is referred to as the time step bisection component. Its purpose is to decide whether or not to reduce the present time step size, Δt_n, and redo the substep with a smaller step size. For example, working from the last converged solution at time point t_{n-1}, the present solution begins with a predicted time step, Δt_n. Equilibrium iterations are performed; and if proper convergence is either not achieved or not anticipated, this time step is reduced to $\Delta t_n/2$ (i.e., it is bisected), and the analysis begins again from time t_{n-1}. Multiple bisections can occur per substep for various reasons (discussed later).

14.6.1. Time Step Prediction

At a given converged solution at time, t_n, and with the previous time increment, Δt_n, the goal is to predict the appropriate time step size to use as the next substep. This step size is derived from the results of several unrelated computations and is most easily expressed as the minimization statement:

$$\Delta t_{n+1} = \min(\Delta t_{eq}, \Delta t_1, \Delta t_2, \Delta t_g, \Delta t_c, \Delta t_p, \Delta t_m) \tag{14.53}$$

where:

Δt_{eq} = time increment which is limited by the number of equilibrium iterations needed for convergence at the last converged time point. The more iterations required for convergence, the smaller the predicted time step. This is a general measure of all active nonlinearities. Increasing the maximum number of equilibrium iterations (using the **NEQIT** command) will tend to promote larger time step sizes.
Δt_1 = time increment which is limited by the response eigenvalue computation for 1st order systems (e.g., thermal transients) (input on the **TINTP** command).
Δt_2 = time increment which is limited by the response frequency computation for 2nd order systems (e.g., structural dynamics). The aim is to maintain 20 points per cycle (described below). Note when the middle step criterion is used, this criterion can be turned off.
Δt_g = time increment that represents the time point at which a gap or a nonlinear (multi-status) element will change abruptly from one condition to another (status change). KEYOPT(7) allows further control for the CONTAC elements.
Δt_c = time increment based on the allowable creep strain increment (described below).
Δt_p = time increment based on the allowable plastic strain increment. The limit is set at 5% per time step (described below).
Δt_m = time increment which is limited by the middle step residual tolerance (described below) for 2nd order systems (e.g., structural dynamics) (input on the **MIDTOL** command). When it is enabled, the Δt_2 criterion can be turned off.

Several trial step sizes are calculated, and the minimum one is selected for the next time step. This predicted value is further restricted to a range of values expressed by

$$\Delta t_{n+1} \le \min(F\Delta t_n, \Delta t_{max}) \tag{14.54}$$

and

$$\Delta t_{n+1} \geq \max(\Delta t_n / F, \Delta t_{min}) \quad (14.55)$$

where:

F = increase/decrease factor. F = 2, if static analysis; F = 3, if dynamic (see the **ANTYPE** and **TIMINT** commands)
Δt_{max} = maximum time step size (*DTMAX* from the **DELTIM** command or the equivalent quantity calculated from the **NSUBST** command)
Δt_{min} = minimum time step size (*DTMIN* from the **DELTIM** command or the equivalent quantity calculated from the **NSUBST** command)

In other words, the current time step is increased or decreased by at most a factor of 2 (or 3 if dynamic), and it may not be less than Δt_{min} or greater than Δt_{max}.

14.6.2. Time Step Bisection

When bisection occurs, the current substep solution (Δt_n) is removed, and the time step size is reduced by 50%. If applied loads are ramped (**KBC**,0), then the current load increment is also reduced by the same amount. One or more bisections can take place for several reasons, namely:

1. The number of equilibrium iterations used for this substep exceeds the number allowed (**NEQIT** command).
2. It appears likely that all equilibrium iterations will be used.
3. A *negative pivot* message was encountered in the solution, suggesting instability.
4. The largest calculated displacement DOF exceeds the limit (*DLIM* on the **NCNV** command).
5. An illegal element distortion is detected (e.g., negative radius in an axisymmetric analysis).
6. For transient structural dynamics, when the middle step residual is greater than the given tolerance. This check is done only when the middle step residual check is enabled by the **MIDTOL** command.

More than one bisection may be performed per substep. However, bisection of the time-step size is limited by the minimum size (defined by *DTMIN* input on the **DELTIM** command or the equivalent **NSUBST** input).

14.6.3. The Response Eigenvalue for 1st Order Transients

The response eigenvalue is used in the computation of Δt_1 and is defined as:

$$\lambda_r = \frac{\{\Delta u\}^T [K^T] \{\Delta u\}}{\{\Delta u\}^T [C] \{\Delta u\}} \quad (14.56)$$

where:

λ_r = response eigenvalue (item RESEIG for POST26 **SOLU** command and ***GET** command)
$\{\Delta u\}$ = substep solution vector (t_{n-1} to t_n)
$[K^T]$ = the Dirichlet matrix. In a heat transfer or an electrical conduction analysis this matrix is referred to as the conductivity matrix; in magnetics this is called the magnetic "stiffness". The superscript T denotes the use of a tangent matrix in nonlinear situations
[C] = the damping matrix. In heat transfer this is called the specific heat matrix.

The product of the response eigenvalue and the previous time step (Δt_n) has been employed by Hughes([145] (p. 929)) for the evaluation of 1st order explicit/implicit systems. In Hughes([145] (p. 929)) the quantity $\Delta t_n \lambda$ is referred to as the "oscillation limit", where λ is the maximum eigenvalue. For unconditionally stable systems, the primary restriction on time-step size is that the inequality $\Delta t_n \lambda >> 1$ should be avoided. Hence it is very conservative to propose that $\Delta t_n \lambda = 1$.

Since the time integration used employs the trapezoidal rule (Equation 15.40 (p. 776)), all analyses of 1st order systems are unconditionally stable. The response eigenvalue supplied by means of Equation 14.56 (p. 687) represents the dominate eigenvalue and not the maximum; and the time-step restriction above is restated as:

$$\Delta t_n \lambda_r \cong f \qquad (f < 1) \tag{14.57}$$

This equation expresses the primary aim of automatic time stepping for 1st order transient analyses. The quantity $\Delta t_n \lambda_r$ appears as the oscillation limit output during automatic loading. The default is f = 1/2, and can be changed (using *OSLM* and *TOL* on the **TINTP** command). The quantity Δt_1 is approximated as:

$$\frac{\Delta t_1}{\Delta t_n} = \frac{f}{\lambda_r \Delta t_n} \tag{14.58}$$

14.6.4. The Response Frequency for Structural Dynamics

The response frequency is used in the computation of Δt_2 and is defined as (Bergan([105] (p. 926))):

$$f_r^2 = \frac{\{\Delta u\}^T [K^T] \{\Delta u\}}{(2\pi)^2 \{\Delta u\}^T [M] \{\Delta u\}} \tag{14.59}$$

where:

f_r = response frequency (item RESFRQ for POST26 **SOLU** command and ***GET** command)
$\{\Delta u\}$ = substep solution vector (t_{n-1} to t_n)
$[K^T]$ = tangent stiffness matrix
[M] = mass matrix

This equation is a nonlinear form of Rayleigh's quotient. The related response period is:

$$T_r = 1/f_r \tag{14.60}$$

Using T_r the time increment limited by the response frequency is:

$$\Delta t_2 = T_r / 20 \tag{14.61}$$

When the middle step criterion is used, this criterion can be turned off.

14.6.5. Creep Time Increment

The time step size may be increased or decreased by comparing the value of the creep ratio C_{max} (Rate-Dependent Plasticity (Including Creep and Viscoplasticity) (p. 105)) to the creep criterion C_{cr}. C_{cr} is equal to .10 unless it is redefined (using the **CRPLIM** command). The time step estimate is computed as:

$$\Delta t_c = \Delta t_n \frac{C_{cr}}{C_{max}} \tag{14.62}$$

Δt_c is used in Equation 14.53 (p. 686) only if it differs from Δt_n by more than 10%.

14.6.6. Plasticity Time Increment

The time step size is increased or decreased by comparing the value of the effective plastic strain increment $\Delta\tilde{\varepsilon}_n^{pl}$ (Equation 4.26 (p. 73)) to 0.05 (5%). The time step estimate is computed as:

$$\Delta t_p = \Delta t_n \frac{.05}{\Delta\tilde{\varepsilon}_n^{pl}} \tag{14.63}$$

Δt_p is used in Equation 14.53 (p. 686) only if it differs from Δt_n by more than 10%.

14.6.7. Midstep Residual for Structural Dynamic Analysis

The midstep residual is used in the computation of Δt_m. The midstep residual for the determination of the time step is based on the following consideration. The solution of the structural dynamic analysis is carried out at the discrete time points, and the solution at the intermediate time remains unknown. However, if the time step is small enough, the solution at the intermediate time should be close enough to an interpolation between the beginning and end of the time step. If so, the unbalanced residual from the interpolation should be small. On the other hand, if the time step is large, the interpolation will be very different from the true solution, which will lead to an unbalanced residual that is too large. The time step is chosen to satisfy the criterion set by the user (e.g. **MIDTOL** command).

Refer to the discussion in Newton-Raphson Procedure (p. 711). The residual force at any time between the time step n and n+1 can be written as:

$$\{R\} = \{F_{n+\nu}^a\} - \{F_{n+\nu}^{nr}\} \tag{14.64}$$

where:

ν = intermediate state between the time step n and n+1 ($0 < \nu < 1$)
$\{R\}$ = residual force vector
$\{F_{n+\nu}^a\}$ = vector of the applied load at $n + \nu$
$\{F_{n+\nu}^{nr}\}$ = vector of the restoring load corresponding to the element internal load at $n + \nu$, which depends on the intermediate state of displacement at $n + \nu$, and also the velocity and acceleration at $n + \nu$. This intermediate state is approximately calculated based on the Newmark assumption.

A measure of the magnitude of $\{R\}$ is established in a manner similar to the convergence check at the end of the time step (see Convergence (p. 716)). After the solution has converged at the end of the time step (n+1), the midstep residual force is compared to the reference value:

$$\varepsilon = \frac{\|\{R\}\|}{R_{ref}} \tag{14.65}$$

where:

$\|\{R\}\|$ = magnitude (vector norm) of residual force vector
R_{ref} = reference force (see Convergence (p. 716))

The convergence criterion for the midstep residual is defined by the value of τ_b (input as *TOLERB* on **MIDTOL** command):

If $\tau_b > 0$, the value is used as a tolerance. If $\tau_b = 0$ is specified or τ_b is not specified, then a default positive value is used as a tolerance. The midstep residual is assumed to have converged if its value is within the desired tolerance ($\varepsilon \leq \tau_b$). Depending on how well the convergence criterion is satisfied the time step size for the next increment is increased or kept unchanged.

If the midstep residual hasn't converged ($\varepsilon > \tau_b$), the time step is repeated with a smaller increment:

$$\Delta t_m^b = \Delta t_n \frac{\tau_b}{\varepsilon} \tag{14.66}$$

where:

Δt_m^b = new (bisected) time step size
Δt_n = old time step size
τ_b = midstep residual tolerance(*TOLERB* on **MIDTOL** command)

If $\tau_b < 0$, the value is used as a reference force (reference moment is computed from reference force value) for midstep convergence check. A procedure similar to the one described above is followed with modified definition of time step size:

$$\Delta t_m^b = \Delta t_n \frac{|\tau_b|}{\|\{R\}\|} \tag{14.67}$$

14.7. Solving for Unknowns and Reactions

In general, the equations that are solved for static linear analyses are:

$$[K]\{u\} = \{F\} \tag{14.68}$$

or

$$[K]\{u\} = \{F^a\} + \{F^r\} \tag{14.69}$$

where:

[K] = total stiffness or conductivity matrix = $\sum_{m=1}^{N} [K_e]$
{u} = nodal degree of freedom (DOF) vector
N = number of elements
$[K_e]$ = element stiffness or conductivity matrix
$\{F^r\}$ = nodal reaction load vector

$\{F^a\}$, the total applied load vector, is defined by:

$$\{F^a\} = \{F^{nd}\} + \{F^e\} \tag{14.70}$$

where:

$\{F^{nd}\}$ = applied nodal load vector

$\{F^{e}\}$ = total of all element load vector effects (pressure, acceleration, thermal, gravity)

Equation 14.68 (p. 690) thru Equation 14.70 (p. 690) are similar to Equation 15.1 (p. 762) thru Equation 15.4 (p. 763).

If sufficient boundary conditions are specified on {u} to guarantee a unique solution, Equation 14.68 (p. 690) can be solved to obtain the node DOF values at each node in the model.

Rewriting Equation 14.69 (p. 690) for linear analyses by separating out the matrix and vectors into those DOFs with and without imposed values,

$$\begin{bmatrix} [K_{cc}] & [K_{cs}] \\ [K_{cs}]^T & [K_{ss}] \end{bmatrix} \begin{Bmatrix} \{u_c\} \\ \{u_s\} \end{Bmatrix} = \begin{Bmatrix} \{F_c^a\} \\ \{F_s^a\} \end{Bmatrix} + \begin{Bmatrix} \{F_c^r\} \\ \{F_s^r\} \end{Bmatrix} \tag{14.71}$$

where:

s = subscript representing DOFs with imposed values (specified DOFs)
c = subscript representing DOFs without imposed values (computed DOFs)

Note that $\{u_s\}$ is known, but not necessarily equal to {0}. Since the reactions at DOFs without imposed values must be zero, Equation 14.71 (p. 691) can be written as:

$$\begin{bmatrix} [K_{cc}] & [K_{cs}] \\ [K_{cs}]^T & [K_{ss}] \end{bmatrix} \begin{Bmatrix} \{u_c\} \\ \{u_s\} \end{Bmatrix} = \begin{Bmatrix} \{F_c^a\} \\ \{F_s^a\} \end{Bmatrix} + \begin{Bmatrix} \{0\} \\ \{F_s^r\} \end{Bmatrix} \tag{14.72}$$

The top part of Equation 14.72 (p. 691) may be solved for $\{u_c\}$:

$$\{u_c\} = [K_{cc}]^{-1}(-[K_{cs}]\{u_s\} + \{F_c^a\}) \tag{14.73}$$

The actual numerical solution process is not as indicated here but is done more efficiently using one of the various equation solvers discussed in Equation Solvers (p. 694).

14.7.1. Reaction Forces

The reaction vector $\{F_s^r\}$, may be developed for linear models from the bottom part of Equation 14.72 (p. 691):

$$\{F_s^r\} = [K_{cs}]^T\{u_c\} + [K_{ss}]\{u_s\} - \{F_s^a\} \tag{14.74}$$

where:

$\{F_s^r\}$ = reaction forces (output using either **OUTPR**,RSOL or **PRRSOL** command)

Alternatively, the nodal reaction load vector may be considered over all DOFs by combining Equation 14.69 (p. 690) and Equation 14.70 (p. 690) to get:

$$\{F^r\} = [K]\{u\} - \{F^{nd}\} - \{F^e\} \tag{14.75}$$

where only the loads at imposed DOF are output. Where applicable, the transient/dynamic effects are added:

$$\{F^r\} = [M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} - \{F^{nd}\} - \{F^e\} \quad (14.76)$$

where:

[M] = total mass matrix
[C] = total damping or conductivity matrix
$\{\dot{u}\}$, $\{\ddot{u}\}$ = defined below

The element static nodal loads are:

$$\{F_e^K\} = -[K_e]\{u\} + \{F^e\} \quad (14.77)$$

where:

$\{F_e^k\}$ = element nodal loads (output using **OUTPR**,NLOAD, or **PRESOL** commands)
e = subscript for element matrices and load vectors

The element damping and inertial loads are:

$$\{F_e^c\} = -[C_e]\{\dot{u}\} \quad (14.78)$$

$$\{F_e^m\} = -[M_e]\{\ddot{u}\} \quad (14.79)$$

where:

$\{F_e^D\}$ = element damping nodal load (output using **OUTPR**,NLOAD, or **PRESOL** commands)

$\{F_e^I\}$ = element inertial nodal load (output using **OUTPR**,NLOAD, or **PRESOL** commands)

Thus,

$$\{F^r\} = -\sum_{e=1}^{N} [\{F_e^K\} + \{F_e^c\} + \{F_e^m\}] - \{F^{nd}\} \quad (14.80)$$

The derivatives of the nodal DOF with respect to time are:

$\{\dot{u}\}$ = first derivative of the nodal DOF with respect to time, e.g., velocity

$\{\ddot{u}\}$ = second derivative of the nodal DOF with respect to time, e.g., acceleration

Transient Analysis (p. 763) and Harmonic Analysis (p. 780) discuss the transient and harmonic damping and inertia loads.

If an imposed DOF value is part of a constraint equation, the nodal reaction load vector is further modified using the appropriate terms of the right hand side of Equation 14.188 (p. 725); that is, the forces on the non-unique DOFs are summed into the unique DOF (the one with the imposed DOF value) to give the total reaction force acting on that DOF.

14.7.2. Disequilibrium

The following circumstances could cause a disequilibrium, usually a moment disequilibrium:

Program Option	Explanation of Possible Difficulty
non-planar, 4-node membrane shell elements SHELL41 SHELL181 with KEYOPT(1) = 1	If the 4 nodes do not lie in a flat plane moment equilibrium may not be preserved, as no internal corrections are done. However, the program requires such elements to be input very close to flat.
nodal coupling constraint equations (**CP**, **CE** commands)	The user can write any form of relationship between the displacements, and these may include fictitious forces or moments. Thus, the reaction forces printout can be used to detect input errors.
MATRIX27 User generated superelement matrix	The user has the option to input almost any type of erroneous input, so that such input should be checked carefully. For example, all terms representing UX degrees of freedom of one UX row of the matrix should sum to zero to preserve equilibrium.
COMBIN37 FLUID38 COMBIN39 COMBIN40	Noncoincident nodes can cause a moment disequilibrium. (This is usually not a problem if one of the nodes is attached to a non-rotating ground).
COMBIN14 (with KEYOPT(2) > 0) MATRIX27 COMBIN37 FLUID38 COMBIN39 COMBIN40	Elements with one node having a different nodal coordinate system from the other are inconsistent.

The following circumstances could cause an apparent disequilibrium:

- All nodal coordinate systems are not parallel to the global Cartesian coordinate system. However, if all nodal forces are rotated to the global Cartesian coordinate system, equilibrium should be seen to be satisfied.
- The solution is not converged. This applies to the potential discrepancy between applied and internal element forces in a nonlinear analysis.
- The mesh is too coarse. This may manifest itself for elements where there is an element force printout at the nodes, such as SHELL61 (axisymmetric-harmonic structural shell).
- The "TOTAL" of the moments (MX, MY, MZ) given with the reaction forces does not necessarily represent equilibrium. It only represents the sum of all applicable moments. Moment equilibrium would also need the effects of forces taken about an arbitrary point.
- Axisymmetric models are used with forces or pressures with a radial component. These loads will often be partially equilibrated by hoop stresses, which do not show up in the reaction forces.
- Shell elements have an elastic foundation described. The load carried by the elastic foundation is not seen in the reaction forces.

- In substructure expansion pass with the resolve method used, the reaction forces at the master degree of freedom are different from that given by the backsubstitution method (see Substructuring Analysis (p. 793)).

14.8. Equation Solvers

The system of simultaneous linear equations generated by the finite element procedure is solved either using a direct elimination process or an iterative method. A direct elimination process is primarily a Gaussian elimination approach which involves solving for the unknown vector of variables {u} in Equation 14.81 (p. 694):

$$[K]\{u\} = \{F\} \tag{14.81}$$

where:

[K] = global stiffness/conductivity matrix
{u} = global vector of nodal unknown
{F} = global applied load vector

The direct elimination process involves decomposition (factorization) of the matrix [K] into lower and upper triangular matrices, [K] = [L][U]. Then forward and back substitutions using [L] and [U] are made to compute the solution vector {u}.

A typical iterative method involves an initial guess, $\{u\}_1$, of the solution vector {u} and then a successive steps of iteration leading to a sequence of vectors $\{u\}_2$, $\{u\}_3$, . . . such that, in the limit, $\{u\}_n = \{u\}$ as n tends to infinity. The calculation of $\{u\}_{n+1}$ involves [K], {F}, and the {u} vectors from one or two of the previous iterations. Typically the solution converges to within a specified tolerance after a finite number of iterations.

In the following sections, all of the solvers are described under two major subsections: Direct Solvers and Iterative Solvers (all accessed with **EQSLV**).

14.8.1. Direct Solvers

The direct solver that is available is the Sparse Direct Solver (accessed with **EQSLV**,SPARSE). The Sparse Direct Solver makes use of the fact that the finite element matrices are normally sparsely populated. This sparsity allows the system of simultaneous equations to be solved efficiently by minimizing the operation counts.

14.8.2. Sparse Direct Solver

As described in the introductory section, the linear matrix equation, (Equation 14.81 (p. 694)) is solved by triangular decomposition of matrix [K] to yield the following equation:

$$[L][U]\{u\} = \{F\} \tag{14.82}$$

where:

[L] = lower triangular matrix
[U] = upper triangular matrix

By substituting:

$$\{w\} = [U]\{u\} \tag{14.83}$$

we can obtain {u} by first solving the triangular matrix system for {w} by using the forward pass operation given by:

$$[L]\{w\} = \{F\} \tag{14.84}$$

and then computing {u} using the back substitution operation on a triangular matrix given by:

$$[U]\{u\} = \{w\} \tag{14.85}$$

When [K] is symmetric, the above procedure could use the substitution:

$$[K] = [L][L]^T \tag{14.86}$$

However, it is modified as:

$$[K] = [L'][D][L']^T \tag{14.87}$$

where:

[D] = a diagonal matrix

The diagonal terms of [D] may be negative in the case of some nonlinear finite element analysis. This allows the generation of [L'] without the consideration of a square root of negative number. Therefore, Equation 14.82 (p. 694) through Equation 14.85 (p. 695) become:

$$[L'][D][L']^T \{u\} = \{F\} \tag{14.88}$$

$$\{w\} = [D][L']^T \{u\} \tag{14.89}$$

$$[L']\{w\} = \{F\} \tag{14.90}$$

and

$$[D][L']^T \{u\} = \{F\} \tag{14.91}$$

Since [K] is normally sparsely populated with coefficients dominantly located around the main diagonal, the Sparse Direct Solver is designed to handle only the nonzero entries in [K]. In general, during the Cholesky decomposition of [K] shown in Equation 14.82 (p. 694) or Equation 14.88 (p. 695), nonzero coefficients appear in [L] or [L'] at coefficient locations where [K] matrix had zero entries. The Sparse Direct Solver algorithm minimizes this fill-in by judiciously reordering the equation numbers in [K].

The performance of a direct solution method is greatly optimized through the equations reordering procedure which involves relabeling of the variables in the vector {u}. This simply amounts to permuting the rows and columns of [K] and the rows of {F} with the objective of minimizing fill-in. So, when the decomposition step in Equation 14.82 (p. 694) or Equation 14.88 (p. 695) is performed on the reordered [K] matrix, the fill-in that occurs in [L] or [L'] matrix is kept to a minimum. This enormously contributes to optimizing the performance of the Sparse Direct Solver.

To achieve minimum fill-in, different matrix coefficient reordering algorithms are available in the literature (George and Liu([302] (p. 938))). The Sparse Direct Solver uses two different reordering schemes. They are the Minimum Degree ordering and the METIS ordering. The choice of which reordering method to use is automated in the solver algorithm in order to yield the least fill-in.

14.8.3. Iterative Solver

The ANSYS program offers a large number of iterative solvers as alternatives to the direct solvers (sparse solver). These alternatives in many cases can result in less I/O or disk usage, less total elapsed time, and more scalable parallel performance. However, in general, iterative solvers are not as robust as the direct solvers. For numerical challenges such as a nearly-singular matrix (matrix with small pivots) or a matrix that includes Lagrangian multipliers, the direct solver is an effective solution tool, while an iterative solver is less effective or may even fail.

The first three iterative solvers are based on the conjugate gradient (CG) method. The first of these three CG solvers is the Jacobi Conjugate Gradient (JCG) solver (Mahinthakumar and Hoole ([144] (p. 929))) (accessed with **EQSLV**,JCG) which is suitable for well-conditioned problems. Well-conditioned problems often arise from heat transfer, acoustics, magnetics and solid 2-D / 3-D structural analyses. The JCG solver is available for real and complex symmetric and unsymmetric matrices. The second solver is the Preconditioned Conjugate Gradient (PCG) solver (accessed with **EQSLV**,PCG) which is efficient and reliable for all types of analyses including the ill-conditioned beam/shell structural analysis. The PCG solver is made available through a license from Computational Applications and System Integration, Inc. of Champaign, Illinois (USA). The PCG solver is only valid for real symmetric stiffness matrices. The third solver is the Incomplete Cholesky Conjugate Gradient (ICCG) solver (internally developed, unpublished work) (accessed with **EQSLV**,ICCG). The ICCG solver is more robust than the JCG solver for handling ill-conditioned matrices. The ICCG solver is available for real and complex, symmetric and unsymmetric matrices.

The typical system of equations to be solved iteratively is given as :

$$[K]\{u\} = \{F\} \tag{14.92}$$

where:

[K] = global coefficient matrix
{u} = unknown vector
{F} = global load vector

In the CG method, the solution is found as a series of vectors $\{p_i\}$:

$$\{u\} = \alpha_1\{p_1\} + \alpha_2\{p_2\} + \ldots + \alpha_m\{p_m\} \tag{14.93}$$

where m is no larger than the matrix size n. The scheme is guaranteed to converge in n or fewer iterations on an infinite precision machine. However, since the scheme is implemented on a machine with finite precision, it sometimes requires more than n iterations to converge. The solvers allow up to a maximum of 2n iterations. If it still does not converge after the 2n iterations, the solution will be abandoned with an error message. The unconverged situation is often due to an inadequate number of boundary constraints being used (rigid body motion). The rate of convergence of the CG algorithm is proportional to the square root of the conditioning number of [K] where the condition number of [K] is equal to the ratio of the maximum eigenvalue of [K] to the minimum eigenvalue of [K] . A preconditioning procedure is used to reduce the condition number of linear Equation 14.92 (p. 696). In the JCG algorithm, the diagonal terms of [K] are used as the preconditioner [Q], while in the ICCG and PCG algorithms, a more sophisticated preconditioner [Q] is used. The CG algorithm with preconditioning is shown collectively as Equation 14.94 (p. 697).

$\{u_0\} = \{0\}$

$\{R_0\} = \{F\}$

$\{z_0\} = [Q]^{-1}\{F\}$

Do i=1, n

If (Norm(R) $\le \varepsilon^2$) then

set$\{u\} = \{u_{i-1}\}$

quit loop

Else

If(i=1)then

$\beta_1 = 0$

$\{p_1\} = \{R_0\}$

$$\alpha_1 = \frac{\{z_0\}^T\{R_0\}}{\{p_1\}^T[K]\{p_1\}} \qquad (14.94)$$

$\{R_1\} = \{R_0\} - \alpha_1[K]\{p_1\}$

Else

Applying preconditioning:$\{z_{i-1}\} = [Q]^{-1}\{R_{i-1}\}$

$$\beta_i \frac{\{z_{i-1}\}^T\{R_{i-1}\}}{\{z_{i-2}\}^T\{R_{i-2}\}}$$

$\{p_i\} = \{z_{i-1}\} + \beta_i\{p_{i-1}\}$

$$\alpha_i = \frac{\{z_{i-1}\}^T\{R_{i-1}\}}{\{p_i\}^T[K]\{p_i\}}$$

$\{R_i\} = \{R_{i-1}\} - \alpha_i[K]\{p_i\}$

Endif

Endif

End loop

Convergence is achieved when:

$$\frac{\{R_i\}^T\{R_i\}}{\{F\}^T\{F\}} \le \varepsilon^2 \qquad (14.95)$$

where:

ε = user supplied tolerance (*TOLER* on the **EQSLV** command; output as SPECIFIED TOLERANCE)
$\{R_i\}$ = {F} - [K] $\{u_i\}$
$\{u_i\}$ = solution vector at iteration i

also, for the JCG and ICCG solvers:

$\{R_i\}^T\{R_i\}$ = output as CALCULATED NORM (14.96)

$\{F\}^T\{F\}\varepsilon^2$ = output as TARGET NORM (14.97)

It is assumed that the initial starting vector $\{u_0\}$ is a zero vector.

14.9. Mode Superposition Method

Mode superposition method is a method of using the natural frequencies and mode shapes from the modal analysis (**ANTYPE**,MODAL) to characterize the dynamic response of a structure to transient (**ANTYPE**,TRANS with **TRNOPT**,MSUP, Transient Analysis (p. 763)), or steady harmonic (**ANTYPE**,HARM with **HROPT**,MSUP, Harmonic Analysis (p. 780)) excitations.

The following topics are available:

14.9.1. General Equations

The equations of motion may be expressed as in Equation 15.5 (p. 764):

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F\} \tag{14.98}$$

{F} is the time-varying load vector, given by

$$\{F\} = \{F^{nd}\} + s\{F^{s}\} \tag{14.99}$$

where:

$\{F^{nd}\}$ = time varying nodal forces
s = load vector scale factor (input on **LVSCALE** command)
$\{F^{s}\}$ = load vector from the modal analysis (see below)

The load vector $\{F^{s}\}$ is computed when doing a modal analysis and its generation is the same as for a substructure load vector, described in Substructuring Analysis (p. 793).

The following development is similar to that given by Bathe([2] (p. 921)):

Define a set of modal coordinates y_i such that

$$\{u\} = \sum_{i=1}^{n} \{\phi_i\} y_i \tag{14.100}$$

where:

$\{\Phi_i\}$ = the i^{th} mode shape
n = the number of modes to be used (input as *MAXMODE* on **TRNOPT** or **HROPT** commands)

Note that Equation 14.100 (p. 698) hinders the use of nonzero displacement input, since defining y_i in terms of {u} is not straight forward. The inverse relationship does exist (Equation 14.100 (p. 698)) for the case where all the displacements are known, but not when only some are known. Substituting Equation 14.100 (p. 698) into Equation 14.98 (p. 698),

$$[M]\sum_{i=1}^{n} \{\phi_i\}\ddot{y}_i + [C]\sum_{i=1}^{n} \{\phi_i\}\dot{y}_i + [K]\sum_{i=1}^{n} \{\phi_i\} y_i = \{F\} \tag{14.101}$$

Premultiply by a typical mode shape $\{\Phi_i\}^T$:

$$\{\phi_j\}^T[M]\sum_{i=1}^{n}\{\phi_i\}\ddot{y}_i + \{\phi_j\}^T[C]\sum_{i=1}^{n}\{\phi_i\}\dot{y}_i + \{\phi_j\}^T[K]\sum_{i=1}^{n}\{\phi_i\}y_i = \{\phi_j\}^T\{F\} \tag{14.102}$$

The orthogonal condition of the natural modes states that

$$\{\phi_j\}^T[M]\{\phi_i\} = 0 \qquad i \neq j \tag{14.103}$$

$$\{\phi_j\}^T[K]\{\phi_i\} = 0\, i \neq j \tag{14.104}$$

In the mode superposition method using the Lanczos and other extraction methods, only Rayleigh or constant damping is allowed so that:

$$\{\phi_j\}^T[C]\{\phi_i\} = 0 \qquad i \neq j \tag{14.105}$$

Applying these conditions to Equation 14.102 (p. 699), only the i = j terms remain:

$$\{\phi_j\}^T[M]\{\phi_j\}\ddot{y}_j + \{\phi_j\}^T[C]\{\phi_j\}\dot{y}_j + \{\phi_j\}^T[K]\{\phi_j\}y_j = \{\phi_j\}^T\{F\} \tag{14.106}$$

The coefficients of $\ddot{y}_j$, $\dot{y}_j$, and y_j, are derived as follows:

1. **Coefficient of** $\ddot{y}_j$:

   By the normality condition (Equation 15.51 (p. 779)),

   $$\{\phi_j\}^T[M]\{\phi_j\} = 1 \tag{14.107}$$

2. **Coefficient of** $\dot{y}_j$:

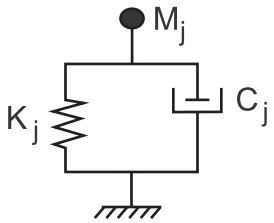
   The damping term is based on treating the modal coordinate as a single DOF system (shown in Equation 14.98 (p. 698)) for which:

   $$\{\phi_j\}^T[C]\{\phi_j\} = C_j\phi_j^2 \tag{14.108}$$

   and

   $$\{\phi_j\}^T[M]\{\phi_j\} = M_j\phi_j^2 = 1 \tag{14.109}$$

Figure 14.5: Single Degree of Freedom Oscillator



Equation 14.109 (p. 699) can give a definition of ϕ_j:

$$\phi_j = \frac{1}{\sqrt{M_j}} \tag{14.110}$$

From (Tse([68] (p. 924))),

$$C_j = 2\xi_j\sqrt{K_j M_j} \tag{14.111}$$

where:

ξ_j = fraction of critical damping for mode j

and,

$$\omega_j = \sqrt{(K_j/M_j)} \tag{14.112}$$

where:

ω_j = natural circular frequency of mode j

Combining Equation 14.110 (p. 700) thru Equation 14.107 (p. 699) with Equation 14.108 (p. 699),

$$\begin{aligned}\{\phi_j\}^T[C]\{\phi_j\} &= 2\xi_j\sqrt{K_j M_j}\left(\frac{1}{\sqrt{M_j}}\right)^2 \\ &= 2\xi_j\,\omega_j\end{aligned} \tag{14.113}$$

3. **Coefficient of** y_j:

From Equation 15.48 (p. 779),

$$[K]\{\phi_j\} = \omega_j^2[M]\{\phi_j\} \tag{14.114}$$

Premultiply by $\{\phi_j\}^T$,

$$\{\phi_j\}^T[K]\{\phi_j\} = \omega_j^2\{\phi_j\}^T[M]\{\phi_j\} \tag{14.115}$$

Substituting Equation 14.107 (p. 699) for the mass term,

$$\{\phi_j\}^T[K]\{\phi_j\} = \omega_j^2 \tag{14.116}$$

For convenient notation, let

$$f_j = \{\phi_j\}^T\{F\} \tag{14.117}$$

represent the right-hand side of Equation 14.106 (p. 699). Substituting Equation 14.107 (p. 699), Equation 14.113 (p. 700), Equation 14.116 (p. 700) and Equation 14.117 (p. 700) into Equation 14.106 (p. 699), the equation of motion of the modal coordinates is obtained:

$$\ddot{y}_j + 2\omega_j\xi_j\dot{y}_j + \omega_j^2 y_j = f_j \tag{14.118}$$

Since j represents any mode, Equation 14.118 (p. 700) represents n uncoupled equations in the n unknowns y_j. The advantage of the uncoupled system (**ANTYPE**,TRAN with **TRNOPT**,MSUP) is that all the computationally expensive matrix algebra has been done in the eigensolver, and long transients may be analyzed inexpensively in modal coordinates with Equation 14.100 (p. 698).

The y_j are converted back into geometric displacements {u} (the system response to the loading) by using Equation 14.100 (p. 698). That is, the individual modal responses y_j are superimposed to obtain the actual response, and hence the name "mode superposition".

14.9.2. Equations for QR Damped Eigensolver Based Analysis

For the QR damped mode extraction method, the general equations apply except the differential equations of motion in modal coordinate are deduced from Equation 14.208 (p. 733) with the right hand side force vector of Equation 14.102 (p. 699). They are written as:

$$[I]\{\ddot{y}\} + [\Phi]^T[C][\Phi]\{\dot{y}\} + ([\Lambda^2] + [\Phi]^T[K_{unsym}][\Phi])\{y\} = [\Phi]^T\{F\} \quad (14.119)$$

where:

$[\Phi]$ = real eigenvector matrix normalized with respect to mass coming from the LANCZOS run of QRDAMP (see QR Damped Method (p. 733) for more details).

$[\Lambda^2]$ = diagonal matrix containing the eigenvalues ω_i squared.

$[K_{unsym}]$ = unsymmetric part of the stiffness matrix.

It can be seen that if [C] is arbitrary and/or [K] is unsymmetric, the modal matrices are full so that the modal equations are coupled.

14.9.3. Equations for Unsymmetric Eigensolver Based Analysis

When using the unsymmetric eigensolver, the matrices are unsymmetric. Both left and right normalized eigenmodes are used to decouple the modal equations as follows.

Equation 14.106 (p. 699) becomes:

$$\{\phi_j^L\}^T[M]\{\phi_j\}\ddot{y}_j + \{\phi_j^L\}^T[C]\{\phi_j\}\dot{y}_j + \{\phi_j^L\}^T[K]\{\phi_j\}y_j = \{\phi_j^L\}^T\{F\} \quad (14.120)$$

where:

$\{\varphi_j^L\}$ = the j^{th} left mode shape

$\{\varphi_j\}$ = the j^{th} right mode shape (also called the j^{th} mode shape). For more information, see Unsymmetric Method (p. 729).

This equations leads to the modal coordinate equation (Equation 14.118 (p. 700)) when Rayleigh damping is considered.

14.9.4. Modal Damping

The modal damping, ξ_j, is the combination of several ANSYS damping inputs, as described in Equation 14.24 (p. 676).

14.9.5. Residual Vector Method

In modal superposition analysis, the dynamic response will be approximate when the applied loading excites the higher frequency modes of a structure. To improve the accuracy of dynamic response, the residual vector method employs additional modal transformation vectors (designated as residual vectors) in addition to the eigenvectors in the modal transformation (Equation 14.100 (p. 698)).

The residual vector method (**RESVEC** command) uses extra residual vectors computed in the modal analysis (**ANTYPE**,MODAL) to characterize the high frequency response of a structure to dynamic loading. It applies to the following subsequent mode superposition analyses:

- Transient (**ANTYPE**,TRANS with **TRNOPT**,MSUP)
- Harmonic (**ANTYPE**,HARM with **HROPT**,MSUP)
- PSD (**ANTYPE**,SPEC with **SPOPT**,PSD)
- Multiple and single point response spectrum (**ANTYPE**,SPEC with **SPOPT**,SPRS or MPRS).

Because of the improved convergence properties of this method, fewer eigenmodes are required from the eigensolution.

The dynamic response of the structure can be divided into two terms:

$$x = x_L + x_H \tag{14.121}$$

where:

x_L = lower mode contributions (Equation 14.100 (p. 698))
x_H = higher mode contributions, which can be expressed as the combination of residual vectors.

First, the flexibility matrix can be expressed as:

$$[G] = \sum_{i=1}^{n} \frac{\{\Phi\}_i \{\Phi\}_i^T}{\omega_i^2} = \sum_{i=1}^{m} \frac{\{\Phi\}_i \{\Phi\}_i^T}{\omega_i^2} + \sum_{i=m+1}^{n} \frac{\{\Phi\}_i \{\Phi\}_i^T}{\omega_i^2} \tag{14.122}$$

where:

[G] = generalized inverse matrix of stiffness matrix K (see Geradin and Rixen [368] (p. 941))
$\{\Phi\}_i$ = elastic normal modes
n = total degree of freedom of the system

$$i = \begin{cases} 1, m & \text{retained elastic normal modes from modal analysis (eigenmodes extracted in modal analysis)} \\ m+1,\ n & \text{truncated elastic normal modes of the structure} \end{cases}$$

The residual flexibility matrix is given by:

$$[\tilde{G}] = \sum_{i=m+1}^{n} \frac{\{\Phi\}_i \{\Phi\}_i^T}{\omega_i^2} = [G] - \sum_{i=1}^{m} \frac{\{\Phi\}_i \{\Phi\}_i^T}{\omega_i^2} \tag{14.123}$$

Define residual vectors as:

$$[R] = [\tilde{G}][F] \tag{14.124}$$

where:

[F] = matrix of force vectors

Orthogonalize the residual vectors with respect to the retained elastic normal modes gives orthogonalized residual vectors $\{\Phi^R\}_j$, that is, pseudo-modes with associated frequencies that are orthogonal to the mass and stiffness matrices (see Dickens et al. [423] (p. 944)).

Then the basis vectors for modal subspace are formed by:

$$[\Phi] = \left[\{\Phi\}_{i=1,m}; \{\Phi^R\}_{j=1,k} \right] \tag{14.125}$$

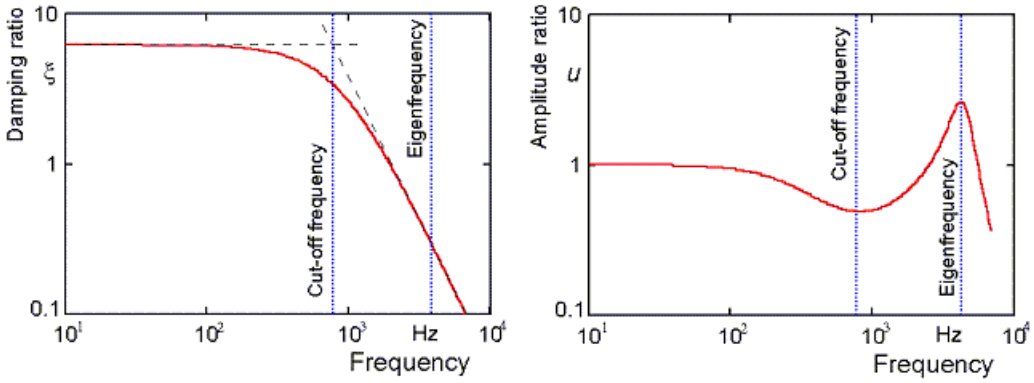
which will be used in modal superposition transient and harmonic analysis.

14.10. Extraction of Modal Damping Parameter for Squeeze Film Problems

A constant damping ratio is often applied for harmonic analysis. In practice this approach only leads to satisfying results if all frequency steps can be represented by the same damping ratio or the operating range encloses just one eigenmode. Difficulties arise if the damping ratio depends strongly on the excitation frequency as happens in case of viscous damping in gaseous environment.

A typical damping ratio verse frequency function is shown below. For this example, the damping ratio is almost constant below the cut-off frequency. Harmonic oscillations at frequencies below cutoff are strongly damped. Above cut-off the damping ratio decreases. Close to the structural eigenfrequency the damping ratio dropped down to about 0.25 and a clear resonance peak can be observed.

Figure 14.6: Damping and Amplitude Ratio vs. Frequency



Damping and stiffness coefficients in modal coordinates are defined based on their nodal coordinate values as:

$$[C] = [\Phi]^T [C^*][\Phi] \tag{14.126}$$

and

$$[K] = [\Phi]^T [K^*][\Phi] \tag{14.127}$$

where:

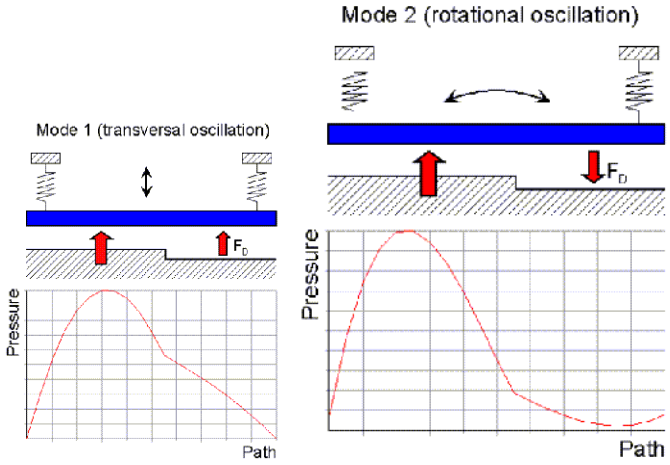
[C] = damping coefficient in modal coordinates

$[\Phi] = [\{\phi_1\}\{\phi_2\}\dots\{\phi_n\}]$

$\{\varphi_i\}$ = eigenvector i (in modal coordinates)
[C*] = finite element damping matrix in modal coordinates
[K*] = finite element stiffness matrix in nodal coordinates

Unfortunately, both matrices [C*] and [K*] are not directly available for the fluid part of the coupled domain problem (e.g., squeeze film elements FLUID136). Moreover eigenvectors are derived from the structural part of the coupled domain problem and consequently neither the modal damping matrix nor the modal stiffness matrix of the fluidic system are necessarily orthogonal. Essential off-diagonal elements occur in case of asymmetric film arrangements or asymmetric plate motion as shown below.

Figure 14.7: Fluid Pressure From Modal Excitation Distribution



The goal is to express the viscous damping in modal coordinates as follows:

$$[C]\{\dot{q}\} + [K]\{q\} = \{F\} \tag{14.128}$$

where:

{F} = modal force vector
{q} = vector of modal displacement amplitudes
$\{\dot{q}\}$ = vector of modal velocity amplitudes
[C] = unknown modal damping matrix
[K] = unknown modal squeeze stiffness matrix

The following algorithm is necessary to compute all coefficients of the modal damping and stiffness matrix:

1. Start with the first mode and excite the fluid elements by wall velocities which correspond to a unit modal velocity. In fact the nodal velocities become equal to the eigenvector of the appropriate mode.

2. Compute the real and imaginary part of the pressure distribution in a harmonic analyses.

3. Compute modal forces with regard to all other modes. The i^{th} modal force states how much the pressure distribution of the first mode really acts on the i^{th} mode.

4. The computed modal forces can be used to extract all damping and squeeze stiffness coefficients of the first column in the [C] and [K] matrices.

5. Repeat step 1 with the next eigenvector and compute the next column of [C] and [K].

The theoretical background is given by the following equations. Each coefficient C_{ji} and K_{ji} is defined by:

$$C_{ji}\ \dot{q}_i + K_{ji}\ q_i = \phi_j^T F(\dot{q}_i) \tag{14.129}$$

and

$$F(q_i) = \int N^T p(\dot{q}_i)\, dA \tag{14.130}$$

where:

$F(q_i)$ = complex nodal damping force vector caused by a unit modal velocity of the source mode i.

$P(\dot{q}_i)$ = complex pressure due to unit modal velocity q_i

Note that the modal forces are complex numbers with a real and imaginary part. The real part, Re, represents the damping force and the imaginary part, Im, the squeeze force, which is cause by the fluid compression. The damping and squeeze coefficients are output by **DMPEXT** and are given by:

$$C_{ji} = \frac{\phi_j^T \int N^T \mathrm{Re}\{p(\dot{q}_i)\} dA}{\dot{q}_i} \tag{14.131}$$

and

$$K_{ji} = \frac{\phi_j^T \int N^T \mathrm{Im}\{p(\dot{q}_i)\} dA}{q_i} \tag{14.132}$$

Assuming the structure is excited by a unit modal velocity we obtain:

$$C_{ji} = \phi_j^T \int N^T \mathrm{Re}\{p(\phi_i)\} dA \tag{14.133}$$

and

$$K_{ji} = \Omega\ \phi_j^T \int N^T \mathrm{Im}\{p(\phi_i)\} dA \tag{14.134}$$

where:

Ω = excitation frequency (input on **DMPEXT** command)

Modal damping ratios ξ or the squeeze stiffness to structural stiffness ratio K_{Ratio} are defined only for the main diagonal elements and are also output with the **DMPEXT** command. These numbers are computed by:

$$\xi_i = \frac{C_{ii}}{2\ \omega_i\ m_i} = \text{modal damping ratio} \tag{14.135}$$

and

$$K_{Ratio} = \frac{K_{ii}}{\omega_i^2} = \text{squeeze stiffness to structural stiffness ratio} \tag{14.136}$$

where:

m_i = modal mass and the eigenfrequency ω_i

The damping ratio is necessary to compute α and β (input as **ALPHAD** and **BETAD** commands) parameters for Rayleigh damping models or to specify constant or modal damping (input by **DMPRAT** or **MDAMP** commands).

The squeeze to stiffness ratio specifies how much the structural stiffness is affected by the squeeze film. It can not directly be applied to structural elements but is helpful for user defined reduced order models.

14.11. Reduced Order Modeling of Coupled Domains

A direct finite element solution of coupled-physics problems is computationally very expensive. The goal of the reduced-order modeling is to generate a fast and accurate description of the coupled-physics systems to characterize their static or dynamic responses. The method presented here is based on a modal representation of coupled domains and can be viewed as an extension of the Mode Superposition Method (p. 698) to nonlinear structural and coupled-physics systems (Gabbay, et al.([230] (p. 933)), Mehner, et al.([250] (p. 935)), Mehner, et al.([335] (p. 939)), and Mehner, et al.([336] (p. 939))).

In the mode superposition method, the deformation state u of the structural domain is described by a factored sum of mode shapes:

$$u(x,y,z,t) = u_{eq} + \sum_{i=1}^{m} q_i(t)\ \phi_i(x,y,z) \tag{14.137}$$

where:

q_i = modal amplitude of mode i
φ_i = mode shape
u_{eq} = deformation in equilibrium state in the initial prestress position
m = number of considered modes

By substituting Equation 14.137 (p. 706) into the governing equations of motion, we obtain m constitutive equations that describe nonlinear structural systems in modal coordinates q_i:

$$m_i\ddot{q}_i + 2\ \xi_i\ \omega_i\ m_i\dot{q}_i + \frac{\partial\ W_{SENE}}{\partial\ q_i} = \sum_k f_i^N + \sum_l S_l\ f_i^S \tag{14.138}$$

where:

m_i = modal mass
ξ_i = modal damping factor
ω_i = angular frequency
W_{SENE} = strain energy
f_i^N = modal node force
f_i^E = modal element force
S_l = element load scale factor (input on **RMLVSCALE** command)

In a general case, Equation 14.138 (p. 706) are coupled since the strain energy W_{SENE} depends on the generalized coordinates q_i. For linear structural systems, Equation 14.138 (p. 706) reduces to Equation 14.118 (p. 700).

Reduced Order Modeling (ROM) substantially reduces running time since the dynamic behavior of most structures can be accurately represented by a few eigenmodes. The ROM method presented here is a three step procedure starting with a Generation Pass, followed by a Use Pass ROM144 - Reduced Order Electrostatic-Structural (p. 527), which can either be performed within ANSYS or externally in system simulator environment, and finally an optional Expansion Pass to extract the full DOF set solution according to Equation 14.137 (p. 706).

The entire algorithm can be outlined as follows:

- Determine the linear elastic modes from the modal analysis (**ANTYPE**,MODAL) of the structural problem.
- Select the most important modes based on their contribution to the test load displacement (**RMMSELECT** command).
- Displace the structure to various linear combinations of eigenmodes and compute energy functions for single physics domains at each deflection state (**RMSMPLE** command).
- Fit strain energy function to polynomial functions (**RMRGENERATE** command).
- Derive the ROM finite element equations from the polynomial representations of the energy functions.

14.11.1. Selection of Modal Basis Functions

Modes used for ROM can either be determined from the results of the test load application or based on their modal stiffness at the initial position.

Case 1: Test Load is Available (TMOD option on **RMMSELECT** command)

The test load drives the structure to a typical deformation state, which is representative for most load situations in the Use Pass. The mode contribution factors a_i are determined from

$$\begin{bmatrix} \phi_1^1 & \phi_1^2 & \cdots & \phi_1^m \\ \phi_2^1 & \phi_2^2 & \cdots & \phi_2^m \\ \phi_3^1 & \phi_3^2 & \cdots & \phi_3^m \\ \vdots & \vdots & \ddots & \vdots \\ \phi_n^1 & \phi_n^2 & \phi_n^3 & \phi_n^m \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{bmatrix} \tag{14.139}$$

where:

φ^i = mode shapes at the neutral plane nodes obtained from the results of the modal analysis (**RMNEVEC** command)

u^i = displacements at the neutral plane nodes obtained from the results of the test load (TLOAD option on **RMNDISP** command).

Mode contribution factors a_i are necessary to determine what modes are used and their amplitude range. Note that only those modes are considered in Equation 14.139 (p. 707), which actually act in the

operating direction (specified on the **RMANL** command). Criterion is that the maximum of the modal displacement in operating direction is at least 50% of the maximum displacement amplitude. The solution vector a_i indicates how much each mode contributes to the deflection state. A specified number of modes (*Nmode* of the **RMMSELECT** command) are considered unless the mode contribution factors are less than 0.1%.

Equation 14.139 (p. 707) solved by the least squares method and the results are scaled in such a way that the sum of all m mode contribution factors a_i is equal to one. Modes with highest a_i are suggested as basis functions.

Usually the first two modes are declared as dominant. The second mode is not dominant if either its eigenfrequency is higher than five times the frequency of the first mode, or its mode contribution factor is smaller than 10%.

The operating range of each mode is proportional to their mode contribution factors taking into account the total deflection range (*Dmax* and *Dmin* input on the **RMMSELECT** command). Modal amplitudes smaller than 2.5% of *Dmax* are increased automatically in order to prevent numerical round-off errors.

Case 2: Test Load is not Available (NMOD option on **RMMSELECT** command)

The first *Nmode* eigenmodes in the operating direction are chosen as basis functions. Likewise, a considered mode must have a modal displacement maximum in operating direction of 50% with respect to the modal amplitude.

The minimum and maximum operating range of each mode is determined by:

$$q_i = \frac{D_{Max/Min}}{\omega_i^2}\left(\sum_{j=1}^{m}\omega_j^{-2}\right)^{-1} \tag{14.140}$$

where:

$D_{Max/Min}$ = total deflection range of the structure (input on **RMMSELECT** command)

14.11.2. Element Loads

Up to 5 element loads such as acting gravity, external acceleration or a pressure difference may be specified in the Generation Pass and then scaled and superimposed in the Use Pass. In the same way as mode contribution factors a_i are determined for the test load, the mode contribution factors e_i^j for each element load case are determined by a least squares fit:

$$\begin{bmatrix} \phi_1^1 & \phi_1^2 & \cdots & \phi_1^k \\ \phi_2^1 & \phi_2^2 & \cdots & \phi_2^k \\ \phi_3^1 & \phi_3^2 & \cdots & \phi_3^k \\ \vdots & \vdots & \ddots & \vdots \\ \phi_n^1 & \phi_n^2 & \phi_n^3 & \phi_n^k \end{bmatrix} \begin{bmatrix} e_1^j \\ e_2^j \\ \vdots \\ e_n^j \end{bmatrix} = \begin{bmatrix} u_1^j \\ u_2^j \\ u_3^j \\ \vdots \\ u_n^j \end{bmatrix} \tag{14.141}$$

where:

u_i^j = displacements at the neutral plane nodes obtained from the results of the element load j (ELOAD option on **RMNDISP** command).

Here index k represents the number of modes, which have been selected for the ROM. The coefficients e_i^j are used to calculate modal element forces (see Element Matrices and Load Vectors (p. 527)).

14.11.3. Mode Combinations for Finite Element Data Acquisition and Energy Computation

In a general case, the energy functions depend on all basis functions. In the case of m modes and k data points in each mode direction one would need k^m sample points.

A large number of examples have shown that lower eigenmodes affect all modes strongly whereby interactions among higher eigenmodes are negligible. An explanation for this statement is that lower modes are characterized by large amplitudes, which substantially change the operating point of the system. On the other hand, the amplitudes of higher modes are reasonably small, and they do not influence the operating point.

Taking advantage of those properties is a core step in reducing the computational effort. After the mode selection procedure, the lowest modes are classified into dominant and relevant. For the dominant modes, the number of data points in the mode direction defaults to 11 and 5 respectively for the first and second dominant modes respectively. The default number of steps for relevant modes is 3. Larger (than the default above) number of steps can be specified on the **RMMRANGE** command.

A very important advantage of the ROM approach is that all finite element data can be extracted from a series of single domain runs. First, the structure is displaced to the linear combinations of eigenmodes by imposing displacement constrains to the neutral plane nodes. Then a static analysis is performed at each data point to determine the strain energy.

Both the sample point generation and the energy computation are controlled by the command **RMSMPLE**.

14.11.4. Function Fit Methods for Strain Energy

The objective of function fit is to represent the acquired FE data in a closed form so that the ROM FE element matrices (ROM144 - Reduced Order Electrostatic-Structural (p. 527)) are easily derived from the analytical representations of energy functions.

The ROM tool uses polynomials to fit the energy functions. Polynomials are very convenient since they can capture smooth functions with high accuracy, can be described by a few parameters and allow a simple computation of their local derivatives. Moreover, strain energy functions are inherent polynomials. In the case of linear systems, the strain energy can be exactly described by a polynomial of order two since the stiffness is constant. Stress-stiffened problems are captured by polynomials of order four.

The energy function fit procedure (**RMRGENERATE** command) calculates n_c coefficients that fit a polynomial to the n values of strain energy:

$$[A]\ \{K_{POLY}\} = \{W_{SENE}\} \tag{14.142}$$

where:

[A] = n x n_c matrix of polynomial terms
$\{K_{POLY}\}$ = vector of desired coefficients

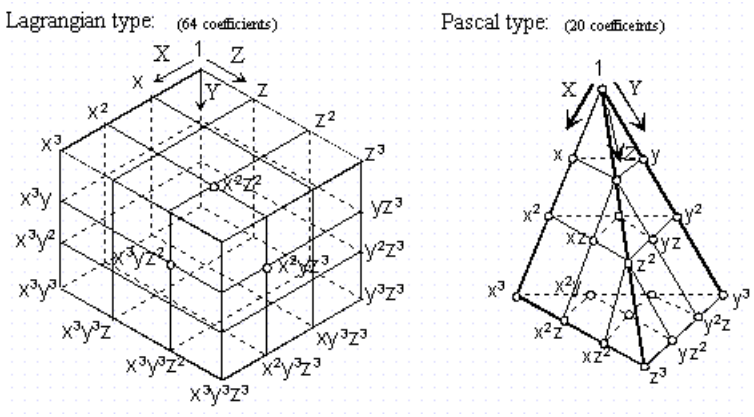
Note that the number of FE data (W_{SENE}) points n for a mode must be larger than the polynomial order P for the corresponding mode (input on **RMPORDER** command). Equation 14.142 (p. 709) is solved by means of a least squares method since the number of FE data points n is usually much larger than the number polynomial coefficients n_c.

The ROM tool uses four polynomial types (input on **RMROPTIONS** command):

Lagrange
Pascal
Reduced Lagrange
Reduced Pascal

Lagrange and Pascal coefficient terms that form matrix [A] in Equation 14.142 (p. 709) are shown in Figure 14.8: Set for Lagrange and Pascal Polynomials (p. 710).

Figure 14.8: Set for Lagrange and Pascal Polynomials



Polynomials for Order 3 for Three Modes (1-x, 2-y, 3-z)

Reduced Lagrange and Reduced Pascal polynomial types allow a further reduction of K_{POLY} by considering only coefficients located on the surface of the brick and pyramid respectively .

14.11.5. Coupled Electrostatic-Structural Systems

The ROM method is applicable to electrostatic-structural systems.

The constitutive equations for a coupled electrostatic-structural system in modal coordinates are:

$$m_i\ddot{q}_i + 2\ \xi_i\ \omega_i\ m_i\dot{q}_i + \frac{\partial\ W_{SENE}}{\partial\ q_i} = \sum_k f_i^N + \sum_l S_l\ f_i^E - \frac{\partial\ W_{el}}{\partial\ q_i} \tag{14.143}$$

for the modal amplitudes and

$$I_i = \dot{Q}_i = \frac{\partial\ W_{el}}{\partial\ V_i} \tag{14.144}$$

where:

I_i = current in conductor i
Q_i = charge on the ith conductor
V_i = ith conductor voltage

The electrostatic co-energy is given by:

$$W_{el} = \sum_r \frac{C_{ij}^r}{2}(V_i - V_j)^2 \tag{14.145}$$

where:

C_{ij} = lumped capacitance between conductors i and j (input on **RMCAP** command)
r = index of considered capacitance

14.11.6. Computation of Capacitance Data and Function Fit

The capacitances C_{ij}, and the electrostatic co-energy respectively, are functions of the modal coordinates q_i. As the strain energy W_{SENE} for the structural domain, the lumped capacitances are calculated for each k data points in each mode direction, and then fitted to polynomials. Following each structural analysis to determine the strain energy W_{SENE}, (n-1) linear simulations are performed in the deformed electrostatic domain, where n is the number of conductors, to calculate the lumped capacitances. The capacitance data fit is similar to the strain energy fit described above (Function Fit Methods for Strain Energy (p. 709)). It is sometimes necessary to fit the inverted capacitance function (using the *Invert* option on the **RMROPTIONS** command).

14.12. Newton-Raphson Procedure

The following Newton-Raphson procedure topics are available:
14.12.1. Overview
14.12.2. Convergence
14.12.3. Predictor
14.12.4. Adaptive Descent
14.12.5. Line Search
14.12.6. Arc-Length Method

14.12.1. Overview

The finite element discretization process yields a set of simultaneous equations:

$$[K]\{u\} = \{F^a\} \tag{14.146}$$

where:

[K] = coefficient matrix
{u} = vector of unknown DOF (degree of freedom) values
$\{F^a\}$ = vector of applied loads

If the coefficient matrix [K] is itself a function of the unknown DOF values (or their derivatives) then Equation 14.146 (p. 711) is a nonlinear equation. The Newton-Raphson method is an iterative process of solving the nonlinear equations and can be written as (Bathe([2] (p. 921))):

$$[K_i^T]\{\Delta u_i\} = \{F^a\} - \{F_i^{nr}\} \tag{14.147}$$

$$\{u_{i+1}\} = \{u_i\} + \{\Delta u_i\} \tag{14.148}$$

where:

$[K_i^T]$ = Jacobian matrix (tangent matrix)
i = subscript representing the current equilibrium iteration
$\{F_i^{nr}\}$ = vector of restoring loads corresponding to the element internal loads

Both $[K_i^T]$ and $\{F_i^{nr}\}$ are evaluated based on the values given by $\{u_i\}$. The right-hand side of Equation 14.147 (p. 711) is the residual or out-of-balance load vector; i.e., the amount the system is out of equilibrium. A single solution iteration is depicted graphically in Figure 14.9: Newton-Raphson Solution - One Iteration (p. 713) for a one DOF model. In a structural analysis, $[K_i^T]$ is the tangent stiffness matrix, $\{u_i\}$ is the displacement vector and $\{F_i^{nr}\}$ is the restoring force vector calculated from the element stresses. In a thermal analysis, $[K_i^T]$ is the conductivity matrix, $\{u_i\}$ is the temperature vector and $\{F_i^{nr}\}$ is the resisting load vector calculated from the element heat flows. In an electromagnetic analysis, $[K_i^T]$ is the Dirichlet matrix, $\{u_i\}$ is the magnetic potential vector, and $\{F_i^{nr}\}$ is the resisting load vector calculated from element magnetic fluxes. In a transient analysis, $[K_i^T]$ is the effective coefficient matrix and $\{F_i^{nr}\}$ is the effective applied load vector which includes the inertia and damping effects.

As seen in the following figures, more than one Newton-Raphson iteration is needed to obtain a converged solution. The general algorithm proceeds as follows:

1. Assume $\{u_0\}$. $\{u_0\}$ is usually the converged solution from the previous time step. On the first time step, $\{u_0\} = \{0\}$.

2. Compute the updated tangent matrix $[K_i^T]$ and the restoring load $\{F_i^{nr}\}$ from configuration $\{u_i\}$.

3. Calculate $\{\Delta u_i\}$ from Equation 14.147 (p. 711).

4. Add $\{\Delta u_i\}$ to $\{u_i\}$ in order to obtain the next approximation $\{u_{i+1}\}$ (Equation 14.148 (p. 712)).

5. Repeat steps 2 to 4 until convergence is obtained.

Figure 14.9: Newton-Raphson Solution - One Iteration

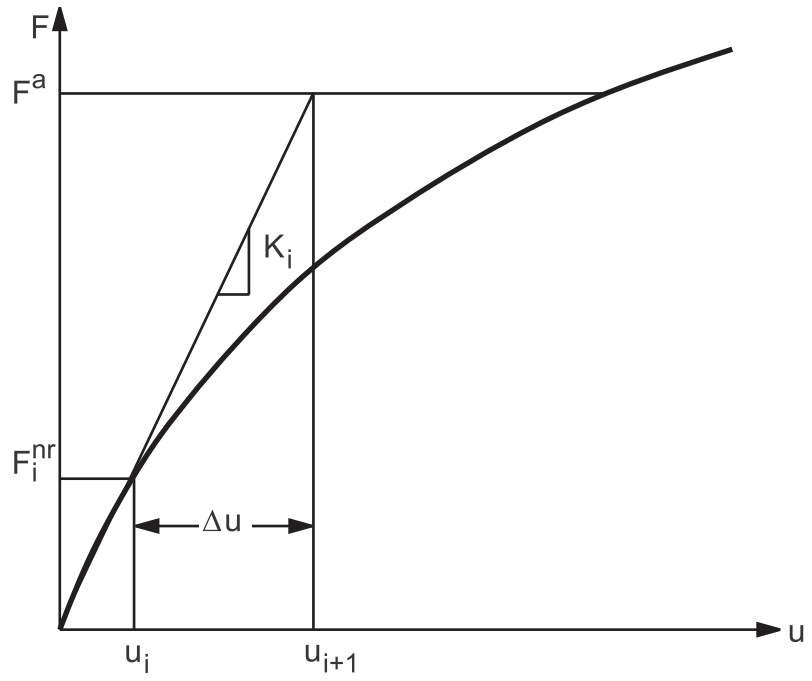
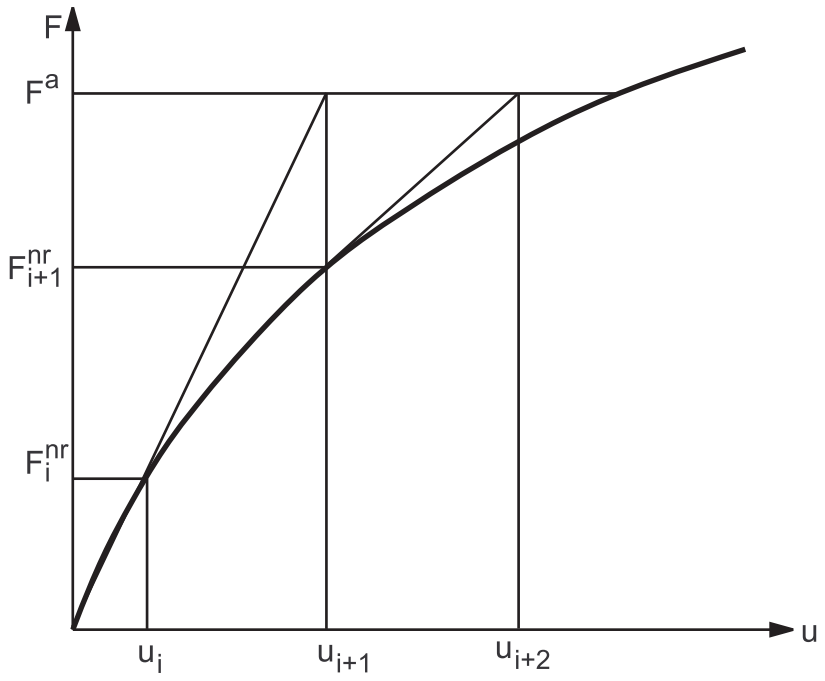


Figure 14.10: Newton-Raphson Solution - Next Iteration (p. 714) shows the solution of the next iteration (i + 1) of the example from Figure 14.9: Newton-Raphson Solution - One Iteration (p. 713). The subsequent iterations would proceed in a similar manner.

The solution obtained at the end of the iteration process would correspond to load level $\{F^a\}$. The final converged solution would be in equilibrium, such that the restoring load vector $\{F_i^{nr}\}$ (computed from the current stress state, heat flows, etc.) would equal the applied load vector $\{F^a\}$ (or at least to within some tolerance). None of the intermediate solutions would be in equilibrium.

Figure 14.10: Newton-Raphson Solution - Next Iteration



If the analysis included path-dependent nonlinearities (such as plasticity), then the solution process requires that some intermediate steps be in equilibrium in order to correctly follow the load path. This is accomplished effectively by specifying a step-by-step incremental analysis; i.e., the final load vector $\{F^a\}$ is reached by applying the load in increments and performing the Newton-Raphson iterations at each step:

$$[K^T_{n,i}]\{\Delta u_i\} = \{F^a_n\} - \{F^{nr}_{n,i}\} \tag{14.149}$$

where:

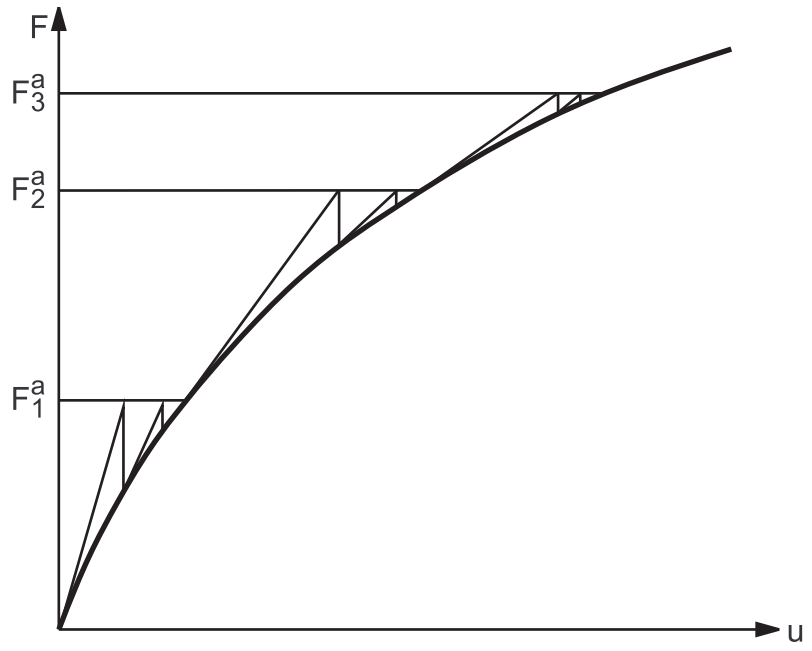
$[K_{n,i}]$ = tangent matrix for time step n, iteration i

$\{F^a_n\}$ = total applied force vector at time step n

$\{F^{nr}_{n,i}\}$ = restoring force vector for time step n, iteration i

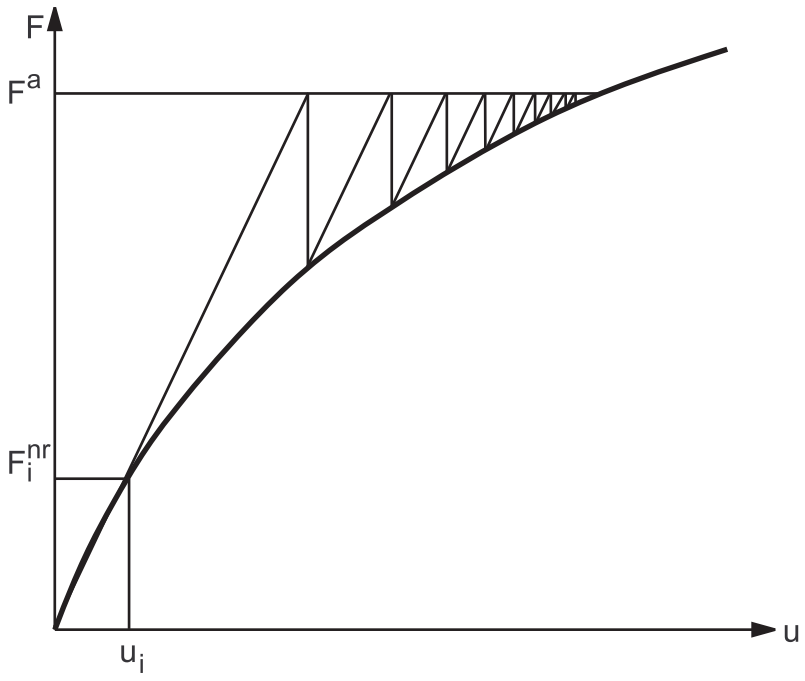
This process is the incremental Newton-Raphson procedure and is shown in Figure 14.11: Incremental Newton-Raphson Procedure (p. 715). The Newton-Raphson procedure guarantees convergence if and only if the solution at any iteration $\{u_i\}$ is "near" the exact solution. Therefore, even without a path-dependent nonlinearity, the incremental approach (i.e., applying the loads in increments) is sometimes required in order to obtain a solution corresponding to the final load level.

Figure 14.11: Incremental Newton-Raphson Procedure



When the stiffness matrix is updated every iteration (as indicated in Equation 14.147 (p. 711) and Equation 14.149 (p. 714)) the process is termed a full Newton-Raphson solution procedure (**NROPT**,FULL or **NROPT**,UNSYM). Alternatively, the stiffness matrix could be updated less frequently using the modified Newton-Raphson procedure (**NROPT**,MODI). Specifically, for static or transient analyses, it would be updated only during the first or second iteration of each substep, respectively. Use of the initial-stiffness procedure (**NROPT**,INIT) prevents any updating of the stiffness matrix, as shown in Figure 14.12: Initial-Stiffness Newton-Raphson (p. 716). If a multistatus element is in the model, however, it would be updated at iteration in which it changes status, irrespective of the Newton-Raphson option. The modified and initial-stiffness Newton-Raphson procedures converge more slowly than the full Newton-Raphson procedure, but they require fewer matrix reformulations and inversions. A few elements form an approximate tangent matrix so that the convergence characteristics are somewhat different.

Figure 14.12: Initial-Stiffness Newton-Raphson



14.12.2. Convergence

The iteration process described in the previous section continues until convergence is achieved. The maximum number of allowed equilibrium iterations (input on **NEQIT** command) are performed in order to obtain convergence.

Convergence is assumed when

$$\|\{R\}\| < \varepsilon_R R_{ref} \qquad \text{(out-of-balance convergence)} \tag{14.150}$$

and/or

$$\|\{\Delta u_i\}\| < \varepsilon_u u_{ref} \qquad \text{(DOF increment convergence)} \tag{14.151}$$

where {R} is the residual vector:

$$\{R\} = \{F^a\} - \{F^{nr}\} \tag{14.152}$$

which is the right-hand side of the Newton-Raphson Equation 14.147 (p. 711). $\{\Delta u_i\}$ is the DOF increment vector, ε_R and ε_u are tolerances (**CNVTOL**,,,*TOLER*) and R_{ref} and u_{ref} are reference values (**CNVTOL**,,*VALUE*). $\|\bullet\|$ is a vector norm; that is, a scalar measure of the magnitude of the vector (defined below).

Convergence, therefore, is obtained when size of the residual (disequilibrium) is less than a tolerance times a reference value and/or when the size of the DOF increment is less than a tolerance times a reference value. The default is to use out-of-balance convergence checking only. The default tolerance are .001 (for both ε_u and ε_R).

There are three available norms (**CNVTOL**,,,,*NORM*) to choose from:

1. Infinite norm $\|\{R\}\|_\infty = \max|R_i|$

2. L1 norm $\|\{R\}\|_1 = \sum|R_i|$

3. L2 norm $\|\{R\}\|_2 = (\sum R_i^2)^{\frac{1}{2}}$

For DOF increment convergence, substitute Δu for R in the above equations. The infinite norm is simply the maximum value in the vector (maximum residual or maximum DOF increment), the L1 norm is the sum of the absolute value of the terms, and the L2 norm is the square root of the sum of the squares (SRSS) value of the terms, also called the Euclidean norm. The default is to use the L2 norm.

The default out-of-balance reference value R_{ref} is $||\{F^a\}||$. For DOFs with imposed displacement constraints, the default R_{ref} value is $\{F^{nr}\}$ at those DOFs.

In the following cases, the default R_{ref} value is the specified or default minimum reference value set via the **CNVTOL**,,,,,,*MINREF* command:

- For structural DOFs if R_{ref} falls below 1.0E-2 (typically occurring in rigid-body motion analyses, such as those involving stress-free rotation)
- For thermal DOFs if R_{ref} falls below 1.0E-6
- For EMAG DOFs if R_{ref} falls below 1.0E-12
- For all other DOFs if R_{ref} is equal to 0.0

The default reference value u_{ref} is $||\{u\}||$.

14.12.3. Predictor

The solution used for the start of each time step n $\{u_{n,0}\}$ is usually equal to the current DOF solution $\{u_{n-1}\}$. The tangent matrix $[K_{n,0}]$ and restoring load $\{F^{n,0}\}$ are based on this configuration. The predictor option (**PRED** command) extrapolates the DOF solution using the previous history in order to take a better guess at the next solution.

In static analyses, the prediction is based on the displacement increments accumulated over the previous time step, factored by the time-step size:

$$\{u_{n,0}\} = \{u_{n-1}\} + \beta\{\Delta u_n\} \tag{14.153}$$

where:

$\{\Delta u_n\}$ = displacement increment accumulated over the previous time step
n = current time step

$$\{\Delta u_n\} = \sum_{i=1}^{NEQIT} \{\Delta u_i\} \tag{14.154}$$

and β is defined as:

$$\beta = \frac{\Delta t_n}{\Delta t_{n-1}} \tag{14.155}$$

where:

Δt_n = current time-step size
$\Delta t_{n-}1$ = previous time-step size

β is not allowed to be greater than 5.

In transient analyses, the prediction is based on the current velocities and accelerations using the Newmark formulas for structural DOFs:

$$\{u_{n,0}\} = \{u_{n-1}\} + \{\dot{u}_{n-1}\}\Delta t_n + (\frac{1}{2} - \alpha)\{\ddot{u}_{n-1}\}\Delta t_n^2 \tag{14.156}$$

where:

$\{u_{n-1}\}, \{\dot{u}_{n-1}\}, \{\ddot{u}_{n-1}\}$ = current displacements, velocities and accelerations
Δt_n = current time-step size
α = Newmark parameter (input on **TINTP** command)

For thermal, magnetic and other first order systems, the prediction is based on the trapezoidal formula:

$$\{u_{n,0}\} = \{u_{n-1}\} + (1-\theta)\{\dot{u}_{n-1}\}\Delta t_n \tag{14.157}$$

where:

$\{u_{n-1}\}$ = current temperatures (or magnetic potentials)
$\{\dot{u}_{n-1}\}$ = current rates of these quantities
θ = trapezoidal time integration parameter (input on **TINTP** command)

See Transient Analysis (p. 763) for more details on the transient procedures.

The subsequent equilibrium iterations provide DOF increments $\{\Delta u\}$ with respect to the predicted DOF value $\{u_{n,0}\}$, hence this is a predictor-corrector algorithm.

14.12.4. Adaptive Descent

Adaptive descent (*Adptky* on the **NROPT** command) is a technique which switches to a "stiffer" matrix if convergence difficulties are encountered, and switches back to the full tangent as the solution convergences, resulting in the desired rapid convergence rate (Eggert([152] (p. 929))).

The matrix used in the Newton-Raphson equation (Equation 14.147 (p. 711)) is defined as the sum of two matrices:

$$[K_i^T] = \xi[K^S] + (1-\xi)[K^T] \tag{14.158}$$

where:

$[K^S]$ = secant (or most stable) matrix
$[K^T]$ = tangent matrix

ξ = descent parameter

The program adaptively adjusts the descent parameter (ξ) during the equilibrium iterations as follows:

1. Start each substep using the tangent matrix ($\xi = 0$).
2. Monitor the change in the residual $||\{R\}||_2$ over the equilibrium iterations:

   If it increases (indicating possible divergence):

   - remove the current solution if $\xi < 1$, reset ξ to 1 and redo the iteration using the secant matrix
   - if already at $\xi = 1$, continue iterating

   If it decreases (indicating converging solution):

   - If $\xi = 1$ (secant matrix) and the residual has decreased for three iterations in a row (or 2 if ξ was increased to 1 during the equilibrium iteration process by (a.) above), then reduce ξ by a factor of 1/4 (set it to 0.25) and continue iterating.
   - If the $\xi < 1$, decrease it again by a factor of 1/4 and continue iterating. Once ξ is below 0.0156, set it to 0.0 (use the tangent matrix).
3. If a negative pivot message is encountered (indicating an ill-conditioned matrix):
   - If $\xi < 1$, remove the current solution, reset $\xi = 1$ and redo the iteration using the secant matrix.
   - If $\xi = 1$, bisect the time step if automatic time stepping is active, otherwise terminate the execution.

The nonlinearities which make use of adaptive descent (that is, they form a secant matrix if $\xi > 0$) include: plasticity, contact, stress stiffness with large strain, nonlinear magnetics using the scalar potential formulation, the concrete element SOLID65 with KEYOPT(7) = 1, and the membrane shell element SHELL41 with KEYOPT(1) = 2. Adaptive descent is used by default in these cases unless the line search or arc-length options are on. It is only available with full Newton-Raphson, where the matrix is updated every iteration. Full Newton-Raphson is also the default for plasticity, contact and large strain nonlinearities.

14.12.5. Line Search

The line search option (accessed with **LNSRCH** command) attempts to improve a Newton-Raphson solution $\{\Delta u_i\}$ by scaling the solution vector by a scalar value termed the line search parameter.

Consider Equation 14.148 (p. 712) again:

$$\{u_{i+1}\} = \{u_i\} + \{\Delta u_i\} \tag{14.159}$$

In some solution situations, the use of the full $\{\Delta u_i\}$ leads to solution instabilities. Hence, if the line search option is used, Equation 14.159 (p. 719) is modified to be:

$$\{u_{i+1}\} = \{u_i\} + s\{\Delta u_i\} \tag{14.160}$$

where:

s = line search parameter, $0.05 < s < 1.0$

s is automatically determined by minimizing the energy of the system, which reduces to finding the zero of the nonlinear equation:

$$g_s = \{\Delta u_i\}^T (\{F^a\} - \{F^{nr}(s\{\Delta u_i\})\}) \quad (14.161)$$

where:

g_s = gradient of the potential energy with respect to s

An iterative solution scheme based on regula falsi is used to solve Equation 14.161 (p. 720) (Schweizerhof and Wriggers([153] (p. 929))). Iterations are continued until either:

1. g_s is less than 0.5 g_o, where g_o is the value of Equation 14.161 (p. 720) at s = 0.0 (that is, using $\{F^{nr}_{n-1}\}$ for $\{F^{nr}(s\{\Delta u\})\}$).
2. g_s is not changing significantly between iterations.
3. Six iterations have been performed.

If $g_o > 0.0$, no iterations are performed and s is set to 1.0. s is not allowed below 0.05.

The scaled solution $\{\Delta u_i\}$ is used to update the current DOF values $\{u_{i+1}\}$ in Equation 14.148 (p. 712) and the next equilibrium iteration is performed.

14.12.6. Arc-Length Method

The arc-length method (accessed with **ARCLEN**,ON) is suitable for nonlinear static equilibrium solutions of unstable problems.

Application of the arc-length method involves the tracing of a complex path in the load-displacement response into the buckling/post buckling regimes. The arc-length method uses Crisfield's method as describe in ([174] (p. 930)) to prevent any fluctuation of the step size during equilibrium iterations. It is assumed that all load magnitudes can be controlled by a single scalar parameter (that is, the total load factor).

An unsmooth or discontinuous load-displacement response, which is sometimes seen in contact analyses and elastic-perfectly plastic analyses, cannot be traced effectively by the arc-length solution method.

Mathematically, the arc-length method can be viewed as the trace of a single equilibrium curve in a space spanned by the nodal displacement variables and the total load factor. Therefore, all options of the Newton-Raphson method are still the basic method for the arc-length solution. As the displacement vectors and the scalar load factor are treated as unknowns, the arc-length method itself is an automatic load step method; therefore, **AUTOTS**,ON is not needed.

For problems with sharp turns in the load-displacement curve or with path-dependent materials, it is necessary to limit the initial arc-length radius (**NSUBST**) and the arc-length radius augmentation (via the *MAXARC* argument of the **ARCLEN** command). During the solution, the arc-length method will vary the arc-length radius at each arc-length substep according to the degree of nonlinearity that is involved. The range of variation of the arc-length radius is limited by the maximum and minimum multipliers (*MAXARC* and *MINARC* on the **ARCLEN** command).

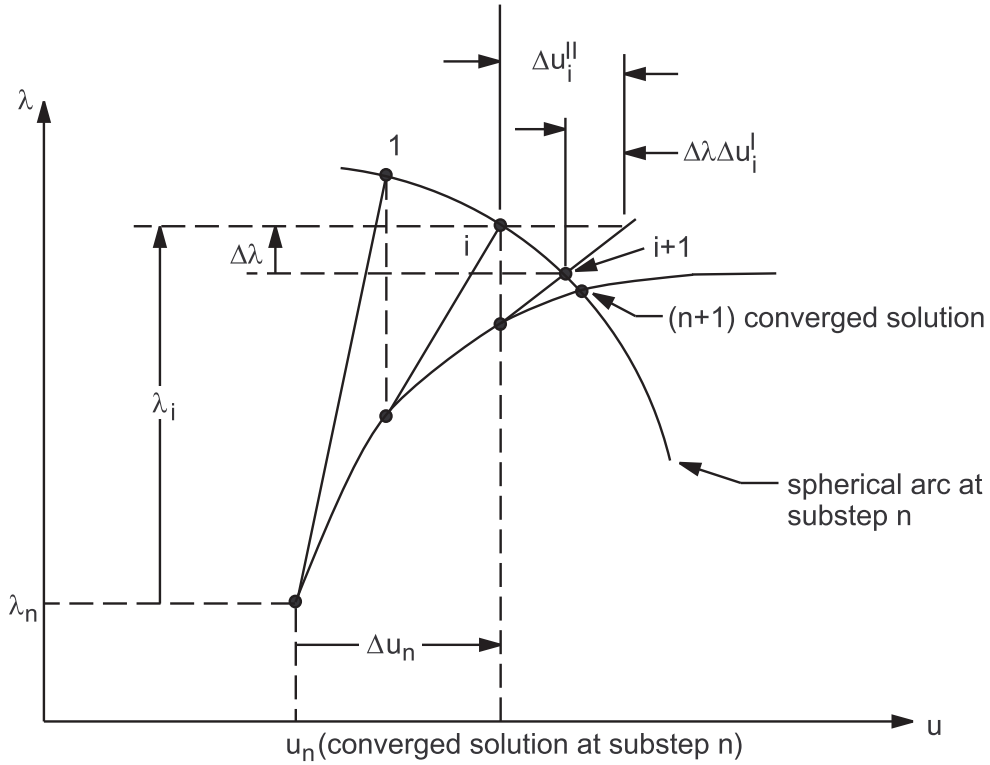
In the arc-length procedure, Equation 14.147 (p. 711) is recast and associated with the total load factor λ:

$$[K_i^T]\{\Delta u_i\} = \lambda\{F^a\} - \{F_i^{nr}\} \quad (14.162)$$

Figure 14.13: Arc-Length Approach with Full Newton-Raphson Method (p. 721) is a graphical representation of the arc-length method. Writing the proportional loading factor λ in an incremental form at substep n and iteration i yields:

$$[K_i^T]\{\Delta u_i\} - \Delta\lambda\{F^a\} = (\lambda_n + \lambda_i)\{F^a\} - \{F_i^{nr}\} = -\{R_i\} \tag{14.163}$$

Figure 14.13: Arc-Length Approach with Full Newton-Raphson Method



Following Equation 14.163 (p. 721), the incremental displacement $\{\Delta u_i\}$ can be written in two parts:

$$\{\Delta u_i\} = \Delta\lambda\{\Delta u_i^I\} + \{\Delta u_i^{II}\} \tag{14.164}$$

where:

$$\{\Delta u_i^I\} = [K_i^T]^{-1}\{F^a\} \tag{14.165}$$

$$\{\Delta u_i^{II}\} = -[K_i^T]^{-1}\{R_i\} \tag{14.166}$$

In each arc-length iteration, it is necessary to use Equation 14.165 (p. 721) and Equation 14.166 (p. 721) to solve for $\{\Delta u_i^I\}$ and $\{\Delta u_i^{II}\}$.

We can define one vector between the previous equilibrium point and a point determined in iteration i by:

$$t_i = \{\Delta u_n\} + \beta\lambda_i \tag{14.167}$$

where:

$\{\Delta u_n\}$ = the displacement increment accumulated over the current time step

β = the scaling vector for unit displacements

The norm of this vector is:

$$|t_i| = \sqrt{\{\Delta u_n\}^t \{\Delta u_n\} + \beta^2 \lambda_i^2} \tag{14.168}$$

At iteration i+1, the vector becomes:

$$t_{i+1} = \{\Delta u_n\} + \{\Delta u_i\} + \beta(\lambda_i + \Delta\lambda) \tag{14.169}$$

Crisfield's method assumes that the norm of the vector is constant along the equilibrium iterations, that is:

$$|t_i| = |t_{i+1}| \tag{14.170}$$

Substituting Equation 14.167 (p. 721) and Equation 14.169 (p. 722) and using Equation 14.164 (p. 721), the following quadratic equation results:

$$a(\Delta\lambda)^2 + b(\Delta\lambda) + c = 0 \tag{14.171}$$

where:

$$a = \beta^2 + \{\Delta u_i^I\}^t \{\Delta u_i^I\}$$

$$b = 2\left(\beta^2 \lambda_i + \{\Delta u_n\}^t \{\Delta u_i^I\} + \{\Delta u_i^{II}\}^t \{\Delta u_i^I\}\right)$$

$$c = 2\{\Delta u_n\}^t \{\Delta u_i^{II}\} + \{\Delta u_i^{II}\}^t \{\Delta u_i^{II}\}$$

This system has two real roots, $\Delta\lambda^{(1)}$ and $\Delta\lambda^{(2)}$, which both satisfy the constant arc-length radius.

For each of these roots, we can define the angle between vector t_{n-1} in the previously converged substep, and vector t_{i+1} in the current substep. This angle may be obtained from:

$$\cos(\theta) = \frac{(t_{i+1})^T \cdot (t_{n-1})}{|t_n| \cdot |t_{n-1}|} \tag{14.172}$$

To move the equilibrium path forward, we choose the route for which the cosine of the associated angle is the closest to 1.

Finally, the solution vectors are updated according to (see Figure 14.13: Arc-Length Approach with Full Newton-Raphson Method (p. 721)):

$$\{u_{i+1}\} = \{u_n\} + \{\Delta u_n\} + \{\Delta u_i\} \tag{14.173}$$

and

$$\lambda_{i+1} = \lambda_n + \lambda_i + \Delta\lambda \tag{14.174}$$

Values of λ_n and $\Delta\lambda$ are available in POST26 corresponding to labels ALLF and ALDLF, respectively, on the **SOLU** command. The normalized arc-length radius label ARCL (also on **SOLU**) corresponds to value

$\left|\frac{t_n}{t_0}\right|$, where $|t_0|$ is the initial arc-length radius defined through Equation 14.168 (p. 722) and Equation 14.175 (p. 723) (an arc-length radius at the first iteration of the first substep).

Defining K_0 as the initial tangent matrix, F^a as the full external load, and λ_0 as the initial load factor (specified using the **NSUBST** command), the initial arc-length radius t_0 is determined by:

$$t_0 = \sqrt{\{u_0\}^T \{u_0\} + \beta^2 \lambda_0^2} \tag{14.175}$$

where:

$$u_0 = \left[K_0^T\right]^{-1} \left\{\lambda_0 F^a\right\}$$

$$\beta = \frac{\sqrt{\{u_0\}^T \{u_0\}}}{\lambda_0}$$

The factors *MAXARC* and *MINARC* (input on the **ARCLEN** command) are used to define the limits of the arc-length radius by the following formulas:

t_{MAX} = *MAXARC* * t_0

t_{MIN} = *MINARC* * t_0

14.13. Constraint Equations

14.13.1. Derivation of Matrix and Load Vector Operations

Given the set of L linear simultaneous equations in unknowns u_j subject to the linear constraint equation (input on **CE** command)

$$\sum_{j=1}^{L} K_{kj} u_j = F_k \qquad (1 \le k \le L) \tag{14.176}$$

where:

K_{kj} = stiffness term relating the force at degrees of freedom k to the displacement at degrees of freedom j
u_j = nodal displacement of degrees of freedom j
F_k = nodal force of degrees of freedom k
k = equation (row) number
j = column number
L = number of equations

$$\sum_{j=1}^{L} C_j u_j = C_o \tag{14.177}$$

normalize Equation 14.177 (p. 723) with respect to the prime degrees of freedom u_i by dividing by C_i to get:

$$\sum_{j=1}^{L} C_j^* u_j = C_o^* \tag{14.178}$$

where:

$$C_j^* = C_j / C_i$$

$$C_o^* = C_o / C_i$$

which is written to a file for backsubstitution. Equation 14.178 (p. 724) is expanded (recall $C_i^* = 1$) as:

$$u_i + \sum_{j=1}^{L} C_j^* u_j = C_o^* \qquad (j \neq i) \tag{14.179}$$

Equation 14.176 (p. 723) may be similarly expanded as:

$$K_{ki} u_i + \sum_{j=1}^{L} K_{kj} u_j = F_k \qquad (j \neq i) \tag{14.180}$$

Multiply Equation 14.179 (p. 724) by K_{ki} and subtract from Equation 14.180 (p. 724) to get:

$$\sum_{j=1}^{L} (K_{kj} - C_j^* K_{ki}) u_j = F_k - C_o^* K_{ki} \, (j \neq i) \tag{14.181}$$

Specializing Equation 14.181 (p. 724) for k = i allows it to be written as:

$$\sum_{j=1}^{L} (K_{ij} - C_j^* K_{ii}) u_j = F_i - C_o^* K_{ii} \qquad (j \neq i) \tag{14.182}$$

This may be considered to be a revised form of the constraint equation. Introducing a Lagrange multiplier λ_k, Equation 14.181 (p. 724) and Equation 14.182 (p. 724) may be combined as:

$$\begin{aligned} &\sum_{j=1}^{L} (K_{kj} - C_j^* K_{ki}) u_j - F_k + C_o^* K_{ki} \\ &+\lambda_k \left[\sum_{j=1}^{L} (K_{ij} - C_j^* K_{ii}) u_j - F_i + C_o^* K_{ii} \right] = 0 \, (j \neq i) \end{aligned} \tag{14.183}$$

By the standard Lagrange multiplier procedure (see Denn([8] (p. 921))):

$$\lambda_k = \frac{\partial u_i}{\partial u_k} \tag{14.184}$$

Solving Equation 14.179 (p. 724) for u_i,

$$u_i = C_o^* - \sum_{j=1}^{L} C_j^* u_j \qquad (j \neq i) \tag{14.185}$$

so that

$$\lambda_k = -C_k^* \tag{14.186}$$

Substituting Equation 14.186 (p. 724) into Equation 14.183 (p. 724) and rearranging terms,

$$\sum_{j=1}^{L}(K_{kj} - C_j^* K_{ki} - C_k^* K_{ij} + C_k^* C_j^* K_{ii})u_j = F_k - C_o^* K_{ki} - C_k^* F_i + C_k^* C_o^* K_{ii} \ (j \neq i) \tag{14.187}$$

or

$$\sum_{j=1}^{L-1} K_{kj}^* u_j = F_k^* \qquad (1 \le k \le L-1) \tag{14.188}$$

where:

$$K_{kj}^* = K_{kj} - C_j^* K_{ki} - C_k^* K_{ij} + C_k^* C_j^* K_{ii}$$

$$F_k^* = F_k - C_o^* K_{ki} - C_k^* F_i + C_k^* C_o^* K_{ii}$$

14.13.2. Constraints: Automatic Selection of Slave DOFs

The constraint equation described by Equation 14.177 (p. 723) can also be written in the following matrix form:

$$[C]\{U\} = \{C_0\} \tag{14.189}$$

where [C] can be further composed into the slave and master DOFs so that the direct elimination method can be used. (In the direct elimination method, Equation 14.187 (p. 725) is used to solve equation systems Equation 14.176 (p. 723) and Equation 14.177 (p. 723) together.)

Equation 14.189 (p. 725) can be re-written as:

$$[C_s]\{U_s\} + [C_m]\{U_m\} = \{C_0\} \tag{14.190}$$

where $\{U_s\}$ is a displacement of the slave DOFs, and $\{U_m\}$ is the master DOF.

If external **CE** or **CP** commands are issued, the user must choose which DOFs are slave and which are master (see the **CE**/**CP** command descriptions). In many applications, the ANSYS program automatically uses the **CE** command and invokes the method of automatic selection.

When solving the equation with the direct elimination method, the $\{U_s\}$ variables can be removed from the system by applying the following transformation:

$$\{U_s\} = [C_s]^{-1} * [\{C_0\} - [C_m]\{U_m\}] \tag{14.191}$$

Because the choice of $\{U_s\}$ is not unique, the program selects $\{U_s\}$ slave DOFs that ensure that Equation 14.177 (p. 723) has the best possible matrix conditioning (to avoid an ill-conditioned matrix) and fewer fill-ins when the sparse direct solver is used.

If a value for $\{U_s\}$ cannot be selected that makes Equation 14.176 (p. 723) and Equation 14.177 (p. 723) solvable together, the equation system is overconstrained. Typically, overconstraint is caused by contradictory constraint equations or an insufficient number of slave DOFs.

The ANSYS program has two automatic methods (which can be selected via the **OVCHECK** command) to compute the slave DOFs and detect overconstrained DOFs:

The Topological Method (OVCHECK,TOPO)

This method is used by default. This simple method requires little computational time to check for overconstraints and determine unique slave DOFs. This method starts by traversing the topology (sparsity) pattern of the [C] matrix (Equation 14.189 (p. 725)) row-by-row. During this process, the first possible DOFs visited are considered slave DOFs. If the slave DOF has been used (or appeared) before, then it selects the next possible DOF visited as a slave DOF. Although this method traverses the entire matrix, it is computationally inexpensive because no numerical computations are involved.

This method does not guarantee that overconstraint or unique slave DOFs can be determined for the entire equation system. If this method fails to detect slave DOFs, warning or error messages display in the program output files.

This method only traverses the constraint equation system and does not visit matrix coefficients associated with the P variable (i.e., Lagrangian multipliers). As a result, the topological method cannot detect overconstraints associated with u-P formulation elements.

The Algebraic Method (OVCHECK,ALGEBRAIC)

This is a mathematical method that requires more computational time than the topological method.

The slave DOFs are chosen using factorization with full pivoting applied to the [C] matrix. If one pivot is too small (for example, lower than 1.e-08 proportionally), the associated constraint equation is considered redundant and removed from the set of constraint equations that must be satisfied.

If redundant constraint equations are detected, the program prints information regarding their removal.

In cases where a model uses the **CE**/**CP** command and u-P formulation elements, overconstraint could also come from the P DOFs, in addition to **CE**/**CP** overconstraint. Detection of overconstraint in this case is similar to the process described above; however, only the algebraic method supports this detection. The algebraic method packs **CE**/**CP** and matrix coefficients from the Lagrange multiplier equations into one assembled [C] matrix (Equation 14.189 (p. 725)), then it executes the algebraic method process described above into the [C] matrix.

If **CE**/**CP** values are invalid or redundant, they are removed from equation (Equation 14.176 (p. 723)) automatically, and the invalid or redundant **CE**/**CP** will not be used in the entire analysis process, which includes the Newton-Raphson nonlinear convergence loop.

If some P constraint equations are invalid or redundant, the application stops and the user is notified which elements (e.g., MPC184, CONTA174, etc.) are causing overconstraint (i.e., the redundant equations).

14.14. Eigenvalue and Eigenvector Extraction

The following extraction methods and related topics are available:

The eigenvalue and eigenvector problem needs to be solved for mode-frequency and buckling analyses. It has the form of:

$$[K]\{\phi_i\} = \lambda_i[M]\{\phi_i\} \tag{14.192}$$

where:

[K] = structural stiffness matrix
$\{\phi_i\}$ = eigenvector
λ_i = eigenvalue
[M] = structural mass matrix

For prestressed modal analyses, the [K] matrix includes the stress stiffness matrix [S]. For eigenvalue buckling analyses, the [M] matrix is replaced with the stress stiffness matrix [S]. The discussions given in the rest of this section assume a modal analysis (**ANTYPE**,MODAL) except as noted, but also generally applies to eigenvalue buckling analyses.

The eigenvalue and eigenvector extraction procedures available include the Block Lanczos, PCG Lanczos, Supernode, Subspace, unsymmetric, damped, and QR damped methods (**MODOPT** and **BUCOPT** commands) outlined in Table 14.1: Procedures Used for Eigenvalue and Eigenvector Extraction (p. 727). Each method is discussed subsequently. Shifting, applicable to most methods, is discussed in a later section (see Shifting (p. 734)).

Table 14.1: Procedures Used for Eigenvalue and Eigenvector Extraction

Procedure	Input	Usages	Applicable Matrices++	Reduction	Extraction Technique
Supernode	**MODOPT**, SNODE	Symmetric (but not applicable to buckling)	K, M	None	Internally uses node grouping, reduced, and Lanczos methods
Block Lanczos	**MODOPT**, LANB	Symmetric	K, M	None	Lanczos which internally uses QL algorithm
PCG Lanczos	**MODOPT**, LANPCG	Symmetric (but not applicable for buckling)	K, M	None	Lanczos which internally uses QL algorithm
Subspace	**MODOPT**, SUBSP	Symmetric	K, M	None	Internally uses Subspace method with auto-shift
Unsymmetric	**MODOPT**, UNSYM	Unsymmetric matrices	K*, M*	None	Lanczos which internally uses QR algorithm
Damped	**MODOPT**, DAMP	Symmetric or unsymmetric damped systems	K*, C*, M*	None	Lanczos which internally uses QR algorithm

Procedure	Input	Usages	Applicable Matrices++	Reduction	Extraction Technique
QR Damped	**MODOPT**, QRDAMP	Symmetric or unsymmetric damped systems	K*, C*, M	Modal	QR algorithm for reduced modal damping matrix
++ K = stiffness matrix, C = damping matrix, M = mass or stress stiffening matrix, * = can be unsymmetric					

The Block Lanczos and PCG Lanczos methods both use Lanczos iterations to extract the requested eigenvalues. However, the Block Lanczos method uses the sparse direct solver, while the PCG Lanczos method uses the PCG iterative solver internally to solve the necessary system of equations at each Lanczos iteration.

14.14.1. Supernode Method

The Supernode (SNODE) solver is used to solve large, symmetric eigenvalue problems for many modes (up to 10,000 and beyond) in one solution. A supernode is a group of nodes from a group of elements. The supernodes for the model are generated automatically by the solver. This method first calculates eigenmodes for each supernode in the range of 0.0 to FREQE**RangeFact* (where *RangeFact* is specified by the **SNOPTION** command and defaults to 2.0), and then uses the supernode eigenmodes to calculate the global eigenmodes of the model in the range of FREQB to FREQE (where *FREQB* and *FREQE* are specified by the **MODOPT** command). Typically, this method offers faster solution times than Block Lanczos or PCG Lanczos if the number of modes requested is more than 200.

The Supernode solver uses an approximate method compared to the Block Lanczos and PCG Lanczos solutions. The accuracy of the Supernode solution can be controlled by the **SNOPTION** command. By default, the eigenmode accuracy is based on the frequency range used, as shown in the following table.

Frequency Range	Accuracy of Supernode solution
0 - 100 Hz	0.01 percent error
100 - 200 Hz	0.05 percent error
200 - 400 Hz	0.20 percent error
400 - 1000 Hz	1.00 percent error
1000 Hz and higher	3.0 - 5.0 percent error

Typically, the reason for seeking many modes is to perform a subsequent mode superposition or PSD analysis to solve for the response in a higher frequency range. The error introduced by the Supernode solver (shown in the table above) is small enough for most engineering purposes. You can use the **SNOPTION** command to increase the accuracy of the solution, but at the cost of increased computing time. Increasing the value of *RangeFact* (on the **SNOPTION** command) results in a more accurate solution.

In each step of the Supernode eigenvalue calculation, a Sturm check is performed. The occurrence of missing modes in the Supernode calculation is rare.

The lumped mass matrix option (**LUMPM**,ON) is not allowed when using the Supernode mode extraction method. The consistent mass matrix option will be used regardless of the **LUMPM** setting.

14.14.2. Block Lanczos

The Block Lanczos eigenvalue extraction method (accessed with **MODOPT**,LANB or **BUCOPT**,LANB) is available for large symmetric eigenvalue problems.

The block shifted Lanczos algorithm is a variation of the classical Lanczos algorithm, where the Lanczos recursions are performed using a block of vectors, as opposed to a single vector. Additional theoretical details on the classical Lanczos method can be found in Rajakumar and Rogers([196] (p. 931)).

A block shifted Lanczos algorithm, as found in Grimes et al.([195] (p. 931)) is the theoretical basis of the eigensolver. The Block Lanczos method employs an automated shift strategy, combined with a Sturm sequence check, to extract the number of eigenvalues requested. The Sturm sequence check ensures that the requested number of eigenfrequencies beyond the user supplied shift frequency (*FREQB* on the **MODOPT** command) is found without missing any modes. At the end of the Block Lanczos calculation, the solver performs a Sturm sequence check automatically. This check computes the number of negative pivots encountered in the range that the minimum and maximum eigenvalues encompass. This number will match the number of converged eigenvalues unless some eigenvalues have been missed. Block Lanczos will report the number of missing eigenvalues, if any.

Use of the Block Lanczos method for solving larger models (1,000,000 DOF, for example) with many constraint equations (CE) can require a significant amount of computer memory. The alternative method of PCG Lanczos, which internally uses the PCG solver, could result in savings of memory and computing time.

14.14.3. PCG Lanczos

The theoretical basis of this eigensolver is found in Grimes et al.([195] (p. 931)), which is the same basis for the Block Lanczos eigenvalue extraction method. However, the implementaion differs somewhat from the Block Lanczos eigensolver, in that the PCG Lanczos eigensolver:

- does not employ an automated shift strategy during the eigenvalue analysis.
- does not perform a Sturm sequence check by default.
- is only available for modal analyses and is not applicable to buckling analyses.

14.14.4. Unsymmetric Method

The unsymmetric eigensolver (accessed with **MODOPT**,UNSYM) is applicable whenever the system matrices are unsymmetric. For example, an acoustic fluid-structure interaction problem using FLUID30 elements results in unsymmetric matrices. Also, certain problems involving the input matrix element MATRIX27 and/or COMBI214 element, such as in brake squeal problems, can give rise to unsymmetric system matrices. A generalized eigenvalue problem given by the following equation:

$$[K]\{\phi_i\} = \lambda_i[M]\{\phi_i\} \tag{14.193}$$

can be setup and solved using the mode-frequency analysis (**ANTYPE**,MODAL). The matrices [K] and [M] are the system stiffness and mass matrices, respectively. Either or both [K] and [M] can be unsymmetric. $\{\varphi_i\}$ is called the eigenvector. It represents the right eigenvector. The left eigenvector, which is extracted for mode superposition harmonic analysis (**MODOPT** with *ModType* = BOTH), satisfies the following equation:

$$[K]^T\{\phi_i^L\} = \lambda_i[M]^T\left\{\phi_i^L\right\} \tag{14.194}$$

where:

$\left\{\phi_i^L\right\}$ = i^{th} left eigenmode

The method employed to solve the unsymmetric eigenvalue problem is a subspace approach based on a method designated as Frequency Derivative Method. The FD method uses an orthogonal set of Krylov sequence of vectors:

$$[Q] = [\{q_1\}\{q_2\}\{q_3\}\dots\{q_m\}] \tag{14.195}$$

To obtain the expression for the sequence of vectors, the generalized eigenvalue Equation 14.193 (p. 729) is differentiated with respect to λ_i to get:

$$-[M]\{\phi_i\} = \{0\} \tag{14.196}$$

Substituting Equation 14.196 (p. 730) into Equation 14.193 (p. 729) and rearranging after applying a shift s, the starting expression for generating the sequence of vectors is given by:

$$[[K] - s[M]]\{q_1\} = \{q_0\} \tag{14.197}$$

$$\{q_0\} = -[M]\{\tilde{q}_0\} \tag{14.198}$$

where:

$\{\tilde{q}_0\}$ = vector of random numbers

s = an initial shift

The general expression used for generating the sequence of vectors is given by:

$$[[K] - s[M]]\{q_{j+1}\} = \{\tilde{q}_j\} \tag{14.199}$$

This matrix equation is solved by a sparse matrix solver (**EQSLV**, SPARSE). However, an explicit specification of the equation solver (**EQSLV** command) is not needed.

The subspace made of these derivatives allows the program to find the closest eigensolutions from this shift point. The shift value *s* is initially determined using the *FREQB* value on the **MODOPT** command.

For a large number of eigenvalues, the UNSYM extraction algorithm is able to move automatically to a new shift if the first solve only finds a subset of eigensolutions. This process will be repeated until all the required eigenvalues are found, unless the algorithm fails several times to find any accurate eigenvalues.

A subspace transformation of Equation 14.193 (p. 729) is performed using the sequence of orthogonal vectors which leads to the reduced eigenproblem:

$$[K^*]\{y_i\} = \mu_i[M^*]\{y_i\} \tag{14.200}$$

where:

$[K^*] = [Q^T]\ [K]\ [Q]$

$[M^*] = [Q^T]\ [M]\ [Q]$

The eigenvalues of the reduced eigenproblem (Equation 14.200 (p. 730)) are extracted using a direct eigenvalue solution procedure. The eigenvalues μ_i are the approximate eigenvalues of the original eigen-

problem and they converge to λ_i with increasing subspace size m. The converged eigenvectors are then computed using the subspace transformation equation:

$$\{\phi_i\} = [Q]\{y_i\} \tag{14.201}$$

For the unsymmetric modal analysis, the real part (ω_i) of the complex frequency is used to compute the element kinetic energy.

This method does not perform a Sturm Sequence check for possible missing modes. At the lower end of the spectrum close to the shift (input as *FREQB* on **MODOPT** command), the frequencies usually converge without missing modes.

14.14.5. Subspace Method

The subspace eigensolver (accessed with **BUCOPT**,SUBSP and **MODOPT**,SUBSP) is applicable for buckling and modal analyses in which the system matrices are symmetric. This eigensolver uses basically the same algorithm as the unsymmetric eigensolver (accessed with **MODOPT**,UNSYM) to solve the generalized eigenvalue problem given by the following equation.

For buckling analysis:

$$\left([K] + \lambda_i[S]\right)\{\Psi_i\} = \{0\} \tag{14.202}$$

where:

[K] = structural stiffness matrix
[S] = stress stiffness matrix
λ_i = ith eigenvalue (used to multiply the loads that generated [S])
Ψ_i = ith eigenvector of displacements

For modal analysis:

$$[K]\{\phi_i\} = \lambda_i[M]\{\phi_i\} \tag{14.203}$$

where:

[K] = structural stiffness matrix
[M] = mass matrix
λ_i = ith eigenvalue
φ_i = ith eigenvector of displacements

The advantage of the SUBSP method over the Block Lanczos method is that both the [K] and [S] / [M] matrices can be indefinite at the same time.

Buckling Analysis

Using the options of the **BUCOPT** command, the program can find the eigenvalues in one of two ways:

- In a specified load multiplier range of interest (*RangeKey* = RANGE), or
- Around an initial shift (*RangeKey* = CENTER).

Note that when using the CENTER option, the automatic (internal) shifting strategy of the algorithm is disabled. The eigenvalues found around the initial shift (CENTER value) are available; the solver may not find eigenvalues far away from the CENTER value.

Modal Analysis

Some options of the subspace algorithm can be set using the **SUBOPT** command. These include memory management and Sturm sequence check. By default, the Sturm check is turned off.

14.14.6. Damped Method

The damped eigensolver (accessed with **MODOPT**,DAMP) is applicable only when the system damping matrix needs to be included in Equation 14.192 (p. 727), where the eigenproblem becomes a quadratic eigenvalue problem given by:

$$[K]\{\phi_i\} + \bar{\lambda}_i[C]\{\phi_i\} = -\bar{\lambda}_i^2[M]\{\phi_i\} \tag{14.204}$$

where:

$\bar{\lambda}_i = \sqrt{-\lambda_i}$ (defined below)
[C] = damping matrix

Matrices may be symmetric or unsymmetric.

The method employed to solve the damped eigenvalue problem is the same as for the UNSYM option. We first transform the initial quadratic equation (Equation 14.204 (p. 732)) in a linear form applying the variable substitutions:

$$[\bar{K}] = \begin{pmatrix} K & 0 \\ 0 & 1 \end{pmatrix}$$

$$[\bar{M}] = \begin{pmatrix} -C & -M \\ 1 & 0 \end{pmatrix}$$

To form the equivalent UNSYM eigenvalue problem.

$$[\bar{K}]\{\bar{\phi}_i\} = \lambda_i[\bar{M}]\{\bar{\phi}_i\} \tag{14.205}$$

Solutions of Equation 14.204 (p. 732) and Equation 14.205 (p. 732) are equivalent, except that only the first-half part of the eigenvevctors $\bar{\phi}_i$ is considered.

The UNSYM method uses Equation 14.205 (p. 732). The default blocksize value to solve a Quadratic Damp Eigenproblem is set to four. This value can be controlled using the blocksize parameter of the **MODOPT** command.

This method does not perform a Sturm Sequence check for possible missing modes. At the lower end of the spectrum, close to the shift (input as *FREQB* on the **MODOPT** command), the frequencies usually converge without missing modes. Furthermore, this method does not employ an automated shift strategy during the eigenvalue analysis.

For the damped modal analysis, the imaginary part ω_i of the complex frequency is used to compute the element kinetic energy.

14.14.7. QR Damped Method

The QR damped method (accessed with **MODOPT**,QRDAMP) is a procedure for determining the complex eigenvalues and corresponding eigenvectors of damped linear systems. This eigensolver allows for nonsymmetric [K] and [C] matrices. The eigensolver is computationally efficient compared to the damped eigensolver (**MODOPT**,DAMP). This method employs the modal orthogonal coordinate transformation of system matrices to reduce the eigenproblem into the modal subspace. The QR algorithm is then used to calculate eigenvalues of the resulting quadratic eigenvalue problem in the modal subspace.

The equations that follow apply to Rayleigh and element-based damping in a QR Damped modal analysis. For the equations with constant structural damping (input with **DMPSTR** or **MP**,DMPR), refer to QR Damped Method with Constant Structural Damping (p. 734).

The equations of elastic structural systems without external excitation can be written in the following form:

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{0\} \tag{14.206}$$

(See Equation 15.5 (p. 764) for definitions).

It has been recognized that performing computations in the modal subspace is more efficient than in the full eigen space. The stiffness matrix [K] can be symmetrized by rearranging the unsymmetric contributions; that is, the original stiffness matrix [K] can be divided into symmetric and unsymmetric parts. By dropping the damping matrix [C] and the unsymmetric contributions of [K], the symmetric Block Lanczos eigenvalue problem is first solved to find real eigenvalues and the corresponding eigenvectors. In the present implementation, the unsymmetric element stiffness matrix is zeroed out for Block Lanczos eigenvalue extraction. Following is the coordinate transformation (see Equation 14.100 (p. 698)) used to transform the full eigen problem into modal subspace:

$$\{u\} = [\Phi]\{y\} \tag{14.207}$$

where:

$[\Phi]$ = eigenvector matrix normalized with respect to the mass matrix [M]
$\{y\}$ = vector of modal coordinates

By using Equation 14.207 (p. 733) in Equation 14.206 (p. 733), we can write the differential equations of motion in the modal subspace as follows:

$$[I]\{\ddot{y}\} + [\Phi]^T[C][\Phi]\{\dot{y}\} + ([\Lambda^2] + [\Phi]^T[K_{unsym}][\Phi])\{y\} = \{0\} \tag{14.208}$$

where:

$[\Lambda^2]$ = a diagonal matrix containing the first n eigen frequencies ω_i

For classically damped systems, the modal damping matrix $[\Phi]^T[C][\Phi]$ is a diagonal matrix with the diagonal terms being $2\xi_i\omega_i$, where ξ_i is the damping ratio of the i-th mode. For non-classically damped systems, the modal damping matrix is either symmetric or unsymmetric. Unsymmetric stiffness contributions of the original stiffness are projected onto the modal subspace to compute the reduced unsymmetric modal stiffness matrix $[\Phi]^T [K_{unsym}] [\Phi]$.

Introducing the 2n-dimensional state variable vector approach, Equation 14.208 (p. 733) can be written in reduced form as follows:

$$[I]\{\dot{z}\} = [D]\{z\} \tag{14.209}$$

where:

$$\{z\} = \begin{Bmatrix} \{y\} \\ \{\dot{y}\} \end{Bmatrix} \tag{14.210}$$

$$[D] = \begin{bmatrix} [O] & [I] \\ -[\Lambda^2] - [\Phi]^T[K_{unsym}][\Phi] & -[\Phi]^T[C][\Phi] \end{bmatrix} \tag{14.211}$$

The 2n eigenvalues of Equation 14.209 (p. 734) are calculated using the QR algorithm (Press et al.([254] (p. 935))). The inverse iteration method (Wilkinson and Reinsch([357] (p. 941))) is used to calculate the complex modal subspace eigenvectors. The full complex eigenvectors, $\{\psi\}$, of original system is recovered using the following equation:

$$\{\psi\} = [\Phi]\{z\} \tag{14.212}$$

Out of 2n solutions, only n are output for post-processing. When the complex eigenvalues are complex conjugate pairs, only the positive imaginary solution (positive frequency) is retained. In the case of high damping, all overdamped modes are also retained.

14.14.7.1. QR Damped Method with Constant Structural Damping

When a constant structural damping ratio is defined in a QR damped modal analysis and the complex mode shapes are requested (*Cpxmod* = YES on the **MODOPT** command), the state-space matrix [D] from Equation 14.211 (p. 734) becomes complex. This can be written as:

$$[D] = \begin{bmatrix} [0] & [I] \\ -[\Lambda] - [\Phi]^t[K_{usym}][\Phi] & [0] \end{bmatrix} + j\begin{bmatrix} [0] & [0] \\ -[\Phi]^t[K_{imag}][\Phi] & [0] \end{bmatrix} \tag{14.213}$$

where:

$$[K_{imag}] = g[K] + \sum_{j=1}^{N_m} 2m_j[K_j]$$

g = constant structural damping coefficient (input with the **DMPSTR** command)
N_m = number of materials with **MP**,DMPR input
m_j = constant structural damping coefficient for material j (input with **MP**,DMPR)
$[K_j]$ = portion of structural stiffness matrix based on material j

14.14.8. Shifting

Various shifting strategies are used by most of the extraction methods in an effort to improve the accuracy, robustness, and efficiency of the algorithms. The logic on how those shift values are chosen is discussed in this section.

In some cases it is desirable to shift the values of eigenvalues either up or down. These fall in two categories:

1. Shifting down, so that the solution of problems with rigid body modes does not require working with a singular matrix.

2. Shifting up, so that the bottom range of eigenvalues will not be computed, because they had effectively been converted to negative eigenvalues. This will, in general, result in better accuracy for the higher modes. The shift introduced is:

$$\lambda = \lambda_o + \lambda_i \tag{14.214}$$

where:

λ = desired eigenvalue
λ_0 = eigenvalue shift
λ_i = eigenvalue that is extracted

λ_0, the eigenvalue shift is computed as:

$$\lambda_o = \begin{cases} s_b & \text{if buckling analysis } (s_b \text{ is input as } \textit{SHIFT} \text{ on } \textbf{BUCOPT} \text{ command}) \\ \text{or} & \\ (2\pi s_m)^2 & \text{if modal analysis } (s_m \text{ is input as } \textit{FREQB} \text{ on } \textbf{MODOPT} \text{ command}) \end{cases} \tag{14.215}$$

When using the Block Lanczos or PCG Lanczos method, if no user input is given for *SHIFT* (**BUCOPT** command) or *FREQB* (**MODOPT** command), the following logic is used:

$$\lambda_0 = \frac{1}{n \sum \frac{M_{ii}}{K_{ii}}} \tag{14.216}$$

where M_{ii} and K_{ii} are the diagonals of the [M] and [K] matrices, respectfully. The summation is taken over all terms where $K_{ii} \neq 0$ and where $\frac{M_{ii}}{K_{ii}} < 10e4$. The number of such terms is *n*.

When using the PCG Lanczos method, if a *Lev_Diff* value of 1, 2, 3, or 4 is chosen (either automatically or by the user, see **PCGOPT**), then $-\lambda_0$ from Equation 14.216 (p. 735) is used when rigid body modes are detected in order to avoid working with indefinite matrices with the PCG iterative solver. When using *Lev_Diff* = 5, λ_0 is used as with the Block Lanczos method.

Equation 14.214 (p. 735) is combined with Equation 14.192 (p. 727) to give:

$$[K]\{\phi_i\} = (\lambda_o + \lambda_i)[M]\{\phi_i\} \tag{14.217}$$

Rearranging,

$$([K] - \lambda_o[M])\{\phi_i\} = \lambda_i[M]\{\phi_i\} \tag{14.218}$$

or

$$[K]'\{\phi_i\} = \lambda_i[M]\{\phi_i\} \tag{14.219}$$

where:

$[K]' = [K] - \lambda_0 [M]$

It may be seen that if [K] is singular, as in the case of rigid body motion, [K]' will not be singular if [M] is not totally zero (which is normally true) and if λ_0 is input as a non-zero number.

Once λ_i is computed, λ is computed from Equation 14.214 (p. 735) and reported.

14.14.9. Repeated Eigenvalues

Repeated roots or eigenvalues are possible to compute. This occurs, for example, for a thin, axisymmetric pole. Two independent sets of orthogonal motions are possible.

In these cases, the eigenvectors are not unique, as there are an infinite number of correct solutions. However, in the special case of two or more identical but disconnected structures run as one analysis, eigenvectors may include components from more than one structure. To reduce confusion in such cases, it is recommended to run a separate analysis for each structure.

14.14.10. Complex Eigensolutions

For problems involving spinning structures with gyroscopic effects, and/or damped structural eigenfrequencies, the eigensolutions obtained with the Damped Method (p. 732) and QR Damped Method (p. 733) are complex. The eigenvalues $\bar{\lambda}_i$ are given by:

$$\bar{\lambda}_i = \sigma_i \pm j\omega_i \tag{14.220}$$

where:

$\bar{\lambda}_i$ = complex eigenvalue
σ_i = real part of the eigenvalue (stability value)
ω_i = imaginary part of the eigenvalue (damped frequency)
$j = \sqrt{-1}$

The dynamic response of the system is given by:

$$\{u_i\} = \{\phi_i\} e^{\bar{\lambda}_i t} \tag{14.221}$$

where:

t = time

The ith eigenvalue is stable if σ_i is negative and unstable if σ_i is positive.

Modal damping ratio

The modal damping ratio is given by:

$$\alpha_i = \frac{-\sigma_i}{|\lambda_i|} = \frac{-\sigma_i}{\sqrt{\sigma_i^2 + \omega_i^2}} \tag{14.222}$$

where:

α_i = modal damping ratio of the ith eigenvalue

It is the ratio of the actual damping to the critical damping.

Logarithmic decrement

The logarithmic decrement represents the logarithm of the ratio of two consecutive peaks in the dynamic response (Equation 14.221 (p. 736)). It can be expressed as:

$$\delta_i = \ln\left(\frac{u_i(t)}{u_i(t+T_i)}\right) = 2\pi\frac{\sigma_i}{\omega_i} \tag{14.223}$$

where:

δ_i = logarithmic decrement of the ith eigenvalue
T_i = damped period of the ith eigenvalue defined by:

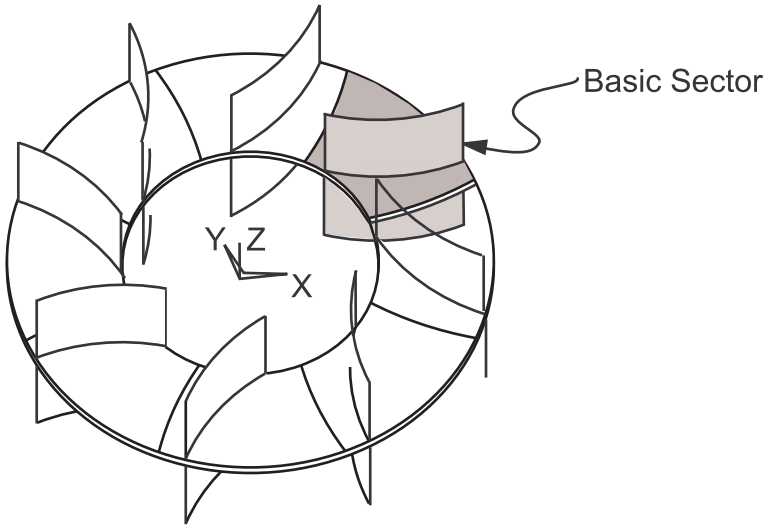
$$T_i = \frac{2\pi}{\omega_i} \tag{14.224}$$

14.15. Analysis of Cyclic Symmetric Structures

14.15.1. Modal Analysis

Given a cyclic symmetric (periodic) structure such as a fan wheel, a modal analysis can be performed for the entire structure by modelling only one sector of it. A proper basic sector represents a pattern that, if repeated n times in cylindrical coordinate space, would yield the complete structure.

Figure 14.14: Typical Cyclic Symmetric Structure

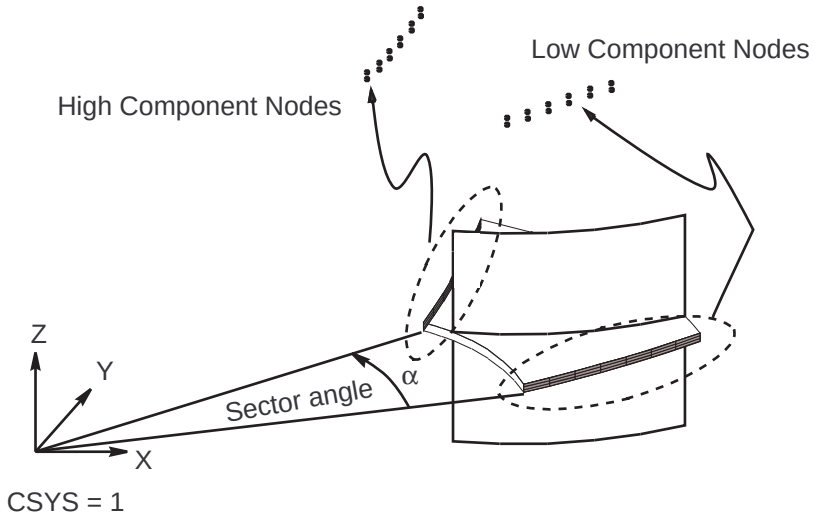


In a flat circular membrane, mode shapes are identified by harmonic indices. For more information, see Cyclic Symmetry Analysis of the *Advanced Analysis Guide*.

Constraint relationships (equations) can be defined to relate the lower ($\theta = 0$) and higher ($\theta = \alpha$, where α = sector angle) angle edges of the basic sector to allow calculation of natural frequencies related to a given number of harmonic indices. The basic sector is duplicated in the modal analysis to satisfy the

required constraint relationships and to obtain nodal displacements. This technique was adapted from Dickens([148] (p. 929)).

Figure 14.15: Basic Sector Definition



Constraint equations relating the lower and higher angle edges of the two sectors are written:

$$\begin{Bmatrix} u'_A \\ u'_B \end{Bmatrix} = \begin{bmatrix} \cos k\alpha & \sin k\alpha \\ -\sin k\alpha & \cos k\alpha \end{bmatrix} \begin{Bmatrix} u_A \\ u_B \end{Bmatrix} \tag{14.225}$$

where:

u_A, u_B = calculated displacements on lower angle side of basic and duplicated sectors (A and B, respectively)

u'_A, u'_B = displacements on higher angle side of basic and duplicated sectors (A and B, respectively) determined from constraint relationships

$$k = \text{harmonic index} = 0,1,2\ldots \begin{cases} N/2 & \text{if N is even} \\ \dfrac{N-1}{2} & \text{if N is odd} \end{cases}$$

$\alpha = 2\pi/N$ = sector angle
N = number of sectors in 360°

Three basic steps in the procedure are briefly:

1. The **CYCLIC** command in /PREP7 automatically detects the cyclic symmetry model information, such as edge components, the number of sectors, the sector angles, and the corresponding cyclic coordinate system.

2. The **CYCOPT** command in /SOLU generates a duplicated sector and applies cyclic symmetry constraints (Equation 14.225 (p. 738)) between the basic and the duplicated sectors.

3. The **/CYCEXPAND** command in /POST1 expands a cyclically symmetry response by combining the basic and the duplicated sectors results (Equation 14.226 (p. 739)) to the entire structure.

14.15.2. Complete Mode Shape Derivation

The mode shape in each sector is obtained from the eigenvector solution. The displacement components (x, y, or z) at any node in sector j for harmonic index k, in the full structure is given by:

$$u = u_A \cos(n-1)k\alpha - u_B \sin(n-1)k\alpha \tag{14.226}$$

where:

n = sector number, varies from 1 to N
u_A = basic sector displacement
u_B = duplicate sector displacement

If the mode shapes are normalized to the mass matrix in the mode analysis (Nrmkey option in the **MODOPT** command), the normalized displacement components in the full structure is given by

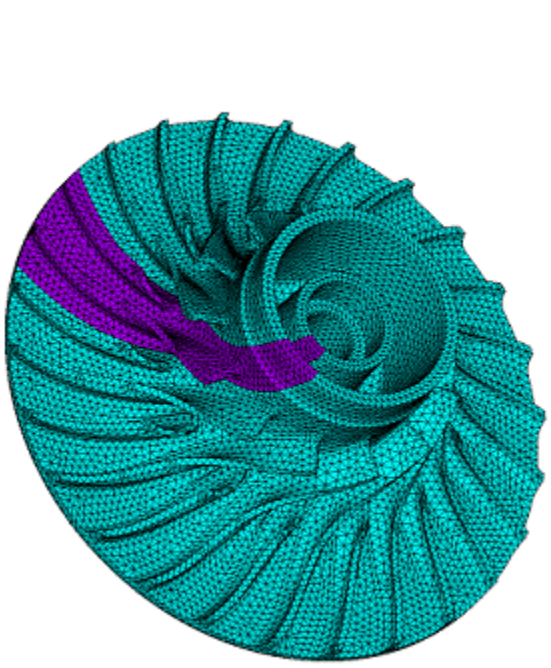
$$u_{normalized} = \begin{cases} \dfrac{u}{\sqrt{N}} & \text{if } k = 0 \text{ or } k = N/2 \\ \dfrac{u}{\sqrt{N/2}} & \end{cases} \tag{14.227}$$

The complete procedure addressing static, modal, and prestressed modal analyses of cyclic symmetric structures is contained in Cyclic Symmetry Analysis of the *Advanced Analysis Guide*.

14.15.3. Mode-Superposition Harmonic Analysis

A bladed disk (axial or centrifugal) is cyclically symmetric about its fundamental sector, as shown in Figure 14.16: Full Model with the Cyclic-Symmetric Sector Highlighted (p. 739). These bladed disks are typically excited by rotationally symmetric excitations which produce forced vibrations.

Figure 14.16: Full Model with the Cyclic-Symmetric Sector Highlighted



The equation of motion for harmonically-varying forced response is:

$$\left(\left[K^s\right]-\Omega^2\left[M^s\right]\right)\left\{u^s\right\}=\left\{F^s\right\} \tag{14.228}$$

where:

$[K^s]$ = stiffness matrix of the bladed disk

$[M^s]$ = mass matrix of the bladed disk

$\{F^s\}$ = load vector

$\{u^s\}$ = displacement vector

Ω = forcing frequency

The superscript s implies the matrices refer to the entire 360° system. Damping is ignored in this equation and is discussed in Damping (p. 745).

14.15.3.1. Transform to Modal Coordinates

Equation 14.228 (p. 740) can be transferred into modal space by:

$$\left\{u^s\right\}=\left[\Phi^s\right]\left\{a^s\right\} \tag{14.229}$$

where:

$[\Phi^s]$ = the set eigenvectors of the full system

$\{a^s\}$ = the vector of modal coordinates

Substituting this into Equation 14.228 (p. 740) and pre-multiplying by $[\Phi^s]^T$ yields the forced response equation in terms of modal coordinates:

$$\left(\left[\Lambda^s\right]-\Omega^2\left[I\right]\right)\left\{a^s\right\}=\left[\Phi^s\right]^T\left\{F^s\right\} \tag{14.230}$$

where:

$[\Lambda^s]$ = the diagonal matrix for the system natural frequencies squared

[I] = the identity matrix

The number of eigenvectors (modes) used in Equation 14.230 (p. 740) is much less than the total number of degrees of freedom, so that Equation 14.230 (p. 740) is significantly smaller system than Equation 14.228 (p. 740).

14.15.3.2. Cyclic Coordinates

A cyclically-symmetric structure has the following transformation from cyclic coordinates to physical coordinates:

$$\left\{x\right\}=\left(\left[F\right]\otimes\left[I\right]\right)\left\{\tilde{x}_h\right\} \tag{14.231}$$

where:

$\left\{\tilde{x}_h\right\}$ = cyclic (or harmonic) coordinates for harmonic index h

[*F*] = real-valued Fourier matrix

$\otimes$ denotes the Kronecker product

The system eigenvectors can also be expressed in terms of their harmonics:

$$\left[\Phi^s\right] = \left(\left[F\right] \otimes \left[I\right]\right) \tilde{B}_{diag}\left(\left[\tilde{\Phi}\right]_h\right) \tag{14.232}$$

where:

$\left[\tilde{\Phi}\right]_h$ = the real-valued cyclic modes corresponding to harmonic h

$\tilde{B}_{diag}()$ = a pseudo-block diagonal matrix. A block is 1×1 for h = 0 and N/2 if N is even, and 2×2 otherwise. N is the number of sectors (**CYCLIC**).

Substituting Equation 14.232 (p. 741) into Equation 14.230 (p. 740) gives the frequency response equation in cyclic coordinates:

$$\left(\left[\Lambda^s\right] - \Omega^2\left[I\right]\right)\left\{a^s\right\} = \left\{f^s\right\} \tag{14.233}$$

where the modal load $\{f^s\}$ is:

$$\left\{f^s\right\} = \tilde{B}_{diag}\left(\left[\tilde{\Phi}\right]_h^T\right)\left(\left[F\right]^T \otimes \left[I\right]\right)\left\{F^s\right\} \tag{14.234}$$

14.15.3.3. Properties of Paired Eigenmodes

The Fourier transformation expression:

$$\left(\left[F\right] \otimes \left[I\right]\right) \tilde{B}_{diag}\left(\left[\tilde{\Phi}\right]_h\right) \tag{14.235}$$

appears in Equation 14.232 (p. 741) and Equation 14.234 (p. 741) (transposed), where $\left[\tilde{\Phi}\right]_h$ are the harmonic modes. $\tilde{B}_{diag}\left(\left[\tilde{\Phi}_h\right]\right)$ has the structure:

$$\tilde{B}_{diag}\left(\left[\tilde{\Phi}\right]_h\right) = \begin{bmatrix} \left[\tilde{\Phi}\right]_0 & & & & \\ & \left[\tilde{\Phi}_{A\mathbb{R}}\right]_1 & \left[\tilde{\Phi}_{B\mathbb{R}}\right]_1 & & \\ & \left[\tilde{\Phi}_{A\mathbb{I}}\right]_1 & \left[\tilde{\Phi}_{B\mathbb{I}}\right]_1 & & \\ & & & \ddots & \\ & & & & \left[\tilde{\Phi}\right]_{N/2} \end{bmatrix} \tag{14.236}$$

The harmonics h (0 < h < N/2) appear as 2×2 sub-blocks, each column being one of the repeated modes and the rows corresponding to the real and imaginary (basic and duplicate sector) solutions. The first mode in a pair is denoted with subscript A and the second mode in the pair is denoted with subscript B.

The real and imaginary terms of each double mode are related and obey the following:

$$\left[\tilde{\Phi}_{A\mathbb{I}}\right] = g_A\left[\tilde{\Phi}_{B\mathbb{R}}\right] \tag{14.237}$$

where g_A = 1 if the mode shapes $\left[\tilde{\Phi}_{A\mathbb{I}}\right]$ and $\left[\tilde{\Phi}_{B\mathbb{R}}\right]$ have the same sign, otherwise g_A = -1. Likewise:

$$\left[\tilde{\Phi}_{B\mathbb{I}}\right] = g_B\left[\tilde{\Phi}_{A\mathbb{R}}\right] \tag{14.238}$$

where $g_B = -g_A$. This property leads to the .MODE file needing to only contain the real (base) sector degrees of freedom for each mode shape. Likewise, the stresses and strains for the expanded modes on the .RST file only contain the results for the base elements (and not the duplicate elements).

14.15.3.4. Forcing

The forcing function is assumed to be time harmonic as well as spatially harmonic in the circumferential direction. The force on any sector n can then be related to the force on the basic sector by only a phase shift:

$$\{F_n\} = \{F\} e^{i\phi(n-1)} \tag{14.239}$$

where:

{F} = complex load vector on the basic sector

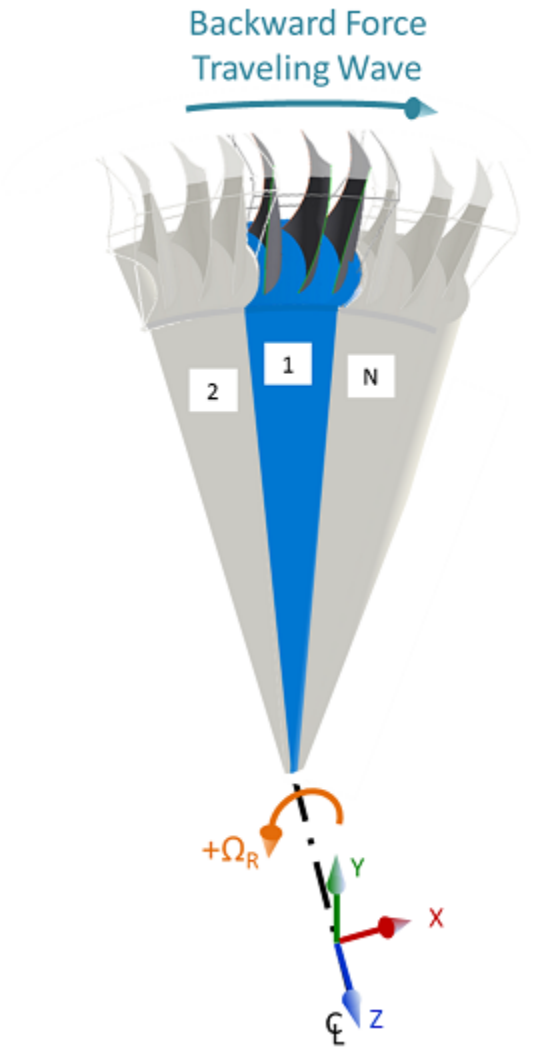
$\phi = \frac{2\pi C}{N}$ = phase of excitation (inter-blade phase angle, IBPA) on sector n

C = engine order (EO) excitation (**CYCFREQ**)

N = number of sectors

Equation 14.239 (p. 742) defines the distribution of force over the blades using the blade numbering convention shown in Figure 14.17: Forcing Sign and Numbering Convention (p. 743). With this convention, blade 2 leads blade 1 by the inter-blade phase angle $\phi = \frac{2\pi C}{N}$; it is subjected to the force first and blade 1 is subjected to the same force after a rotation of φ radians. The forcing wave travels in the direction shown in Figure 14.17: Forcing Sign and Numbering Convention (p. 743), which is a backward traveling wave with respect to the rotation Ω_R (**OMEGA** or **CMOMEGA**).

Figure 14.17: Forcing Sign and Numbering Convention



The right hand side of Equation 14.233 (p. 741) is Equation 14.234 (p. 741):

$$\left\{f^s\right\} = \tilde{B}_{diag}\left(\left[\tilde{\Phi}\right]_h^T\right)\left([F]^T \otimes [I]\right)\left\{F^s\right\} \tag{14.240}$$

Equation 14.240 (p. 743) can be written as the sum of each sector, if only blade loads are considered:

$$\left\{f^s\right\} = \Sigma\tilde{B}_{diag}\left(\left[\tilde{\Phi}\right]_h^T\right)\left([F_n]^T \otimes [I]\right)\{F_n\} \tag{14.241}$$

where:

$[F_n]$ = the n^{th} row of the Fourier matrix $[F]$

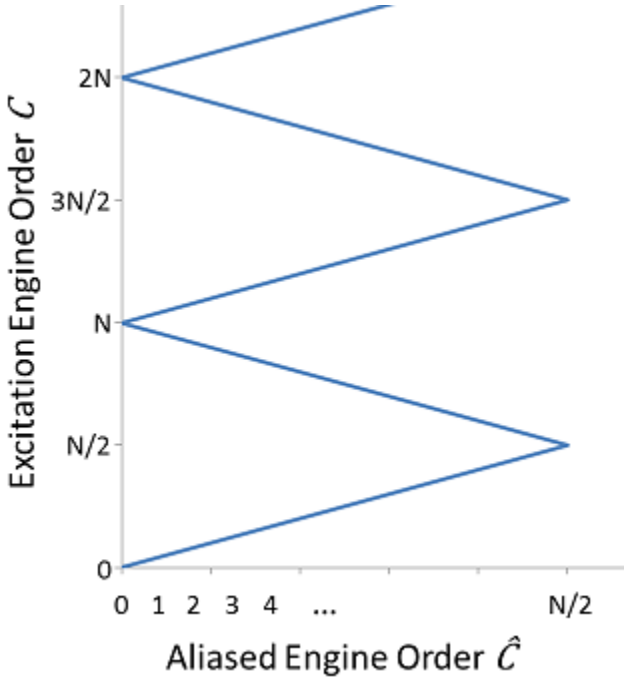
While the engine order can take any integer value, a given engine order C will only excite a certain harmonic index. This aliased engine order $\hat{C}$ is determined from the input engine order (the number of preceding stators) as outlined in the following table:

Table 14.2: Aliased Engine Order (Excited Harmonic Index)

Engine Order C		Aliased Engine Order $\hat{C}$
N Even	N Odd	
$C \leq N / 2$	$C \leq (N - 1) / 2$	C
$N / 2 < C \leq N - 1$	$2(N + 1) \leq C \leq N$	N - C
$N \leq C \leq 3N / 2$	$N \leq C \leq (3N - 1) / 2$	C - N
$3N / 2 \leq C \leq 2N$	$(3N +1) / 2 \leq C \leq 2N$	2N - C
…	…	…

This leads to the well-known zigzag diagram in Figure 14.18: Zigzag Diagram for an Even Number of Sectors (p. 744). The positive slopes are forward traveling waves and the negative slopes are backward traveling.

Figure 14.18: Zigzag Diagram for an Even Number of Sectors



Due to the orthogonality of the engine order phasing with the sine and cosine terms of the Fourier matrix [*F*], only the harmonic indices h = $\hat{C}$ of [F_n] are non-zero.

The sign of $\hat{C}$ is determined as follows:

1. If the aliased engine order is on a negative slope of Figure 14.18: Zigzag Diagram for an Even Number of Sectors (p. 744), it is assigned a negative value.

2. If the rotation is opposite that illustrated in Figure 14.17: Forcing Sign and Numbering Convention (p. 743), in other words a negative Ω_R in the cyclic coordinate system, the value from the first step is negated.

3. If the engine order was input as a negative value, the value after the second step is negated again.

This process ensures that the force is correctly applied to the blades via Equation 14.240 (p. 743), while respecting the blade numbering, rotation direction, and the relationship between engine order and nodal diameter.

Returning to the harmonic equation of motion, Equation 14.233 (p. 741), the force is assumed to be harmonic and of the form:

$$\left\{f^s\right\} e^{i\Omega t} \tag{14.242}$$

The excitation frequency Ω, and the corresponding frequency sweep range $\Omega \pm \Delta\Omega$ (**HARFRQ**) is typically related to the rotor speed by:

$$\Omega = C\Omega_R \tag{14.243}$$

The maximum response will occur when this excitation frequency crosses a natural frequency of that nodal diameter, for instance when $C\Omega_R = \omega_h$.

14.15.3.5. Damping

Damping may be included in two forms:

- Global structural damping g (**DMPSTR**)
- Global Rayleigh damping α and β (**ALPHAD** and **BETAD**)

The damping matrix in cyclic modal coordinates is:

$$\tilde{C} = \left(\beta + \frac{2}{\Omega} g\right)\left[\Lambda^s\right] + \alpha[I] \tag{14.244}$$

where $[\Lambda^s]$ is the diagonal matrix of system frequencies. The frequency response equation of motion, including damping, is:

$$\left((1 - i2g)\left[\Lambda^s\right] + i\Omega\left(\beta\left[\Lambda^s\right] + \alpha[I]\right) - \Omega^2[I]\right)\left\{a^s\right\} = \left\{f^s\right\} \tag{14.245}$$

14.15.3.6. Expansion to Output Quantities

Once the vector of modal coordinates $\{a^s\}$ is obtained for a given excitation frequency Ω (Equation 14.245 (p. 745)), we can expand back to the physical solution quantities for the entire bladed disk. The displacements of sector n are determined by:

$$\{u_n\} = \left(\left[F_n\right] \otimes [I]\right)\tilde{B}_{diag}\left(\left[\tilde{\Phi}\right]_h\right)\left\{a^s\right\} \tag{14.246}$$

Stresses and strains can be similarly evaluated by using the stress and strain mode shapes for $\left[\tilde{\Phi}\right]_h$.

14.15.4. Cyclic Symmetry Transformations

The cyclic symmetric solution sequences consist of three basic steps. The first step transforms applied loads to cyclic symmetric components using finite Fourier theory and enforces cyclic symmetry constraint equations (see Equation 14.225 (p. 738)) for each harmonic index (nodal diameter) (k = 0, 1, . . ., N/2).

Any applied load on the full 360° model is treated through a Fourier transformation process and applied on to the cyclic sector. For each value of harmonic index, k, the procedure solves the corresponding linear equation. The responses in each of the harmonic indices are calculated as separate load steps at the solution stage. The responses are expanded via the Fourier expansion (Equation 14.226 (p. 739)). They are then combined to get the complete response of the full structure in postprocessing.

The Fourier transformation from physical components, X, to the different harmonic index components, $\bar{X}$, is given by the following:

Harmonic Index, k = 0 (symmetric mode):

$$\bar{X}_{k=0} = \frac{1}{N}\sum_{n=1}^{N} X_n \tag{14.247}$$

Harmonic Index, 0 < k < N/2 (degenerate mode)

Basic sector:

$$(\bar{X}_k)_A = \frac{2}{N}\sum_{n=1}^{N} X_n \cos(n-1)k\alpha \tag{14.248}$$

Duplicate sector:

$$(\bar{X}_k)_B = \frac{2}{N}\sum_{n=1}^{N} X_n \sin(n-1)k\alpha \tag{14.249}$$

For N even only, Harmonic Index, k = N/2 (antisymmetric mode):

$$\bar{X}_{k=N/2} = \frac{1}{N}\sum_{n=1}^{N} (-1)^{(n-1)} X_n \tag{14.250}$$

where:

X = any physical component, such as displacements, forces, pressure loads, temperatures, and inertial loads

$\bar{X}$ = cyclic symmetric component

The transformation to physical components, X, from the cyclic symmetry, $\bar{X}$, components is recovered by the following equation:

$$X_n = \bar{X}_{k=0} + \sum_{k=1}^{K} [\bar{X}_{kA}\cos(n-1)k\alpha + \bar{X}_{kB}\sin(n-1)k\alpha] + (-1)^{n-1}\bar{X}_{k=N/2} \tag{14.251}$$

The last term $(-1)^{n-1}\bar{X}_{k=N/2}$ exists only for N even.

14.16. Mass Related Information

The mass related information (mass, center of mass, and mass moments of inertia) is printed out in a mass summary for all analyses that include mass.

Depending on the model, mass related information is calculated using one of two different methods. If the model is three-dimensional, a precise computation is performed, as detailed in Precise Calculation of Mass Related Information (p. 747). For all other cases, a lumped calculation is performed, as described in Lumped Calculation of Mass Related Information (p. 749) along with its limitations.

The mass summary by element type is always based on the basic calculation.

14.16.1. Precise Calculation of Mass Related Information

The total rigid body mass properties with respect to the origin is given by:

$$\left[M_{rig}\right] = \sum_{i=1}^{N} \left[D_i\right]^t \left[M_i\right]\left[D_i\right] \tag{14.252}$$

where:

$[M_{rig}]$ is the total rigid body mass properties of the model.
N is the number of elements.
$[D_i]$ is a matrix containing the six rigid body motion vectors of the i^{th} element. See Equation 15.145 (p. 801) for more information about these vectors.
$[M_i]$ is the mass matrix of the i^{th} element.

The total rigid body mass matrix can be partitioned as follows:

$$\left[M_{rig}\right] = \begin{bmatrix} \left[M_t\right] & \left[M_{tr}\right] \\ \left[M_{rt}\right] & \left[M_r\right] \end{bmatrix} \tag{14.253}$$

where:

$[M_t]$ is the translational mass matrix.
$[M_{tr}]$ and $[M_{rt}]$ are the coupled translational/rotational mass matrices.
$[M_r]$ is the rotational mass matrix and contains the mass moments of inertia.

The translational mass principal characteristics are obtained from the eigensolution of matrix $[M_t]$. The eigenvalues are the principal masses and the eigenvectors are the mass principal directions:

$$\begin{bmatrix} M_X & 0 & 0 \\ 0 & M_Y & 0 \\ 0 & 0 & M_Z \end{bmatrix} = \left[\Phi\right]^t \left[M_t\right]\left[\Phi\right] \tag{14.254}$$

where:

M_X, M_Y, and M_Z are the principal masses.
$[\Phi]$ is the matrix of the translational mass eigenvectors representing the mass principal directions.

If the principal masses are not equal, the center of mass location with respect to the principal axes is not unique. These locations are computed as:

$$cdm_X = \left(\frac{\hat{M}_{tr}(1,1)}{M_X}, \frac{-\hat{M}_{tr}(1,3)}{M_X}, \frac{\hat{M}_{tr}(1,2)}{M_X} \right) \tag{14.255}$$

$$cdm_Y = \left(\frac{\hat{M}_{tr}(2,3)}{M_Y}, \frac{\hat{M}_{tr}(2,2)}{M_Y}, \frac{-\hat{M}_{tr}(2,1)}{M_Y} \right) \tag{14.256}$$

$$cdm_Z = \left(\frac{-\hat{M}_{tr}(3,2)}{M_Z}, \frac{\hat{M}_{tr}(3,1)}{M_Z}, \frac{\hat{M}_{tr}(3,3)}{M_Z} \right) \tag{14.257}$$

where:

cdm_X, cdm_Y, and cdm_Z are the center of mass locations with respect to the mass principal axes.

$$\left[\hat{M}_{tr}\right] = [\Phi]^t [M_{tr}][\Phi]$$

If the principal masses are equal, the center of mass location is unique:

$$cdm = \left(\frac{\hat{M}_{tr}(2,3)}{M}, \frac{-\hat{M}_{tr}(1,3)}{M}, \frac{\hat{M}_{tr}(1,2)}{M} \right) \tag{14.258}$$

where:

$M = M_X = M_Y = M_Z$ is the mass of the model.

The mass moments of inertia with respect to the mass principal axes are calculated as:

$$\hat{[I]} = \begin{bmatrix} \hat{M}_r(1,1) - M_Y cdm_Y^2(3) - M_Z cdm_Z^2(2) & -\hat{M}_r(1,2) - M_Z cdm_Z(1) cdm_Z(2) & -\hat{M}_r(1,3) - M_Y cdm_Y(1) cdm_Y(3) \\ & \hat{M}_r(2,2) - M_Z cdm_Z^2(1) - M_X cdm_X^2(3) & -\hat{M}_r(2,3) - M_X cdm_X(2) cdm_X(3) \\ \text{symmetric} & & \hat{M}_r(3,3) - M_X cdm_X^2(2) - M_Y cdm_Y^2(1) \end{bmatrix} \tag{14.259}$$

where:

$$\left[\hat{M}_r\right] = [\Phi]^t [M_r][\Phi]$$

The inertia principal characteristics are obtained from the eigensolution of matrix $[\hat{I}]$. The eigenvalues are the principal moments of inertia and the eigenvectors are the moment of inertia principal directions.

This precise mass calculation is not used for:

- 1-D and 2-D models
- Axiharmonic models. Note that the precise mass summary *is* used for generalized axiharmonic models.
- Models where UX, UY, and UZ degrees of freedom are not present at all element nodes
- Cyclic symmetry models
- Models with acoustic fluid elements (FLUID29, FLUID30, FLUID220, or FLUID221)

These cases use the lumped approximation outlined in Lumped Calculation of Mass Related Information (p. 749).

14.16.2. Lumped Calculation of Mass Related Information

The computation of the mass moments and products of inertia, as well as the model center of mass, is described in this section. This approach assumes that the mass is lumped at the center of each element. The model center of mass is computed as:

$$X_c = \frac{A_x}{M} \tag{14.260}$$

$$Y_c = \frac{A_y}{M} \tag{14.261}$$

$$Z_c = \frac{A_z}{M} \tag{14.262}$$

where typical terms are:

X_c = X coordinate of model center of mass (output as XC)

$$A_x = \sum_{i=1}^{N} m_i X_i$$

$$m_i = \text{mass of element } i = \begin{cases} \text{function of real constants, if applicable} \\ \text{or} \\ \rho V_i \end{cases}$$

ρ = element density, based on average element temperature
V_i = volume of element i

$$X_i = \text{X coordinate of the centroid of element } i = \{N_o\}^T \{X_i\}$$

$\{N_o\}$ = vector of element shape functions, evaluated at the origin of the element coordinate system
$\{X_i\}$ = global X coordinates of the nodes of element i

$$M = \sum_{i=1}^{N} m_i = \text{mass of model (output as TOTAL MASS)}$$

The moments and products of inertia with respect to the origin are:

$$I_{xx} = \sum_{i=1}^{N} m_i((Y_i)^2 + (Z_i)^2) \tag{14.263}$$

$$I_{yy} = \sum_{i=1}^{N} m_i((X_i)^2 + (Z_i)^2) \tag{14.264}$$

$$I_{zz} = \sum_{i=1}^{N} m_i((X_i)^2 + (Y_i)^2) \tag{14.265}$$

$$I_{xy} = -\sum_{i=1}^{N} m_i((X_i)(Y_i)) \tag{14.266}$$

$$I_{yz} = -\sum_{i=1}^{N} m_i((Y_i)(Z_i)) \tag{14.267}$$

$$I_{xz} = -\sum_{i=1}^{N} m_i((X_i)(Z_i)) \tag{14.268}$$

where typical terms are:

I_{xx} = mass moment of inertia about the X axis through the model center of mass (output as IXX)
I_{xy} = mass product of inertia with respect to the X and Y axes through the model center of mass (output as IXY)

Equation 14.263 (p. 749) and Equation 14.265 (p. 749) are adjusted for axisymmetric elements.

The moments and products of inertia with respect to the model center of mass (the components of the inertia tensor) are:

$$I'_{xx} = I_{xx} - M((Y_c)^2 + (Z_c)^2) \tag{14.269}$$

$$I'_{yy} = I_{yy} - M((X_c)^2 + (Z_c)^2) \tag{14.270}$$

$$I'_{zz} = I_{zz} - M((X_c)^2 + (Y_c)^2) \tag{14.271}$$

$$I'_{xy} = I_{xy} + MX_cY_c \tag{14.272}$$

$$I'_{yz} = I_{yz} + MY_cZ_c \tag{14.273}$$

$$I'_{xz} = I_{xz} + MX_cZ_c \tag{14.274}$$

where typical terms are:

I'_{xx} = mass moment of inertia about the X axis through the model center of mass (output as IXX)

I'_{xy} = mass product of inertia with respect to the X and Y axes through the model center of mass (output as IXY)

14.16.2.1. Accuracy of the Lumped Calculation

The above mass calculations are not intended to be precise for all situations, but rather have been programmed for speed. It may be seen from the above development that only the mass (m_i) and the center of mass (X_i, Y_i, and Z_i) of each element are included. Effects that are not considered are:

1. The mass being different in different directions.
2. The presence of rotational inertia terms.
3. The mixture of axisymmetric elements with non-axisymmetric elements (can cause negative moments of inertia).
4. Tapered thicknesses.

5. Offsets used with beams and shells.

6. Trapezoidal-shaped elements.

7. The generalized plane strain option of PLANE182 - 2-D 4-Node Structural Solid (p. 588) and PLANE183 - 2-D 8-Node Structural Solid (p. 589). (When these are present, the center of mass and moment calculations are completely bypassed.)

Thus, if these effects are important, a separate analysis can be performed using inertia relief to find more precise center of mass and moments of inertia (using **IRLF**,-1). Inertia relief logic uses the element mass matrices directly; however, its center of mass calculations also do not include the effects of offsets.

It should be emphasized that the computations for displacements, stresses, reactions, etc. are correct with none of the above approximations.

14.16.2.2. Effect of KSUM, LSUM, ASUM, and VSUM Commands

The center of mass and mass moment of inertia calculations for keypoints, lines, areas, and volumes (accessed by **KSUM**, **LSUM**, **ASUM**, **VSUM**, and ***GET** commands) use equations similar to Equation 14.260 (p. 749) through Equation 14.274 (p. 750) with the following changes:

1. Only selected solid model entities are included.

2. Lines, areas, and volumes are approximated by numerically integrating to account for rotary inertias.

3. Keypoints are assumed to be unit masses without rotary inertia.

4. Lines are assumed to have unit mass per unit length.

5. Each area uses the thickness as:

$$t = \begin{cases} \text{first real constant in the table assigned to the} \\ \text{area (by the } \textbf{AATT} \text{ or } \textbf{AMESH} \text{ command)} \\ \\ \text{1.0 if there is no such assignment or real constant table} \end{cases} \tag{14.275}$$

   where:

   t = thickness

6. Each area or volume is assumed to have density as:

$$\rho = \begin{cases} \text{input density (DENS for the material assigned to the area} \\ \text{or volume (by the } \textbf{AATT/VATT} \text{ or } \textbf{AMESH/VMESH} \text{ command)} \\ \\ \text{1.0 if there is no such assignment or material property} \end{cases} \tag{14.276}$$

   where:

   ρ = density

Composite material elements presume the element material number (defined with the **MAT** command).

14.17. Energies

Energies are available in the solution printout (by setting Item = VENG on the **OUTPR** command) or in postprocessing (by choosing items SENE, TENE, KENE, and AENE on the **ETABLE** command or using the **PRENERGY** command). For each element,

$$E_e^{po} = \begin{cases} \frac{1}{2}\sum_{i=1}^{NINT}\{\sigma\}^T\{\varepsilon^{el}\}vol_i + E_e^{pl} + E_s & \text{if element allows only displacement and rotational degree of freedom (DOF), either is nonlinear or uses integration points, and is not a p-element} \\ \frac{1}{2}\{u_e\}^T([K_e]+[S_e])\{u_e\} & \text{all other cases} \end{cases} \quad (14.277)$$

= potential energy (includes strain energy) (accessed with SENE or TENE on **ETABLE** command)

$$E_e^{ki} = \frac{1}{2}\{\dot{u}_e\}^T[M_e]\{\dot{u}_e\} \quad (14.278)$$

= kinematic energy (accessed with KENE on **ETABLE** command) (computed only for transient and modal analyses)

$$E_e^{art} = \int_{j=1}^{NCS}\frac{1}{2}\{\gamma\}^t[Q]\{\gamma\} \quad (14.279)$$

= artificial energy associated with hourglass control (accessed with AENE on **ETABLE** command) (SOLID45, SOLID182, SOLID185, SHELL181 only)

where:

NINT = number of integration points
$\{\sigma\}$ = stress vector
$\{\varepsilon^{el}\}$ = elastic strain vector
vol_i = volume of integration point i
E_e^{pl} = plastic strain energy
E_s = stress stiffening energy

$$= \begin{cases} \frac{1}{2}\{u_e\}^T[S_e]\{u_e\} & \text{if } [S_e] \text{ is available and } \textbf{NLGEOM},\text{OFF used} \\ 0.0 & \text{all other cases} \end{cases}$$

$[K_e]$ = element stiffness/conductivity matrix
$[S_e]$ = element stress stiffness matrix
$\{u\}$ = element DOF vector
$\{\dot{u}\}$ = time derivative of element DOF vector
$[M_e]$ = element mass matrix
NCS = total number of converged substeps
$\{\gamma\}$ = hourglass strain energy defined in Flanagan and Belytschko([242] (p. 934)) due to one point integrations.

[Q] = hourglass control stiffness defined in Flanagan and Belytschko([242] (p. 934)).

As may be seen from the bottom part of Equation 14.277 (p. 752) as well as Equation 14.278 (p. 752), all types of DOFs are combined, e.g., SOLID5 using both UX, UY, UZ, TEMP, VOLT, and MAG DOF. An exception to this is the piezoelectric elements, described in Piezoelectrics (p. 313), which do report energies by separate types of DOFs in the NMISC record of element results. See Eigenvalue and Eigenvector Extraction (p. 726) when complex frequencies are used. Also, if the bottom part of Equation 14.277 (p. 752) is used, any nonlinearities are ignored. Elements with other incomplete aspects with respect to energy are reported in Table 14.3: Exceptions for Element Energies (p. 753).

Artificial energy has no physical meaning. It is used to control the hourglass mode introduced by reduced integration. The rule-of-thumb to check if the element is stable or not due to the use of reduced integration is if $\frac{\text{AENE}}{\text{SENE}} < 5\%$ is true. When this inequality is true, the element using reduced integration is considered stable (i.e., functions the same way as fully integrated element).

Element type limitations for energy computation are given in Table 14.3: Exceptions for Element Energies (p. 753).

Table 14.3: Exceptions for Element Energies

Element	Exception
FLUID29	No potential energy
FLUID30	No potential energy
LINK31	No potential energy
LINK34	No potential energy
COMBIN39	No potential energy
SHELL41	Foundation stiffness effects not included
SHELL61	Thermal effects not included

1. Warping implies for example that temperatures T1 + T3 ≠ T2 + T4, i.e., some thermal strain is locked in.

14.18. Reduced-Order Modeling for State-Space Matrices Export

The n second order modal equations (Equation 14.118 (p. 700)) are transformed into 2n first order equations, where n is input as *NMODE* on the **SPMWRITE** command, using the following coordinate transformation:

$$\{z\} = \begin{Bmatrix} y \\ \dot{y} \end{Bmatrix} \tag{14.280}$$

The equation becomes:

$$\{\dot{z}\} = [A]\{z\} + [B]\{F\} \tag{14.281}$$

[A] is a (2n x 2n) state-space matrix defined by:

$$[A]=\begin{bmatrix}0 & I\\ \Gamma_1 & \Gamma_2\end{bmatrix}$$

$$[\Gamma_1]=\begin{bmatrix}-\omega_1^2 & 0 & 0 & & \\ 0 & \dots & & & \\ 0 & & -\omega_j^2 & & 0\\ & & & \dots & 0\\ & & 0 & 0 & -\omega_n^2\end{bmatrix} \qquad (14.282)$$

$$[\Gamma_2]=\begin{bmatrix}-2\zeta_1\omega_1 & 0 & 0 & & \\ 0 & \dots & & & \\ 0 & & -2\zeta_j\omega_j & & 0\\ & & & \dots & 0\\ & & 0 & 0 & -2\zeta_n\omega_n\end{bmatrix}$$

Where ω_j is the frequency of mode j, ξ_j is the effective modal damping of mode j (see Modal Damping (p. 701)), and {F} is the vector of input forces:

$$\{F\}=\begin{Bmatrix}F_1(t)\\ \dots\\ F_{ninput}(t)\end{Bmatrix} \qquad (14.283)$$

Where ninput is the number of scalar input forces derived from *Inputs* on the **SPMWRITE** command.

[B] is a (2n x ninput) state-space matrix defined by:

$$[B]=\begin{bmatrix}0\\ \Gamma_3\end{bmatrix} \qquad (14.284)$$

With

$$[\Gamma_3]=[\Phi]^T[F_u] \qquad (14.285)$$

Where $[\Phi]$ is the matrix of eigenvectors and $[F_u]$ is a unit force matrix with size (ndof x ninput). It has 1 at the degrees of freedom where input forces are active and 0 elsewhere.

Now that the states {z} have been expressed as a function of the input loads, the equation for the degrees of freedom observed (outputs w) is written as:

$$\begin{Bmatrix}w\\ \dot{w}\\ \ddot{w}\end{Bmatrix}=[C]\{z\}+[D]\{F\} \qquad (14.286)$$

[C] is a (3*noutput x 2*n) state-space matrix, where noutput is derived from *Outputs* on the **SPMWRITE** command, and is defined by:

$$[C]=\begin{bmatrix}\Gamma_4 & 0\\ 0 & \Gamma_4\\ \Gamma_4\Gamma_1 & \Gamma_4\Gamma_2\end{bmatrix} \qquad (14.287)$$

with

$$[\Gamma_4] = [U_u][\Phi] \tag{14.288}$$

$[U_u]$ is a unit displacement matrix with size (noutput x ndof). It has 1 on degrees of freedom where output is requested and 0 elsewhere.

[D] is a (3*noutput x ninput) state-space matrix defined by:

$$[D] = \begin{bmatrix} 0 \\ 0 \\ \Gamma_4\Gamma_3 \end{bmatrix} \tag{14.289}$$

$\dot{w}$ and $\ddot{w}$ are included only if VelAccKey = ON on the **SPMWRITE** command, otherwise the last two rows of [C] are not written and [D] is zero so it is not written.

14.19. Enforced Motion in Structural Analysis

In structural analysis, enforced motion is a common excitation. Examples of this behavior include the response of a building to an earthquake, the vibration of a device carried by a vehicle, etc.

The equations of motion in terms of absolute displacements can be expressed by:

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F^a\} \tag{14.290}$$

where:

$[M]$ = structural mass matrix

$[C]$ = structural damping matrix

$[K]$ = structural stiffness matrix

$\{\ddot{u}\}$ = nodal acceleration vector

$\{\dot{u}\}$ = nodal velocity vector

$\{u\}$ = nodal displacement vector

$\{F^a\}$ = applied load vector

Partition the degrees of freedom into two sets:

$$\{u\} = \begin{Bmatrix} \{u_1\} \\ \{u_2\} \end{Bmatrix} \tag{14.291}$$

where:

$\{u_1\}$ = displacements remaining free

$\{u_2\}$ = displacements with enforced motion

Assuming that the only excitation source is the enforced motion, the load vector applied on degree of freedom $\{u_1\}$ is zero. Equation 14.290 (p. 755) can be expanded to:

$$\begin{bmatrix}[M_{11}] & [M_{12}]\\ [M_{21}] & [M_{22}]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}_1\}\\ \{\ddot{u}_2\}\end{Bmatrix}+\begin{bmatrix}[C_{11}] & [C_{12}]\\ [C_{21}] & [C_{22}]\end{bmatrix}\begin{Bmatrix}\{\dot{u}_1\}\\ \{\dot{u}_2\}\end{Bmatrix}+\begin{bmatrix}[K_{11}] & [K_{12}]\\ [K_{21}] & [K_{22}]\end{bmatrix}\begin{Bmatrix}\{u_1\}\\ \{u_2\}\end{Bmatrix}=\begin{Bmatrix}\{0\}\\ \{F_2\}\end{Bmatrix} \tag{14.292}$$

Where $\{F_2\}$ = the reaction force between the structure and its supports.

14.19.1. Full Method for Transient and Harmonic Analyses

In transient and harmonic analyses using the full method, the upper part of Equation 14.292 (p. 756) is rearranged and solved as:

$$[M_{11}]\{\ddot{u}_1\}+[C_{11}]\{\dot{u}_1\}+[K_{11}]\{u_1\}=-[M_{12}]\{\ddot{u}_2\}-[C_{12}]\{\dot{u}_2\}-[K_{12}]\{u_2\} \tag{14.293}$$

where

$\{\ddot{u}_2\}$ is the enforced acceleration. In a full transient analysis, it is input with the **D** command and *Lab* = ACCX, ACCY, ACCZ, DMGX, DMGY, or DMGZ (not supported in harmonic).

$\{\dot{u}_2\}$ is the enforced velocity. In a full transient analysis, it is input with the **D** command and *Lab* = VELX, VELY, VELZ, OMGX, OMGY, or OMGZ (not supported in harmonic).

$\{u_2\}$ is the enforced displacement. In full transient and harmonic analyses, it is input with the **D** command and *Lab* = UX, UY, UZ, ROTX, ROTY, or ROTZ.

The solution of Equation 14.293 (p. 756) is the absolute displacement vector $\{u_1\}$.

14.19.2. Enforced Motion Method for Transient and Harmonic Analyses

The nodal displacement due to enforced motion can be separated into the quasi-static response and the dynamic response, also called relative motion (see Paultre [421] (p. 944)):

$$\{u\}=\begin{Bmatrix}\{u_1\}\\ \{u_2\}\end{Bmatrix}=\begin{Bmatrix}\{u_1^{qs}\}\\ \{u_2\}\end{Bmatrix}+\begin{Bmatrix}\{y\}\\ \{0\}\end{Bmatrix} \tag{14.294}$$

where:

{y} is the dynamic response, representing the relative motion of the structure with respect to the base motion.

$\{u_1^{qs}\}$ is the quasi-static response.

By introducing Equation 14.294 (p. 756) into Equation 14.292 (p. 756), and neglecting time-derivative terms, the following equation is obtained:

$$\begin{bmatrix} [K_{11}] & [K_{12}] \\ [K_{21}] & [K_{22}] \end{bmatrix} \begin{Bmatrix} \{u_1^{qs}\} \\ \{u_2\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{F_2\} \end{Bmatrix} \tag{14.295}$$

From the upper part of Equation 14.295 (p. 757), $\{u_1^{qs}\}$ is derived as:

$$\{u_1^{qs}\} = -[K_{11}]^{-1}[K_{12}]\{u_2\} \tag{14.296}$$

The absolute displacement can be written as:

$$\{u_1\} = \{u_1^{qs}\} + \{y\} = -[K_{11}]^{-1}[K_{12}]\{u_2\} + \{y\} \tag{14.297}$$

Substituting Equation 14.296 (p. 757) and Equation 14.297 (p. 757) into Equation 14.293 (p. 756) gives:

$$[M_{11}]\{\ddot{y}\} + [C_{11}]\{\dot{y}\} + [K_{11}]\{y\} = \left([M_{11}][K_{11}]^{-1}[K_{12}] - [M_{12}]\right)\{\ddot{u}_2\} + \left([C_{11}][K_{11}]^{-1}[K_{12}] - [C_{12}]\right)\{\dot{u}_2\} \tag{14.298}$$

Neglecting the enforced velocity term on the right hand side, the equation reduces to:

$$[M_{11}]\{\ddot{y}\} + [C_{11}]\{\dot{y}\} + [K_{11}]\{y\} = \left([M_{11}][K_{11}]^{-1}[K_{12}] - [M_{12}]\right)\{\ddot{u}_2\} \tag{14.299}$$

This equation is exact in the case of stiffness-based proportional damping (**BETAD**) and results will match those obtained with a full method using Equation 14.293 (p. 756). It is approximate for all other cases of damping.

14.19.2.1. Structure Subjected to Differential Support Motion

When a structure is subjected to excitations from different supports and/or along different axes, Equation 14.299 (p. 757) can be solved using the mode superposition method. In that case, the degrees of freedom $\{u_2\}$ are first fixed for the modal analysis. The natural frequencies and mode shapes of the conservative system without damping, derived from Equation 14.299 (p. 757) are used to obtain the uncoupled equations, as explained in Mode Superposition Method (p. 698) in this guide.

If different supports are excited, the vector of displacements $\{u_2\}$ from enforced motion and the associated accelerations $\{\ddot{u}_2\}$ are composed of subsets with degrees of freedom corresponding to the different supports. Each subset, also called base, is identified using the *Value* argument on the **D** command during the modal analysis, and for all the degrees of freedom of a subset, the common value of enforced displacement $\{u_2\}^s$ or enforced acceleration $\{\ddot{u}_2\}^s$ is input using the **DVAL** command in the mode superposition transient and harmonic analyses.

In a harmonic analysis, depending on whether displacements or accelerations are input with the **DVAL** command, the following relationship is used internally:

$$\{\ddot{u}_2\}^s = -\Omega^2 \{u_2\}^s$$

where Ω is the current forced circular frequency.

This equation defines the acceleration of supports in Equation 14.299 (p. 757) when displacements are specified and defines displacements of supports in Equation 14.296 (p. 757) when accelerations are specified.

In a transient analysis, the Newmark time integration method described in Transient Analysis (p. 763) is internally used to calculate the enforced displacements when enforced accelerations are specified. Conversely, the same method is used to calculate the enforced accelerations when enforced displacements are specified.

When *KeyCal* is ON in the **DVAL** command, the final displacement vector $\{u_1\}$ is calculated with Equation 14.294 (p. 756) and displacements are absolute.

When *KeyCal* is OFF in the **DVAL** command, the final displacement vector is {y} and displacements are relative.

If excitations act in different directions and/or intensity, stresses from the quasi-static solution are not zero. Therefore, stresses obtained with *KeyCal* set to ON are different from those obtained with *KeyCal* set to OFF.

For more information, see Enforced Motion Method for Mode-Superposition Transient and Harmonic Analyses in the *Structural Analysis Guide*.

14.19.2.2. Structure Subjected to Global Support Acceleration

When a structure is subjected to global motion of a rigid support (see Geradin and Rixen [368] (p. 941)), reaction force vector $\{F_2\}$ is zero and the quasi-static response defined in Equation 14.296 (p. 757) can be rewritten as:

$$\left\{u_1^{qs}\right\} = -\left[K_{11}\right]^{-1}\left[K_{12}\right]\left\{u_2\right\} = \left[r_1\right]\left\{u_2\right\} \tag{14.300}$$

where:

$[r_1]$ represents the rigid-body modes of the part of the structure remaining free

$\{u_2\}$ = the support motion, the acceleration of which is:

$$\left\{\ddot{u}_2\right\} = \left\{I\right\}\gamma \tag{14.301}$$

where:

γ = the global acceleration of the support, input on the **ACEL** command.

In this case, the coupled mass between the free and constrained degrees of freedom ($[M_{12}]$) is ignored in Equation 14.299 (p. 757) to obtain:

$$\left[M_{11}\right]\left\{\ddot{y}\right\} + \left[C_{11}\right]\left\{\dot{y}\right\} + \left[K_{11}\right]\left\{y\right\} = -\left[M_{11}\right]\left[r_1\right]\left\{I\right\}\gamma \tag{14.302}$$

To solve Equation 14.302 (p. 758), degrees of freedom $\{u_2\}$ must be fixed and the full method transient and harmonic analyses are used. The solution is the relative displacement vector {y}.

As stresses of the quasi-static solution are zero, the stresses obtained with a full method working with absolute displacements (**D**) or relative displacements (**ACEL**) are identical.

14.19.3. Large Mass Method

The large mass method is an approximate technique that treats the response to acceleration excitation as a response to external forces.

Assuming that the masses associated with subsystem are augmented so that $[M_{22}]$ becomes $[M_{22}+M_{22}^{large}]$, then Equation 14.292 (p. 756) becomes:

$$\begin{bmatrix}[M_{11}] & [M_{12}]\\ [M_{21}] & \left[M_{22}+M_{22}^{large}\right]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}_1\}\\ \{\ddot{u}_2\}\end{Bmatrix}+\begin{bmatrix}[C_{11}] & [C_{12}]\\ [C_{21}] & [C_{22}]\end{bmatrix}\begin{Bmatrix}\{\dot{u}_1\}\\ \{\dot{u}_2\}\end{Bmatrix}+\begin{bmatrix}[K_{11}] & [K_{12}]\\ [K_{21}] & [K_{22}]\end{bmatrix}\begin{Bmatrix}\{u_1\}\\ \{u_2\}\end{Bmatrix}=\begin{Bmatrix}\{0\}\\ \{F_2\}\end{Bmatrix} \tag{14.303}$$

Solve the equations of $\{u_2\}$:

$$\{\ddot{u}_2\}=\left[M_{22}+M_{22}^{large}\right]^{-1}\left\{\{F_2\}-[K_{22}]\{u_2\}-[K_{21}]\{u_1\}-[C_{22}]\{\dot{u}_2\}-[C_{21}]\{\dot{u}_1\}-[M_{21}]\{\ddot{u}_1\}\right\} \tag{14.304}$$

Define $\left[\bar{M}\right]=\left[M_{22}+M_{22}^{large}\right]^{-1}$ and substitute Equation 14.304 (p. 759) into the upper part of Equation 14.303 (p. 759) to get:

$$\left([M_{11}]-[M_{12}]\left[\bar{M}\right][M_{21}]\right)\{\ddot{u}_1\}+\left([C_{11}]-[M_{12}]\left[\bar{M}\right][C_{21}]\right)\{\dot{u}_1\}+\left([K_{11}]-[M_{12}]\left[\bar{M}\right][K_{21}]\right)\{u_1\}=\left\{\bar{F}\right\} \tag{14.305}$$

where:

$$\left\{\bar{F}\right\}=-[K_{12}]\{u_2\}-[C_{12}]\{\dot{u}_2\}-[M_{12}]\left[\bar{M}\right]\left(\{F_2\}-[K_{22}]\{u_2\}-[C_{22}]\{\dot{u}_2\}\right) \tag{14.306}$$

If $\left[M_{22}^{large}\right]$ is large enough, $\left[\bar{M}\right]$ tends to zero, and the following approximation is verified:

$$[F_2]=\left[M_{22}+M_{22}^{large}\right]\{\ddot{u}_2\}\approx\left[M_{22}^{large}\right]\{\ddot{u}_2\} \tag{14.307}$$

Equation 14.305 (p. 759) will be in the same form as Equation 14.293 (p. 756), and Equation 14.303 (p. 759) is equivalent to Equation 14.292 (p. 756).

A ratio of the large mass to the mass of the entire structure in the range of 10^4 to 10^8 keeps the modeling error small (see Léger et al. [422] (p. 944))

Large lumped masses can be implemented by using element type MASS21.

The solution of Equation 14.305 (p. 759) is the absolute displacement vector $\{u_1\}$.

Chapter 15: Analysis Procedures

This chapter presents the theoretical basis of the various analysis procedures. The derivation of the individual element matrices and load vectors is discussed in Derivation of Structural Matrices (p. 12), Derivation of Electromagnetic Matrices (p. 193), Derivation of Heat Flow Matrices (p. 235), and Derivation of Acoustic Matrices (p. 256).

In the matrix displacement method of analysis based upon finite element idealization, the structure being analyzed must be approximated as an assembly of discrete regions (called elements) connected at a finite number of points (called nodes). If the "force-displacement" relationship for each of these discrete structural elements is known (the element "stiffness" matrix) then the "force-displacement relationship" for the entire "structure" can be assembled using standard matrix methods. These methods are well documented (see, for example, Zienkiewicz([39] (p. 922))) and are also discussed in Analysis Tools (p. 665). Thermal, fluid flow, and electromagnetic analyses are done on an analogous basis by replacing the above words in quotes with the appropriate terms. However, the terms displacement, force, and stiffness are used frequently throughout this chapter, even though it is understood that the concepts apply to all valid effects also.

All analysis types for iterative or transient problems automatically reuse the element matrices or the overall structural matrix whenever it is applicable. See Reuse of Matrices (p. 392) for more details.

Analysis procedure information is available for the following analysis types:

15.1. Static Analysis

The following static analysis topics are available:

15.1.1. Assumptions and Restrictions

The static analysis (**ANTYPE**,STATIC) solution method is valid for all degrees of freedom (DOFs). Inertial and damping effects are ignored, except for static acceleration fields.

15.1.2. Description of Structural Systems

The overall equilibrium equations for linear structural static analysis are:

$$[K]\{u\} = \{F\} \tag{15.1}$$

or

$$[K]\{u\} = \{F^a\} + \{F^r\} \tag{15.2}$$

where:

$[K]$ = total stiffness matrix = $\sum_{m=1}^{N} [K_e]$

$\{u\}$ = nodal displacement vector

N = number of elements

$[K_e]$ = element stiffness matrix (described in Element Library (p. 411)) (may include the element stress stiffness matrix (described in Stress Stiffening (p. 41)))

$\{F^r\}$ = reaction load vector

$\{F^a\}$, the total applied load vector, is defined by:

$$\{F^a\} = \{F^{nd}\} + \{F^{ac}\} + \sum_{m=1}^{N} (\{F_e^{th}\} + \{F_e^{pr}\}) \tag{15.3}$$

where:

$\{F^{nd}\}$ = applied nodal load vector

$\{F^{ac}\}$ = - $[M]$ $\{a_c\}$ = acceleration load vector

$[M]$ = total mass matrix = $\sum_{m=1}^{N} [M_e]$

$[M_e]$ = element mass matrix (described in Derivation of Structural Matrices (p. 12))

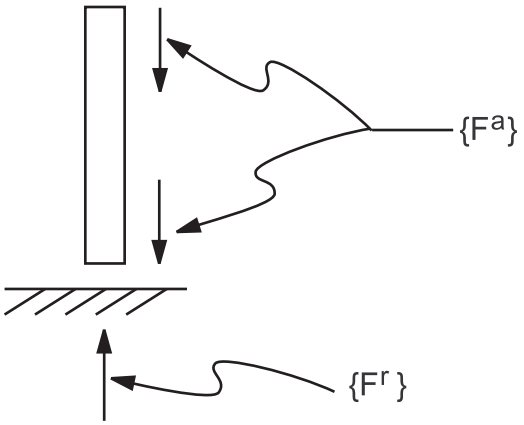
$\{a_c\}$ = total acceleration vector (defined in Acceleration Effect (p. 665))

$\{F_e^{th}\}$ = element thermal load vector (described in Derivation of Structural Matrices (p. 12))

$\{F_e^{pr}\}$ = element pressure load vector (described in Derivation of Structural Matrices (p. 12))

To illustrate the load vectors in Equation 15.2 (p. 762), consider a one element column model, loaded only by its own weight, as shown in Figure 15.1: Applied and Reaction Load Vectors (p. 763). Note that the lower applied gravity load is applied directly to the imposed displacement, and therefore causes no strain; nevertheless, it contributes to the reaction load vector just as much as the upper applied gravity load. Also, if the stiffness for a certain DOF is zero, any applied loads on that DOF are ignored.

Figure 15.1: Applied and Reaction Load Vectors



Solving for Unknowns and Reactions (p. 690) discusses the solution of Equation 15.2 (p. 762) and the computation of the reaction loads. Newton-Raphson Procedure (p. 711) describes the global equation for a nonlinear analysis. Inertia relief is discussed in Inertia Relief (p. 669).

15.1.3. Description of Thermal, Magnetic and Other First Order Systems

The overall equations for linear 1st order systems are the same as for a linear structural static analysis, Equation 15.1 (p. 762) and Equation 15.2 (p. 762). [K], though, is the total coefficient matrix (e.g., the conductivity matrix in a thermal analysis) and {u} is the nodal DOF values. $\{F^a\}$, the total applied load vector, is defined by:

$$\{Q^a\} = \{Q^{nd}\} + \sum_{m=1}^{N} \{Q_e\} \tag{15.4}$$

Table 15.1: Nomenclature (p. 763) relates the nomenclature used in Derivation of Heat Flow Matrices (p. 235) and Derivation of Electromagnetic Matrices (p. 193) for thermal, magnetic and electrical analyses to Equation 15.2 (p. 762) and Equation 15.4 (p. 763). See Table 10.3: Nomenclature of Coefficient Matrices (p. 306) for a more detailed nomenclature description.

Table 15.1: Nomenclature

	{u}	$\{F^{nd}\}$	$\{F_e\}$
Thermal	{T} temperature	$\{Q^{nd}\}$ heat flow	$\{Q_e\} + \{Q_e^g\} + \{Q_e^c\}$ heat flux heat generation convection
Scalar Magnetic	{ϕ} scalar potential	$\{F^{nd}\}$ flux	$\{F_e\}$ coercive force
Vector Magnetic	{A} vector potential	$\{F^{nd}\}$ current segment	$\{F_e\}$ current density and coercive force
Electrical	{V} voltage	$\{I^{nd}\}$ current	-

Solving for Unknowns and Reactions (p. 690) discusses the solution of Equation 15.2 (p. 762) and Newton-Raphson Procedure (p. 711) describes the global equation for a nonlinear analysis.

15.2. Transient Analysis

The following transient analysis topics are available:

The transient analysis solution method (**ANTYPE**,TRANS) used depends on the DOFs involved. Structural, acoustic, and other second order systems (that is, the systems are second order in time) are solved using one method and the thermal, magnetic, electrical and other first order systems are solved using another. Each method is described subsequently. If the analysis contains both first and second order DOFs (e.g., structural and magnetic), then each DOF is solved using the appropriate method. For matrix coupling between first and second order effects such as for piezoelectric analysis, a combined procedure is used.

15.2.1. Assumptions and Restrictions

1. Initial conditions are known.

2. Gyroscopic or Coriolis effects are included in a structural analysis when requested (using the **CORIOLIS** command).

15.2.2. Description of Structural and Other Second Order Systems

For most structural dynamics problems of a mechanical system, the spatial discretization for the principle of virtual work using the finite element method gives the finite element semi-discrete equation of motion as follows:

$$[M]\{\ddot{u}(t)\} + [C]\{\dot{u}(t)\} + \{F^i(t)\} = \{F^a(t)\} \tag{15.5}$$

where:

[M] = structural mass matrix
[C] = structural damping matrix
$\{\ddot{u}(t)\}$ = nodal acceleration vector
$\{\dot{u}(t)\}$ = nodal velocity vector
{u(t)} = nodal displacement vector
$\{F^i(t)\}$ = internal load vector
$\{F^a(t)\}$ = applied load vector

Three methods are available for solving Equation 15.5 (p. 764):

- *Central difference time integration method* -- Used for explicit transient analyses only and described in the *LS-DYNA Theoretical Manual*([199] (p. 932)).

- *Newmark time integration method* -- Used for implicit transient analyses as described below. This method is requested by setting *TINTOPT* = NMK (which is the default) on the **TRNOPT** command.

- *HHT time integration method* -- Used also for implicit transient analyses as described below. This method is an extension of the Newmark time integration method and is requested by setting *TINTOPT* = HHT on the **TRNOPT** command.

The structural dynamics problems concerned with the mechanical behavior governed by the above differential equation can be classified into two classes; that is, linear and nonlinear problems.

15.2.2.1. Time Integration Scheme for Linear Systems

In linear structural dynamics systems, the internal load is linearly proportional to the nodal displacement, and the structural stiffness matrix remains constant. Therefore, Equation 15.5 (p. 764) can be rewritten as:

$$[M]\{\ddot{u}(t)\} + [C]\{\dot{u}(t)\} + [K]\{u(t)\} = \{F^a(t)\} \tag{15.6}$$

where:

[K] = structural stiffness matrix

Among direct time integration methods for numerically solving the finite element semi-discrete equation of motion given in Equation 15.6 (p. 765), several methods such as the Newmark method (Newmark([405] (p. 943))) and the generalized-α method (Chung and Hulbert([351] (p. 940))) are incorporated in the program. As the , ill generalized-α method recovers the Wood-Bosak-Zienkiewicz method (also called WBZ-α method) (Wood et al.([353] (p. 940))), the Hilber-Hughes-Taylor method (also called HHT-α method) (Hilber et al.([352] (p. 940))), and the Newmark family of time integration algorithms, the program allows a user to take advantage of any of the these methods.

Newmark Method

The Newmark family of time integration algorithms (Newmark([405] (p. 943))) is one of the most popular time integration methods as a single step algorithm. The semi-discrete equation of motion given in Equation 15.6 (p. 765) can be rewritten as (Hughes([165] (p. 930))):

$$[M]\{\ddot{u}_{n+1}\} + [C]\{\dot{u}_{n+1}\} + [K]\{u_{n+1}\} = \{F^a_{n+1}\} \tag{15.7}$$

where:

$\{\ddot{u}_{n+1}\}$ = the nodal acceleration vector $\{\ddot{u}(t_{n+1})\}$ at time t_{n+1}

$\{\dot{u}_{n+1}\}$ = the nodal velocity vector $\{\dot{u}(t_{n+1})\}$ at time t_{n+1}

$\{u_{n+1}\}$ = the nodal displacement vector $\{u(t_{n+1})\}$ at time t_{n+1}

$\{F^a_{n+1}\}$ = the applied load $\{F^a_{n+1}(t_{n+1})\}$ at time t_{n+1}

In addition to Equation 15.7 (p. 765), the Newmark family of time integration algorithms requires the displacement and velocity to be updated as follows:

$$\{\dot{u}_{n+1}\} = \{\dot{u}_n\} + [(1-\delta)\{\ddot{u}_n\} + \delta\{\ddot{u}_{n+1}\}]\Delta t \tag{15.8}$$

$$\{u_{n+1}\} = \{u_n\} + \{\dot{u}_n\}\Delta t + \left[\left(\frac{1}{2} - \alpha\right)\{\ddot{u}_n\} + \alpha\{\ddot{u}_{n+1}\}\right]\Delta t^2 \tag{15.9}$$

where:

α, δ = Newmark's integration parameters

$\{\ddot{u}_n\}$ = nodal acceleration vector $\{\ddot{u}(t_n)\}$ at time t_n

$\{\dot{u}_n\}$ = nodal velocity vector $\{\dot{u}(t_n)\}$ at time t_n

$\{u_n\}$ = nodal displacement vector $\{u(t_n)\}$ at time t_n

Thus, the Newmark family of time integration algorithms can be determined by the Newmark integration parameters. In the end, the Newmark integration scheme consists of the three finite difference equations presented in Equation 15.7 (p. 765) through Equation 15.9 (p. 765), and the three unknowns $\{\ddot{u}_{n+1}\}$, $\{\dot{u}_{n+1}\}$, and $\{u_{n+1}\}$ can be numerically calculated by the three algebraic equations along with the three known quantities $\{\ddot{u}_n\}$, $\{\dot{u}_n\}$, and $\{u_n\}$.

By making use of the three algebraic equations given in Equation 15.7 (p. 765) through Equation 15.9 (p. 765), a single-step time integrator in terms of the unknown $\{u_{n+1}\}$ and the three known quantities can be written as:

$$(a_0[M] + a_1[C] + [K])\{u_{n+1}\} = \{F^a_{n+1}\} + [M](a_0\{u_n\} + a_2\{\dot{u}_n\} + a_3\{\ddot{u}_n\}) + [C](a_1\{u_n\} + a_4\{\dot{u}_n\} + a_5\{\ddot{u}_n\}) \tag{15.10}$$

where:

$$a_0 = \frac{1}{\alpha\Delta t^2} \qquad a_1 = \frac{\delta}{\alpha\Delta t}$$

$$a_2 = \frac{1}{\alpha\Delta t} \qquad a_3 = \frac{1}{2\alpha} - 1$$

$$a_4 = \frac{\delta}{\alpha} - 1 \qquad a_5 = \frac{\Delta t}{2}\left(\frac{\delta}{\alpha} - 2\right)$$

$$a_6 = \Delta t(1-\delta) \qquad a_7 = \delta\Delta t$$

First, the unknown $\{u_{n+1}\}$ is calculated using Equation 15.10 (p. 766)). Then, the program computes the two unknowns $\{\dot{u}_{n+1}\}$ and $\{\ddot{u}_{n+1}\}$ by using the following equations:

$$\{\dot{u}_{n+1}\} = a_1(\{u_{n+1}\} - \{u_n\}) - a_4\{\dot{u}_n\} - a_5\{\ddot{u}_n\} \tag{15.11}$$

$$\{\ddot{u}_{n+1}\} = a_0(\{u_{n+1}\} - \{u_n\}) - a_2\{\dot{u}_n\} - a_3\{\ddot{u}_n\} \tag{15.12}$$

The most important factors in choosing an appropriate time integration scheme for the finite element semi-discrete equation of motion given in Equation 15.5 (p. 764) are accuracy, stability, and dissipation. In conditionally stable time integration algorithms, stability is affected by a chosen size of the time step; whereas in unconditionally stable time integration algorithms, a time step size can be chosen independent of stability considerations.

In the Newmark method, the amount of numerical algorithm dissipation can be controlled by one of Newmark's parameters, δ, as follows:

$$\delta \geq \frac{1}{2}\ , \qquad \alpha \geq \frac{1}{4}\left(\frac{1}{2} + \delta\right)^2 \tag{15.13}$$

With the Newmark parameters satisfying the above conditions, the Newmark family of methods may be unconditionally stable (Hughes([165] (p. 930))). By introducing the amplitude decay factor $\gamma \geq 0$, the above conditions can be written:

$$\delta = \frac{1}{2} + \gamma, \qquad \alpha = \frac{1}{4}(1+\gamma)^2, \qquad \gamma \geq 0 \tag{15.14}$$

Consequently, the program provides the user with the Newmark integration procedure, which is unconditionally stable via input of the amplitude decay factor γ on the **TINTP** command. Alternatively, the α and δ parameters may be input directly using the **TINTP** command.

Generalized HHT-α Method

In the Newmark method, the amount of numerical dissipation can be controlled by one parameter δ in Equation 15.13 (p. 766) or γ in Equation 15.14 (p. 766). However, in low frequency modes the Newmark method fails to retain the second-order accuracy as $\delta > \frac{1}{2}$. Note that the Newmark implicit method (constant average method; namely, $\delta = \frac{1}{2}$ and $\alpha = \frac{1}{4}$), which is unconditionally stable and second-order accurate, has no numerical damping. If other sources of numerical damping are not introduced, the lack of numerical damping can be undesirable so that the higher frequencies of the structure can produce unacceptable levels of numerical noise (Hughes([165] (p. 930))).

To circumvent the drawbacks of the Newmark family of methods, the program implements the generalized HHT-α method which sufficiently damps out spurious high-frequency response via introducing controllable numerical dissipation in higher frequency modes, while maintaining the second-order accuracy. It should be noted that the generalized HHT-α method incorporated in the program is capable of recovering the WBZ-α method (Wood et al.([353] (p. 940))) and the HHT-α method (Hilber et al.([352] (p. 940))) as well as the Newmark family of time integration algorithms, depending upon the user's input on the **TINTP** command.

To solve for the three unknowns $\{\ddot{u}_{n+1}\}$, $\{\dot{u}_{n+1}\}$, and $\{u_{n+1}\}$, along with Equation 15.8 (p. 765) and Equation 15.9 (p. 765) the generalized HHT-α method uses the algebraic equation:

$$[M]\{\ddot{u}_{n+1-\alpha_m}\} + [C]\{\dot{u}_{n+1-\alpha_f}\} + [K]\{u_{n+1-\alpha_f}\} = \{F^a(t_{n+1-\alpha_f})\} \tag{15.15}$$

where:

$$\{\ddot{u}_{n+1-\alpha_m}\} = (1-\alpha_m)\{\ddot{u}_{n+1}\} + \alpha_m\{\ddot{u}_n\}$$

$$\{\dot{u}_{n+1-\alpha_f}\} = (1-\alpha_f)\{\dot{u}_{n+1}\} + \alpha_f\{\dot{u}_n\}$$

$$\{u_{n+1-\alpha_f}\} = (1-\alpha_f)\{u_{n+1}\} + \alpha_f\{u_n\}$$

$$\{F^a(t_{n+1-\alpha_f})\} = (1-\alpha_f)\{F^a_{n+1}\} + \alpha_f\{F^a_n\}$$

Equation 15.15 (p. 767) give the finite difference form:

$$\begin{aligned}(a_0[M] + a_1[C] + (1-\alpha_f)[K])\{u_{n+1}\} &= (1-\alpha_f)\{F^a_{n+1}\} + \alpha_f\{F^a_n\} - \alpha_f[K]\{u_n\} + \\ [M](a_0\{u_n\} + a_2\{\dot{u}_n\} + a_3\{\ddot{u}_n\}) &+ [C](a_1\{u_n\} + a_4\{\dot{u}_n\} + a_5\{\ddot{u}_n\})\end{aligned} \tag{15.16}$$

where:

$$a_0 = \frac{1-\alpha_m}{\alpha\Delta t^2}$$

$$a_1 = \frac{(1-\alpha_f)\delta}{\alpha\Delta t}$$

$$a_2 = a_0 \Delta t$$

$$a_3 = \frac{1-\alpha_m}{2\alpha} - 1$$

$$a_4 = \frac{(1-\alpha_f)\delta}{\alpha} - 1$$

$$a_5 = (1-\alpha_f)(\frac{\delta}{2\alpha} - 1)\Delta t$$

Analogous to the Newmark method, the generalized HHT-α method calculates the unknown $\{u_{n+1}\}$ at time t_{n+1} by making use of Equation 15.16 (p. 767). Then, the program computes the two unknowns $\{\dot{u}_{n+1}\}$ and $\{\ddot{u}_{n+1}\}$ by using the equations given in Equation 15.11 (p. 766) and Equation 15.12 (p. 766). Since the generalized HHT-α method is also an implicit time scheme, the structural stiffness matrix must be factorized to solve for $\{u_{n+1}\}$ at time t_{n+1}.

As mentioned in the literature (Chung and Hulbert([351] (p. 940))), the generalized HHT-α method is unconditionally stable and second-order accurate if the parameters meet the following conditions:

$$\begin{aligned} &\delta = \frac{1}{2} - \alpha_m + \alpha_f \\ &\alpha \geq \frac{1}{2}\delta \\ &\alpha_m \leq \alpha_f \leq \frac{1}{2} \end{aligned} \tag{15.17}$$

where $\alpha_m \leq 0$ (Wood et al.([353] (p. 940))) and $0 \leq \alpha_f \leq \frac{1}{3}$ (Hilber et al.([352] (p. 940))). For the generalized HHT-α method, the user can input the four parameters on the **TINTP** command. By introducing the amplitude decay factor $\gamma \geq 0$, the program also allows the user to control the amount of numerical damping if the four parameters on the **TINTP** command meet the following conditions:

$$\begin{aligned} &\alpha = \frac{1}{4}(1+\gamma)^2 \\ &\delta = \frac{1}{2} + \gamma \\ &\alpha_f = \gamma \geq 0 \\ &\alpha_m = 0 \end{aligned} \tag{15.18}$$

If the WBZ-α method is desired, the user can control the amount of numerical damping if the four parameters on the **TINTP** command meet the following conditions:

$$\begin{aligned} &\alpha = \frac{1}{4}(1+\gamma)^2 \\ &\delta = \frac{1}{2} + \gamma \\ &\alpha_f = 0 \\ &\alpha_m = -\gamma \leq 0 \end{aligned} \tag{15.19}$$

Finally, the program also allows a user who wants to use the generalized HHT-α method to control the amount of numerical damping if the four parameters on the **TINTP** command meet the following conditions:

$$\alpha = \frac{1}{4}(1+\gamma)^2$$
$$\delta = \frac{1}{2} + \gamma$$
$$\alpha_f = \frac{1-\gamma}{2}$$
$$\alpha_m = \frac{1-3\gamma}{2} \tag{15.20}$$

It should be noted that the generalized HHT-α method is second-order accurate and unconditionally stable. This method allows you to control the amount of numerical damping. The amplitude decay factor is recommended to be set as γ = 0.05 (Hughes([165] (p. 930))), with which any spurious participation of the higher modes can be damped out and the lower modes are not affected. A significant amount of numerical damping may be introduced by setting $\gamma = \frac{1}{3}$, but it is not recommended.

15.2.2.2. Time Integration Scheme for Nonlinear Systems

In nonlinear structural dynamics problems, the internal load is no longer linearly proportional to the nodal displacement, and the structural stiffness matrix is dependent on the current displacement. Therefore, Instead of Equation 15.6 (p. 765), any time integration scheme should be applied to the nonlinear semi-discrete equation:

$$[M]\{\ddot{u}(t)\} + [C]\{\dot{u}(t)\} + \{F^i(t)\} = \{F^a(t)\} \tag{15.21}$$

Equation 15.21 (p. 769) represents a nonlinear system of simultaneous algebraic equations; hence, any time integration operator may be used in association with the Newton-Raphson iterative algorithm. For nonlinear structural dynamics problems, both the Newmark method and the generalized HHT-α method are incorporated in the program.

Newmark Method

The Newmark method assumes that at the time t_{n+1}, the semi-discrete equation of motion given in Equation 15.21 (p. 769) can be rewritten as:

$$[M]\{\ddot{u}_{n+1}\} + [C]\{\dot{u}_{n+1}\} + \{F^i_{n+1}(\{u_{n+1}\})\} = \{F^a_{n+1}\} \tag{15.22}$$

Note that $\{F^i_{n+1}(\{u_{n+1}\})\}$ is dependent on the current displacement $\{u_{n+1}\}$ at time t_{n+1}. In addition to Equation 15.22 (p. 769), the Newmark family of time integration algorithms requires the displacement and velocity to be updated as given in Equation 15.8 (p. 765) and Equation 15.9 (p. 765).

By introducing the residual vector $\{R_{n+1}(\{u_{n+1}\})\}$, Equation 15.22 (p. 769) can be written as:

$$\{R_{n+1}(\{u_{n+1}\})\} = \{F^a_{n+1}\} - \{F^i_{n+1}(\{u_{n+1}\})\} - [M]\{\ddot{u}_{n+1}\} - [C]\{\dot{u}_{n+1}\} \tag{15.23}$$

It is important to note that the time integration operator given in either Equation 15.22 (p. 769) or Equation 15.23 (p. 769) represents a nonlinear system of simultaneous algebraic equations. Therefore, a

linearized form of the time integration operator can be obtained by the Newton-Raphson method as follows:

$$\{R_{n+1}(\{u^k_{n+1}\})\} + \frac{\partial\{R_{n+1}(u^k_{n+1})\}}{\partial\{u^i_{n+1}\}}\{\Delta u^k_{n+1}\} = \{0\} \tag{15.24}$$

where:

$\{u^k_{n+1}\}$ = the estimate of $\{u_{n+1}\}$ at the k^{th} iteration

$\{\Delta u^k_{n+1}\}$ = the displacement increment of $\{u_{n+1}\}$ at the k^{th} iteration

$R_{n+1}(\{u^k_{n+1}\}) = \{F^a_{n+1}\} - \{F^i_{n+1}(\{u^k_{n+1}\})\} - [M]\{\ddot{u}_{n+1}\} - [C]\{\dot{u}_{n+1}\}$

Equation 15.24 (p. 770) gives:

$$\left[(a_0[M] + a_1[C]) + [K^T_{n+1}(\{u^k_{n+1}\})]\right]\{\Delta u^k_{n+1}\} = \{R_{n+1}(\{u^k_{n+1}\})\} \tag{15.25}$$

where:

$a_0 = \dfrac{1}{\alpha\Delta t^2}$

$a_1 = \dfrac{\delta}{\alpha\Delta t}$

$\left[K^T_{n+1}(\{u^k_{n+1}\})\right]$ = the tangent stiffness matrix at time t_{n+1}

For nonlinear structural dynamics problems, the program allows a user to input the amplitude decay factor γ or the Newmark integration parameters on the **TINTP** command.

Generalized HHT-α Method

The generalized HHT-α method for nonlinear structural dynamics problems assumes:

$$[M]\{\ddot{u}_{n+1-\alpha_m}\} + [C]\{\dot{u}_{n+1-\alpha_f}\} + \{F^i_{n+1}(\{u_{n+1-\alpha_f}\})\} = \{F^a(t_{n+1-\alpha_f})\} \tag{15.26}$$

where:

$$\left\{F^i_{n+1}\left(\left\{u_{n+1-\alpha_f}\right\}\right)\right\} = (1-\alpha_f)\left\{F^i_{n+1}\left(\left\{u_{n+1}\right\}\right)\right\} + \alpha_f\left\{F^i_n\left(\left\{u_n\right\}\right)\right\}$$

$$\left\{F^a(t_{n+1-\alpha_f})\right\} = (1-\alpha_f)\left\{F^a_{n+1}\right\} + \alpha_f\left\{F^a_n\right\}$$

By introducing the residual vector $\{R_{n+1}(\{u_{n+1}\})\}$, Equation 15.26 (p. 770) can be written as:

$$\left\{R_{n+1}\left(\left\{u_{n+1}\right\}\right)\right\} = \left\{F^a(t_{n+1-\alpha_f})\right\} - \left\{F^i_{n+1}\left(\left\{u_{n+1-\alpha_f}\right\}\right)\right\} - [M]\left\{\ddot{u}_{n+1-\alpha_m}\right\} - [C]\left\{\dot{u}_{n+1-\alpha_f}\right\} \tag{15.27}$$

The time integration operator given in Equation 15.26 (p. 770) or Equation 15.27 (p. 770) also represents a nonlinear system of simultaneous algebraic equations. Therefore, a linearized form of the time integration operator can be obtained by the Newton-Raphson method as follows:

$$\left\{R_{n+1}\left(\left\{u_{n+1}^{k}\right\}\right)\right\}+\frac{\partial\left\{R_{n+1}\left(u_{n+1}^{k}\right)\right\}}{\partial\left\{u_{n+1}^{i}\right\}}\left\{\Delta u_{n+1}^{k}\right\}=\{0\} \tag{15.28}$$

where:

$$R_{n+1}\left(\left\{u_{n+1}^{k}\right\}\right)=\left\{F^{a}\left(t_{n+1-\alpha_f}\right)\right\}-\left\{F_{n+1}^{i}\left(\left\{u_{n+1-\alpha_f}^{k}\right\}\right)\right\}-[M]\left\{\ddot{u}_{n+1-\alpha_m}\right\}-[C]\left\{\dot{u}_{n+1-\alpha_f}\right\}$$

$$\left\{F_{n+1}^{i}\left(\left\{u_{n+1-\alpha_f}^{k}\right\}\right)\right\}=\left(1-\alpha_f\right)\left\{F_{n+1}^{i}\left(\left\{u_{n+1}^{k}\right\}\right)\right\}+\alpha_f\left\{F_{n}^{i}\left(\{u_n\}\right)\right\}$$

Equation 15.28 (p. 771) gives:

$$\left[\left(a_0[M]+a_1[C]\right)+\left(1-\alpha_f\right)\left[K_{n+1}^{T}\left(\left\{u_{n+1}^{k}\right\}\right)\right]\right]\left\{\Delta u_{n+1}^{k}\right\}=\left\{R_{n+1}\left(\left\{u_{n+1}^{k}\right\}\right)\right\} \tag{15.29}$$

where:

$$a_0=\frac{1-\alpha_m}{\alpha\Delta t^2}$$

$$a_1=\frac{(1-\alpha_f)\delta}{\alpha\Delta t}$$

15.2.2.3. Solution

Two methods of solution for the Newmark method (Equation 15.10 (p. 766)) are available: full and mode superposition (**TRNOPT** command). Each are described subsequently. Only the full solution method is available for HHT (Equation 15.15 (p. 767)).

Full Solution Method

The full solution method (**TRNOPT**,FULL) solves Equation 15.10 (p. 766) directly and makes no additional assumptions. In a nonlinear analysis, the Newton-Raphson method (Newton-Raphson Procedure (p. 711)) is employed along with the Newmark assumptions. Automatic Time Stepping (p. 685) discusses the procedure for the program to automatically determine the time step size required for each time step.

Inherent to the Newmark method is that the values of $\{u_o\}$, $\{\dot{u}_o\}$, and $\{\ddot{u}_o\}$ at the start of the transient must be known. Nonzero initial conditions are input either directly (with the **IC** commands) or by performing a static analysis load step (or load steps) prior to the start of the transient itself. Static load steps are performed in a transient analysis by turning off the transient time integration effects (with the **TIMINT**,OFF command). The transient itself can then be started (by **TIMINT**,ON). The default with transient analysis (**ANTYPE**,TRANS) is for the transient to be running (**TIMINT**,ON); that is, to start the transient immediately. (This implies $\{u\} = \{\dot{u}\} = \{\ddot{u}\} = 0$. The initial conditions are outlined in the subsequent paragraphs. Cases referring to "no previous load step" mean that the first load step is transient.

Initial Displacement -

The initial displacements are:

$$\{u_o\} = \begin{cases} \{0\} & \text{if no previous load step available and no initial conditions (\textbf{IC} commands) are used.} \\ \{u'_s\} & \text{if no previous load step available but initial conditions (\textbf{IC} commands) are used.} \\ \{u_s\} & \text{if previous load step available which was run as a static analysis (\textbf{TIMINT},OFF)} \end{cases} \quad (15.30)$$

where:

$\{u_o\}$ = vector of initial displacements

$\{u'_s\}$ = displacement vector specified by the initial conditions (**IC** command)

$\{u_s\}$ = displacement vector resulting from a static analysis (**TIMINT**,OFF) of the previous load step

Initial Velocity -

The initial velocities are:

$$\{\dot{u}_o\} = \begin{cases} \{0\} & \text{if no previous load step available and no initial conditions (\textbf{IC} commands) are used.} \\ \{\dot{u}'_s\} & \text{if no previous load step available but initial conditions (\textbf{IC} commands) are used.} \\ \dfrac{\{u_s\} - \{u_{s-1}\}}{\Delta t} & \text{if previous load step available which was run as a static analysis (\textbf{TIMINT},OFF)} \end{cases} \quad (15.31)$$

where:

$\{\dot{u}_o\}$ = vector of initial velocities

$\{\dot{u}'_s\}$ = vector of velocities specified by the initial conditions (**IC** commands)

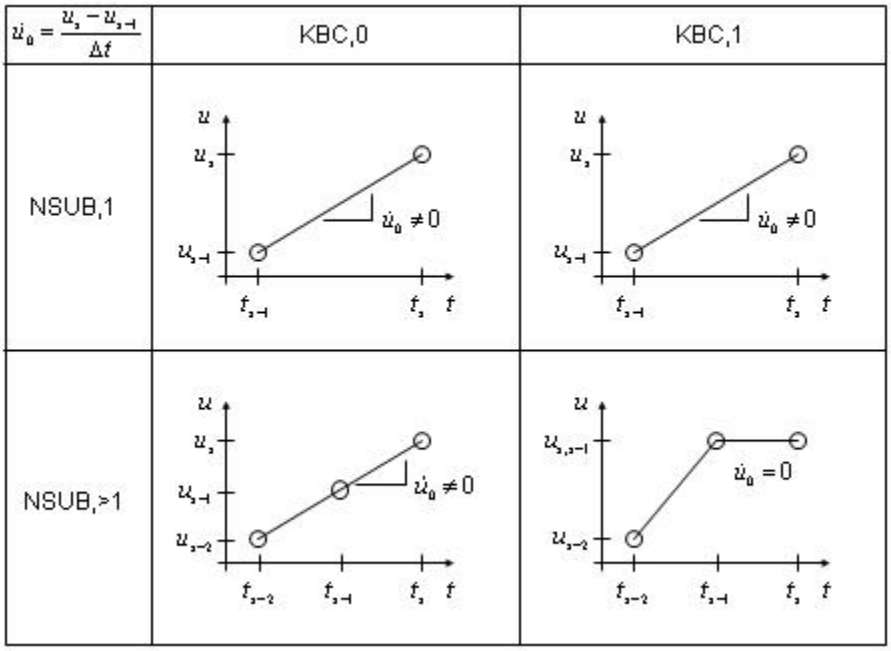
$\{u_s\}$ = displacements from a static analysis (**TIMINT**,OFF) of the previous load step

$\{u_{s-1}\}$ = displacement corresponding to the time point before $\{u_s\}$ solution. $\{u_{s-1}\}$ is $\{0\}$ if $\{u_s\}$ is the first solution of the analysis (i.e. load step 1 substep 1).

Δt = time increment between s and s-1

If the previous load step was run as a static analysis (**TIMINT**,OFF), initial velocities are calculated using the previous two displacements and the previous time increment. Using either a single substep (**NSUBST**,1) or ramped loading (**KBC**,0) within the previous load step will result in nonzero initial velocities (assuming nonzero displacement), as shown in Figure 15.2: Effect of Number of Substeps (NSUBST) and Ramping (KBC) on Initial Velocity for TIMINT,OFF (p. 773). Zero initial velocities may be obtained by using multiple substeps (**NSUBST**,>1) and stepped loading (**KBC**,1).

Figure 15.2: Effect of Number of Substeps (NSUBST) and Ramping (KBC) on Initial Velocity for TIMINT,OFF



Initial Acceleration -

The initial acceleration is simply:

$$\{\ddot{u}_o\} = \{0\} \tag{15.32}$$

where:

$\{\ddot{u}_o\}$ = vector of initial accelerations

If a nonzero initial acceleration is required as for a free fall problem, an extra load step at the beginning of the transient can be used. This load step would have a small time span, step boundary conditions, and a few time steps which would allow the acceleration to be well represented at the end of the load step.

Nodal and Reaction Load Computation -

Inertia, damping and static loads on the nodes of each element are computed.

The inertial load part of the element output is computed by:

$$\{F_e^m\} = -[M_e]\{\ddot{u}_e\} \tag{15.33}$$

where:

$\{F_e^m\}$ = vector of element inertial forces

[Me] = element mass matrix

$\{\ddot{u}_e\}$ = element acceleration vector

The acceleration of a typical DOF is given by Equation 15.12 (p. 766) for time t_{n+1}. By default, the acceleration vector $\{\ddot{u}_e\}$ is the average acceleration between time t_{n+1} and time t_n, since the Newmark assumptions (Equation 15.8 (p. 765) and Equation 15.9 (p. 765)) assume the average acceleration represents the true acceleration. Smoothing can be suppressed using the **TINTP** command (AVSMOOTH option).

The damping load part of the element output is computed by:

$$\{F_e^c\} = -[C_e]\{\dot{u}_e\} \tag{15.34}$$

where:

$\{F_e^c\}$ = vector of element damping forces

$[C_e]$ = element damping matrix

$\{\dot{u}_e\}$ = element velocity vector

The velocity of a typical DOF is given by Equation 15.11 (p. 766).

The static load is part of the element output computed in the same way as in a static analysis (Solving for Unknowns and Reactions (p. 690)). The nodal reaction loads are computed as the negative of the sum of all three types of loads (inertia, damping, and static) over all elements connected to a given fixed displacement node.

Mode Superposition Method

The mode superposition method (**TRNOPT**,MSUP) uses the natural frequencies and mode shapes of a linear structure to predict the response to transient forcing functions. This solution method imposes the following additional assumptions and restrictions:

1. Constant [K] and [M] matrices. (A gap condition is permitted.) This implies no large deflections or change of stress stiffening, as well as no plasticity, creep, or swelling.
2. Constant time step size.
3. There are no element damping matrices. However, various types of system damping are available.
4. Time varying imposed displacements are not allowed.

The development of the general mode superposition procedure is described in Mode Superposition Method (p. 698). Equation 14.118 (p. 700) and Equation 14.119 (p. 701) are integrated through time for each mode by the Newmark method.

The initial value of the modal coordinates at time = 0.0 are computed by solving Equation 14.118 (p. 700) with $\{\ddot{y}_o\}$ and $\{\dot{y}_o\}$ assumed to be zero.

$$y_j = \{\phi_j\}^T\{F_o\}/\omega_j^2 \tag{15.35}$$

where:

$\{F_o\}$ = the forces applied at time = 0.0

A "quasi-linear" analysis variation is also available with the mode superposition method. This variation allows interfaces (gaps) between any of the master DOFs and ground, or between any pair of master

DOFs. If the gap is initially closed, these interfaces are accounted for by including the stiffness of the interface in the stiffness matrix, but if the gap should later open, a force is applied in the load vector to nullify the effect to the stiffness. If the gap is initially open, it causes no effect on the initial solution, but if it should later close, a force is again applied in the load vector.

The force associated with the gap is:

$$F_{gp} = k_{gp} u_g \tag{15.36}$$

where:

k_{gp} = gap stiffness (input as *STIF*, **GP** command)

$u_g = u_A - u_B - u^{gp}$

u_A, u_B = displacement across gap (must be master degrees of freedom)

u^{gp}= initial size of gap (input as *GAP*, **GP** command)

This mode superposition method with a gap definition (**GP** command) adds an explicit term to the implicit integration procedure. An alternate procedure is to use the full method, modeling the linear portions of the structure as superelements (using the CMS method) and the gaps as gap elements. This latter procedure (implicit integration) normally allows larger time steps because it modifies both the stiffness matrix and load vector when the gaps change status.

The load vector, which must be converted to modal coordinates (Equation 14.117 (p. 700)) at each time step, is given by

$$\{F\} = \{F^{nd}\} + s\{F^{s}\} + \{F_{gp}\} + \{F_{ma}\} \tag{15.37}$$

where:

$\{F^{nd}\}$ = nodal force vector

s = load vector scale factor (input as *FACT*, **LVSCALE** command)

$\{F^{s}\}$ = load vector from the modal analysis (see Mode Superposition Method (p. 698)).

$\{F_{gp}\}$ = gap force vector (Equation 15.36 (p. 775)) (not available for QR damped eigensolver).

$\{F_{ma}\}$ = inertial force ($\{F_{ma}\} = [M]\{a\}$)

{a} = acceleration vector (input with **ACEL** command) (see Acceleration Effect (p. 665))

In the modal superposition method, the damping force associated with gap is added to Equation 15.36 (p. 775):

$$\{F_{gp}\} = [K_{gp}]\{u_g\} + C_{gp}\{\dot{u}_g\} \tag{15.38}$$

where:

C_{gp} = gap damping (input as DAMP, **GP** command)

$\{\dot{u}_g\} = \{\dot{u}_A\} - \{\dot{u}_B\}$

$\{\dot{u}_A\} - \{\dot{u}_B\}$ = velocity across gap

Expansion Pass -

The expansion pass of the mode superposition transient analysis involves computing element stresses.

Nodal load output consists of the static loads only as described for a static analysis (Solving for Unknowns and Reactions (p. 690)). The reaction load values represent the negative of the sum of the static loads over all elements connected to a given fixed displacement node. Damping and inertia forces are not included in the reaction loads.

15.2.3. Description of Thermal, Magnetic and Other First Order Systems

The governing equation of interest is as follows:

$$[C]\{\dot{u}\} + [K]\{u\} = \{F^a\} \tag{15.39}$$

where:

[C] = damping matrix
[K] = coefficient matrix
{u} = vector of DOF values
$\{\dot{u}\}$ = time rate of the DOF values
$\{F^a\}$ = applied load vector

In a thermal analysis, [C] is the specific heat matrix, [K] the conductivity matrix, {u} the vector of nodal temperatures and $\{F^a\}$ the applied heat flows. Table 15.2: Nomenclature (p. 776) relates the nomenclature used in Derivation of Heat Flow Matrices (p. 235) and Derivation of Electromagnetic Matrices (p. 193) for thermal, magnetic and electrical analyses to Equation 15.39 (p. 776).

Table 15.2: Nomenclature

	{u}	$\{F^a\}$
Thermal	{T} temperature	$\{Q^a\}$ heat flow
Scalar Magnetic	$\{\phi\}$ scalar potential	$\{F^a\}$ flux
Vector Magnetic	{A} vector potential	$\{F^a\}$ current segment
Electrical	{V} voltage	$\{I^a\}$ current

The mode superposition procedure does not apply to first order systems.

The procedure employed for the solution of Equation 15.39 (p. 776) is the generalized trapezoidal rule (Hughes([165] (p. 930))):

$$\{u_{n+1}\} = \{u_n\} + (1-\theta)\Delta t\{\dot{u}_n\} + \theta\Delta t\{\dot{u}_{n+1}\} \tag{15.40}$$

where:

θ = transient integration parameter (input on **TINTP** command)
$\Delta t = t_{n+1} - t_n$
$\{u_n\}$ = nodal DOF values at time t_n
$\{\dot{u}_n\}$ = time rate of the nodal DOF values at time t_n (computed at previous time step)

Equation 15.39 (p. 776) can be written at time t_{n+1} as:

$$[C]\{\dot{u}_{n+1}\} + [K]\{u_{n+1}\} = \{F^a\} \tag{15.41}$$

Substituting $\{\dot{u}_{n+1}\}$ from Equation 15.40 (p. 776) into this equation yields:

$$\left(\frac{1}{\theta\Delta t}[C]+[K]\right)\{u_{n+1}\}=\{F^a\}+[C]\left(\frac{1}{\theta\Delta t}\{u_n\}+\frac{1-\theta}{\theta}\{\dot{u}_n\}\right) \tag{15.42}$$

The solution of Equation 15.42 (p. 777) employs the same solvers used for static analysis in Static Analysis (p. 761). Once {un+1} is obtained, $\{\dot{u}_{n+1}\}$ is updated using Equation 15.40 (p. 776). In a nonlinear analysis, the Newton-Raphson method (Newton-Raphson Procedure (p. 711)) is employed along with the generalized trapezoidal assumption, Equation 15.40 (p. 776).

The transient integration parameter θ (input on **TINTP** command) defaults to 0.5 (Crank-Nicholson method) if solution control is not used (**SOLCONTROL**,OFF) and 1.0 (backward Euler method) if solution control is used (**SOLCONTROL**,ON). If $\theta = 1$, the method is referred to as the backward Euler method. For all $\theta > 0$, the system equations that follow are said to be implicit. In addition, for the more limiting case of $\theta \geq 1/2$, the solution of these equations is said to be unconditionally stable; i.e., stability is not a factor in time step (Δt) selection. The available range of θ (using **TINTP** command) is therefore limited to

$$\frac{1}{2}\leq\theta\leq 1 \tag{15.43}$$

which corresponds to an unconditionally stable, implicit method. For a piezoelectric analysis, the Crank-Nicholson and constant average acceleration methods must both be requested with α (ALPHA) = 0.25, δ (DELTA) = 0.5, and θ = 0.5 (on the **TINTP** command). Since the $\{\dot{u}_n\}$ influences $\{u_{n+1}\}$, sudden changes in loading need to be handled carefully for values of $\theta < 1.0$. See the *Basic Analysis Guide* for more details.

The generalized-trapezoidal method requires that the values of $\{u_o\}$ and $\{\dot{u}_o\}$ at the start of the transient must be known. Nonzero initial conditions are input either directly (with the **IC** command) (for $\{u_o\}$) or by performing a static analysis load step (or load steps) prior to the start of the transient itself. Static load steps are performed in a transient analysis by turning off the transient time integration effects (with the **TIMINT**,OFF command). The transient itself can then started (**TIMINT**,ON). The default for transient analysis (**ANTYPE**,TRANS) is to start the transient immediately (**TIMINT**,ON). This implies ($\{u\} = \{\dot{u}\} = \{0\}$). The initial conditions are outlined in the subsequent paragraphs.

Initial DOF Values -

The initial DOF values for first order systems are:

$$\{u_o\}=\begin{cases}\{a\} & \text{if no previous load step available and no initial conditions (}\textbf{IC}\text{ commands) are used}\\ \{u_s'\} & \text{if no previous load step available but the initial conditions (}\textbf{IC}\text{ commands) are used}\\ \{u_s\} & \text{if previous load step available run as a static analysis (}\textbf{TIMINT}\text{,OFF)}\end{cases} \tag{15.44}$$

where:

$\{u_o\}$ = vector of initial DOF values

$\{a\}$ = vector of uniform DOF values

$\{u'_s\}$ = DOF vector directly specified (**IC** command)

$\{u_s\}$ = DOF vector resulting from a static analysis (**TIMINT**,OFF) of the previous load step available

{a} is set to TEMP (**BFUNIF** command) and/or to the temperature specified by the initial conditions (**IC** commands) for thermal DOFs (temperatures) and zero for other DOFs.

Nodal and Reaction Load Computation -

Damping and static loads on the nodes of each element are computed.

The damping load part of the element output is computed by:

$$\{F_e^c\} = [C_e]\{\dot{u}_e\} \tag{15.45}$$

where:

$\{F_e^c\}$ = vector of element damping loads

$[C_e]$ = element damping matrix

$\{\dot{u}_e\}$ = element velocity vector

The velocity of a typical DOF is given by Equation 15.40 (p. 776). The velocity vector $\{\dot{u}_e\}$ is the average velocity between time t_n and time t_{n+1}, since the general trapezoidal rule (Equation 15.40 (p. 776)) assumes the average velocity represents the true velocity.

The static load is part of the element output computed in the same way as in a static analysis (Solving for Unknowns and Reactions (p. 690)). The nodal reaction loads are computed as the negative of the sum of both types of loads (damping and static) over all elements connected to a given fixed DOF node.

15.3. Mode-Frequency Analysis

The following mode frequency analysis topics are available:

15.3.1. Assumptions and Restrictions

1. Valid for structural and fluid degrees of freedom (DOFs). Electrical and thermal DOFs may be present in the coupled field mode-frequency analysis using structural DOFs.
2. The structure has constant stiffness and mass effects.
3. There is no damping, unless the damped eigensolver (**MODOPT**,DAMP or **MODOPT**,QRDAMP) is selected.
4. The structure has no time varying forces, displacements, pressures, or temperatures applied (free vibration).

15.3.2. Description of Analysis

This analysis type (accessed with **ANTYPE**,MODAL) is used for natural frequency and mode shape determination. The equation of motion for an undamped system, expressed in matrix notation using the above assumptions is:

$$[M]\{\ddot{u}\} + [K]\{u\} = \{0\} \tag{15.46}$$

Note that [K], the structure stiffness matrix, may include prestress effects (**PSTRES**,ON). For a discussion of the damped eigensolver (**MODOPT**,DAMP or **MODOPT**,QRDAMP) see Eigenvalue and Eigenvector Extraction (p. 726).

For a linear system, free vibrations will be harmonic of the form:

$$\{u\} = \{\phi\}_i \cos \omega_i t \tag{15.47}$$

where:

$\{\phi\}_i$ = eigenvector representing the mode shape of the ith natural frequency
ω_i = ith natural circular frequency (radians per unit time)
t = time

Thus, Equation 15.46 (p. 779) becomes:

$$(-\omega_i^2[M] + [K])\{\phi\}_i = \{0\} \tag{15.48}$$

This equality is satisfied if either $\{\phi\}_i = \{0\}$ or if the determinant of ([K] - ω^2 [M]) is zero. The first option is the trivial one and, therefore, is not of interest. Thus, the second one gives the solution:

$$\left| [K] - \omega^2[M] \right| = 0 \tag{15.49}$$

This is an eigenvalue problem which may be solved for up to n values of ω^2 and n eigenvectors $\{\phi\}_i$ which satisfy Equation 15.48 (p. 779) where n is the number of DOFs. The eigenvalue and eigenvector extraction techniques are discussed in Eigenvalue and Eigenvector Extraction (p. 726).

Rather than outputting the natural circular frequencies $\{\omega\}$, the natural frequencies (f) are output; where:

$$f_i = \frac{\omega_i}{2\pi} \tag{15.50}$$

where:

f_i = ith natural frequency (cycles per unit time)

If normalization of each eigenvector $\{\phi\}_i$ to the mass matrix is selected (**MODOPT**,,,,,,OFF):

$$\{\phi\}_i^T [M]\{\phi\}_i = 1 \tag{15.51}$$

If normalization of each eigenvector $\{\phi\}_i$ to 1.0 is selected (**MODOPT**,,,,,,ON), $\{\phi\}_i$ is normalized such that its largest component is 1.0 (unity).

A discussion of effective mass is given in Spectrum Analysis (p. 799).

15.4. Harmonic Analysis

The following harmonic analysis topics are available:

The harmonic analysis (**ANTYPE**,HARMIC) solves the time-dependent equations of motion (Equation 15.5 (p. 764)) for linear structures undergoing steady-state vibration.

15.4.1. Harmonic Analysis Assumptions and Restrictions

1. Valid for structural, fluid, magnetic, and electrical degrees of freedom (DOFs). Thermal DOFs may be present in a coupled field harmonic analysis using structural DOFs.
2. The entire structure has constant or frequency-dependent stiffness, damping, and mass effects.
3. All loads and displacements vary sinusoidally at the same known frequency (although not necessarily in phase).
4. Element loads are assumed to be real (in-phase) only, except for:
   a. current density
   b. pressures in SURF153, SURF154, SURF156, and SURF159

15.4.2. Description of Harmonic Analysis

Consider the general equation of motion for a structural system (Equation 15.5 (p. 764)).

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F^a\} \tag{15.52}$$

where:

[M] = structural mass matrix
[C] = structural damping matrix
[K] = structural stiffness matrix
$\{\ddot{u}\}$ = nodal acceleration vector
$\{\dot{u}\}$ = nodal velocity vector
{u} = nodal displacement vector
$\{F^a\}$ = applied load vector

As stated above, all points in the structure are moving at the same known frequency, however, not necessarily in phase. Also, it is known that the presence of damping causes phase shifts. Therefore, the displacements may be defined as:

$$\{u\} = \{u_{max} e^{i\phi}\} e^{i\Omega t} \tag{15.53}$$

where:

u_{max} = maximum displacement
i = square root of -1
Ω= imposed circular frequency (radians/time) = $2\pi f$
f = imposed frequency (cycles/time) (input as *FREQB* and *FREQE* on the **HARFRQ** command)
t = time
Φ = displacement phase shift (radians)

Note that u_{max} and Φ may be different at each DOF. The use of complex notation allows a compact and efficient description and solution of the problem. Equation 15.53 (p. 781) can be rewritten as:

$$\{u\} = \{u_{max}(\cos\phi + i\sin\phi)\} e^{i\Omega t} \tag{15.54}$$

or as:

$$\{u\} = (\{u_1\} + i\{u_2\}) e^{i\Omega t} \tag{15.55}$$

where:

$\{u_1\} = \{u_{max} \cos \Phi\}$ = real displacement vector (input as *VALUE* on **D** command, when specified)
$\{u_2\} = \{u_{max} \sin \Phi\}$ = imaginary displacement vector (input as *VALUE2* on **D** command, when specified)

The force vector can be specified analogously to the displacement:

$$\{F\} = \{F_{max} e^{i\psi}\} e^{i\Omega t} \tag{15.56}$$

$$\{F\} = \{F_{max}(\cos\psi + i\sin\psi)\} e^{i\Omega t} \tag{15.57}$$

$$\{F\} = (\{F_1\} + i\{F_2\}) e^{i\Omega t} \tag{15.58}$$

where:

F_{max} = force amplitude
ψ = force phase shift (radians)
$\{F_1\} = \{F_{max} \cos \psi\}$ = real force vector (input as *VALUE* on **F** command, when specified)
$\{F_1\} = \{F_{max} \sin \psi\}$ = imaginary force vector (input as on *VALUE2* on **F** command, when specified)

Substituting Equation 15.55 (p. 781) and Equation 15.58 (p. 781) into Equation 15.52 (p. 780) gives:

$$(-\Omega^2[M] + i\Omega[C] + [K])(\{u_1\} + i\{u_2\}) e^{i\Omega t} = (\{F_1\} + i\{F_2\}) e^{i\Omega t} \tag{15.59}$$

The dependence on time ($e^{i\Omega t}$) is the same on both sides of the equation and may therefore be removed:

$$([K] - \Omega^2[M] + i\Omega[C])(\{u_1\} + i\{u_2\}) = \{F_1\} + i\{F_2\} \tag{15.60}$$

The solution of this equation is discussed later.

15.4.3. Harmonic Analysis Complex Displacement Output

The complex displacement output at each DOF may be given in one of two forms:

1. The same form as u_1 and u_2 as defined in Equation 15.55 (p. 781) (selected with the command **HROUT**,ON).
2. The form u_{max} and Φ (amplitude and phase angle (in degrees)), as defined in Equation 15.54 (p. 781) (selected with the command **HROUT**,OFF). These two terms are computed at each DOF as:

$$u_{max} = \sqrt{u_1^2 + u_2^2} \tag{15.61}$$

$$\phi = \tan^{-1}\frac{u_2}{u_1} \tag{15.62}$$

Note that the response lags the excitation by a phase angle of Φ-Ψ.

15.4.4. Nodal and Reaction Load Computation in a Harmonic Analysis

Inertia, damping and static loads on the nodes of each element are computed.

The inertia load is expressed as:

$$\left\{F^m\right\}_e = \Omega^2\left[M_e\right]\{u\}_e \tag{15.63}$$

where:

$\{F^m\}_e$ = vector of element inertia forces
$[M_e]$ = element mass matrix
$\{u\}_e$ = element displacement vector

The real and imaginary inertia load parts of the element output are then computed by:

$$\{F_1^m\}_e = \Omega^2[M_e]\{u_1\}_e \tag{15.64}$$

$$\{F_2^m\}_e = \Omega^2[M_e]\{u_2\}_e \tag{15.65}$$

where:

$\{F_1^m\}_e$ = vector of element inertia forces (real part)
$\{u_1\}_e$ = element real displacement vector
$\{F_2^m\}_e$ = vector of element inertia forces (imaginary part)
$\{u_2\}_e$ = element imaginary displacement vector

The damping load is expressed as:

$$\left\{F^c\right\}_e = -j\Omega\left[C_e\right]\{u\}_e \tag{15.66}$$

where:

$\{F^c\}_e$ = vector of element damping forces
$[C_e]$ = element damping matrix

The real and imaginary damping loads part of the element output are then computed by:

$$\{F_1^c\}_e = -\Omega[C_e]\{u_2\}_e \tag{15.67}$$

$$\{F_2^c\}_e = \Omega[C_e]\{u_1\}_e \tag{15.68}$$

where:

$\{F_1^c\}_e$ = vector of element damping forces (real part)

$\{F_2^c\}_e$ = vector of element damping forces (imaginary part)

The real static load is computed the same way as in a static analysis (Solving for Unknowns and Reactions (p. 690)) using the real part of the displacement solution $\{u_1\}_e$. The imaginary static load is computed also the same way, using the imaginary part $\{u_2\}_e$. Note that the imaginary part of the element loads (e.g., $\{F^{pr}\}$) are normally zero, except for current density loads.

The nodal reaction loads are computed as the sum of all three types of loads (inertia, damping, and static) over all elements connected to a given fixed displacement node.

15.4.5. Harmonic Analysis Solution

Three methods of solution to Equation 15.60 (p. 781) are available: full, mode superposition, and Variational Technology. Each are described subsequently.

15.4.5.1. Full Solution Method

The full solution method (**HROPT**,FULL) solves Equation 15.60 (p. 781) directly. Equation 15.60 (p. 781) may be expressed as:

$$[K_c]\{u_c\} = \{F_c\} \tag{15.69}$$

where c denotes a complex matrix or vector. Equation 15.69 (p. 783) is solved using the same sparse solver used for a static analysis in Equation Solvers (p. 694), except that it is done using complex arithmetic.

15.4.5.2. Mode Superposition Method

The mode superposition method (**HROPT**,MSUP) uses the natural frequencies and mode shapes to compute the response to a sinusoidally varying forcing function. This solution method imposes the following additional assumptions and restrictions:

1. Nonzero imposed harmonic displacements are not allowed.

2. There are no element damping matrices. However, various types of system damping are available.

The development of the general mode superposition procedure is given in Mode Superposition Method (p. 698). The equation of motion (Equation 15.52 (p. 780)) is converted to modal form, as described in Mode Superposition Method (p. 698). Equation 14.118 (p. 700) is:

$$\ddot{y}_j + 2\omega_j\xi_j\dot{y}_j + \omega_j^2 y_j = f_j \tag{15.70}$$

where:

y_j = modal coordinate
ω_j = natural circular frequency of mode j
ξ_i = fraction of critical damping for mode j
f_j = force in modal coordinates

The load vector which is converted to modal coordinates (Equation 14.117 (p. 700)) is given by

$$\{F\} = \{F^{nd}\} + s\{F^s\} \tag{15.71}$$

where:

$\{F^{nd}\}$ = nodal force vector
s = load vector scale factor, (input as *FACT*, **LVSCALE** command)
$\{F^s\}$ = load vector from the modal analysis (see Mode Superposition Method (p. 698)).

For a steady sinusoidal vibration, f_j has the form

$$f_j = f_{jc}e^{i\Omega t} \tag{15.72}$$

where:

f_{jc} = complex force amplitude
Ω = imposed circular frequency

For Equation 15.70 (p. 784) to be true at all times, yj must have a similar form as f_j, or

$$y_j = y_{jc}e^{i\Omega t} \tag{15.73}$$

where:

y_{jc} = complex amplitude of the modal coordinate for mode j.

Differentiating Equation 15.73 (p. 784), and substituting Equation 15.72 (p. 784) and Equation 15.73 (p. 784) into Equation 15.70 (p. 784),

$$-\Omega^2 y_{jc}e^{i\Omega t} + 2\omega_j\xi_j(i\Omega y_{jc}e^{i\Omega t}) + \omega_j^2 y_{jc}e^{i\Omega t} = f_{jc}e^{i\Omega t} \tag{15.74}$$

Collecting coefficients of y_{jc} and dividing by $(e^{i\Omega t})$

$$(-\Omega^2 + i2\omega_j\Omega\xi_j + \omega_j^2)y_{jc} = f_{jc} \tag{15.75}$$

solving for y_{jc},

$$y_{jc} = \frac{f_{jc}}{(\omega_j^2 - \Omega^2) + i(2\omega_j\Omega\xi_j)} \tag{15.76}$$

The contribution from each mode is:

$$\{C_j\} = \{\phi_j\} y_{jc} \tag{15.77}$$

where:

$\{C_j\}$ = contribution of mode j (output if *Mcont* = ON, on the **HROUT** command)
$\{\phi_j\}$ = mode shape for mode j

Finally, the complex displacements are obtained from Equation 14.100 (p. 698) as

$$\{u_c\} = \sum_{j=1}^{n} \{C_j\} \tag{15.78}$$

where:

$\{u_c\}$ = vector of complex displacements

In the case of the QR damped mode extraction method, one substitutes Equation 14.119 (p. 701) for Equation 14.118 (p. 700), so Equation 15.75 (p. 784) becomes:

$$\left[-\Omega^2[I] + i\Omega[\Phi]^T[C][\Phi] + [\Lambda^2]\right]\{y\} = [\Phi]^T\{F\} \tag{15.79}$$

Solving the above equation and multiplying by the eigenvectors, one can calculate the complex displacements shown in Equation 15.78 (p. 785).

15.4.5.2.1. Expansion Pass

The expansion pass of the mode superposition method involves computing element stresses.

15.4.6. Harmonic Analysis Variational Technology Method

A common way to compute the harmonic response of a structure is to compute the normal modes of the undamped structure, and to use a modal superposition method to evaluate the response, after determining the modal damping. Determining the modal damping can be based on modal testing, or by using empirical rules. However, when the structure is non-metallic, the elastic properties can be highly dependent on the frequency and the damping can be high enough that the undamped modes and the damped modes are significantly different, and an approach based on a real, undamped modes is not appropriate.

One alternative to straight modal analysis is to build multiple modal bases, for different property values, and combine them together over the frequency range of the analysis. This technique is complex, error prone, and does not address the problem of determining the modal damping factors. Another alternative is a direct frequency response, updating the elastic properties for every frequency step. This technique give a much better prediction of the frequency response, but is CPU intensive.

The variational technology method (**HROPT**,VT) is available as the harmonic sweep capability of the VT Accelerator add-on. You can define the material elastic properties as being frequency-dependent (using **TB**,ELASTIC and **TB**,SDAMP) and efficiently compute the frequency response over an entire frequency range. For the Variational Technology theory, see Guillaume([333] (p. 939)) and Beley, Broudiscou, et al.([360] (p. 941)).

15.4.6.1. Viscous or Hysteretic Damping

When using the Variational Technology method, the user has a choice between viscous and hysteretic damping.

Viscous Damping

Consider a spring-damper-mass system subjected to a harmonic excitation. The response of the system is given by:

$$\{u\} = \{u_{max} e^{i\phi}\} e^{i\Omega t} \tag{15.80}$$

Due to the damping, the system is not conservative and the energy is dissipated. Using viscous damping, the energy dissipated by the cycle is proportional to the frequency, Ω. In a single DOF spring-mass-damper system, with a viscous damper C:

$$\Delta U = C\pi\Omega u_{max}^2 \tag{15.81}$$

where:

ΔU = change of energy
C = viscous damper

Hysteretic Damping

Experience shows that energy dissipated by internal friction in a real system does not depend on frequency, and approximately is a function of u_{max}^2:

$$\Delta U = \beta^{\xi} u_{max}^2 \tag{15.82}$$

where:

β^{ξ} = frequency-dependent damping (input using **TB**,SDAMP command)

β^{ξ} damping is known as structural or hysteretic damping. It can be included in the elastic properties by using a complex Young's modulus:

$$E_{complex} = E(1 + i\beta^{\xi}) \tag{15.83}$$

where:

E = Young's modulus (input using **TB**,ELASTIC command)

Using this kind of representation, the equations of motion of the system become:

$$[M]\{\ddot{u}\} + [K + iH]\{u\} = \{F^a\} \tag{15.84}$$

where:

[M] = structural mass matrix
[K] = structural stiffness matrix
[H] = structural damping matrix
$\{\ddot{u}\}$ = nodal acceleration vector
{u} = nodal displacement vector

$\{F^a\}$ = applied load vector

Note

The Variational Technology method does not support a material with both viscous and hysteretic damping properties in the same model. In the case, please use the **HROPT**,FULL command to analyze the material.

15.4.7. Automatic Frequency Spacing in a Harmonic Analysis

In a harmonic analysis, the imposed frequencies that involve the most kinetic energy are those near the natural frequencies of the structure. The automatic frequency spacing or "cluster" option (*Clust* = ON, on the **HROUT** command) provides an approximate method of choosing suitable imposed frequencies. The nearness of the imposed frequencies to the natural frequencies depends on damping, because the resonance peaks narrow when the damping is reduced. Figure 15.3: Frequency Spacing (p. 788) shows two typical resonance peaks and the imposed frequencies chosen by this method, which are computed from:

$$\Omega^i_{-j} = \omega_i / a_{ij} \tag{15.85}$$

$$\Omega^i_{+j} = \omega_i a_{ij} \tag{15.86}$$

Equation 15.85 (p. 787) gives frequencies slightly less than the natural circular frequency ω_j. Equation 15.86 (p. 787) gives slightly higher frequencies. The spacing parameter a_{ij} is defined as:

$$a_{ij} = 1 + (\xi_i)^b \tag{15.87}$$

where:

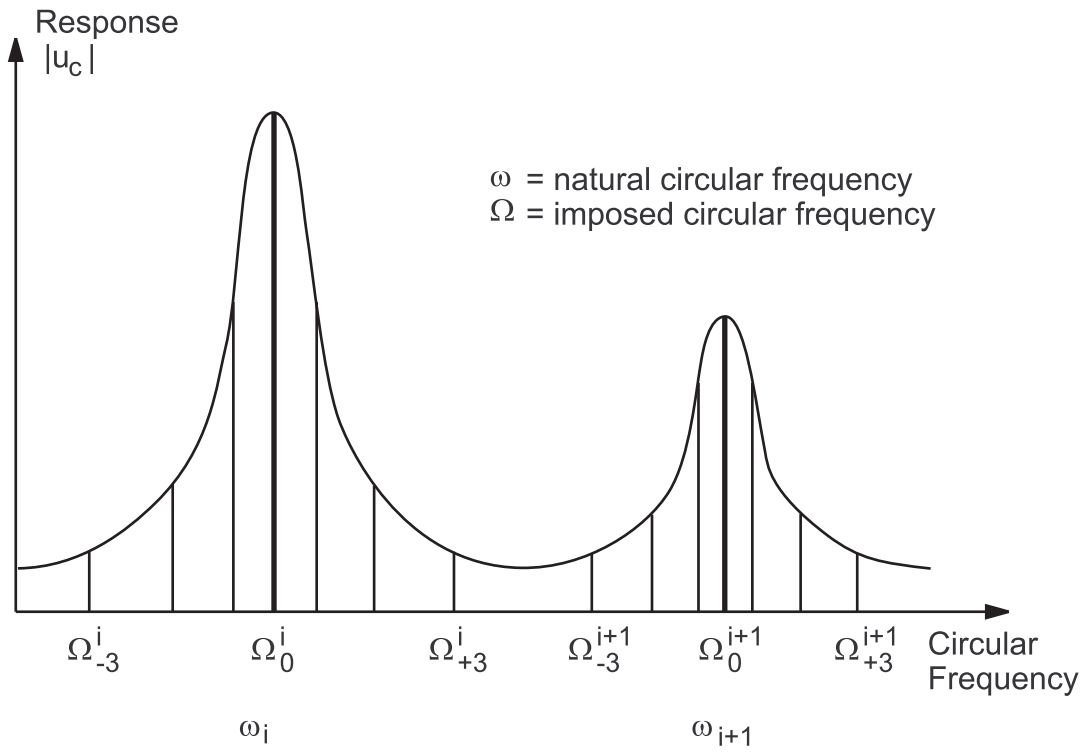
ξ_i = modal damping. (If ξ_i is computed as 0.0, it is redefined to be 0.005 for this equation only).

$$b = \frac{2(N-j)}{N-1}$$

N = integer constant (input as *NSBSTP*, **NSUBST** command) which may be between 2 and 20. Anything above this range defaults to 10 and anything below this range defaults to 4.
j = 1, 2, 3, ... N

Each natural frequency, as well as frequencies midway between, are also chosen as imposed frequencies.

Figure 15.3: Frequency Spacing



15.4.8. Logarithm Frequency Spacing in a Harmonic Analysis

The logarithm frequency spacing can be defined over an n^{th}-octave band or a general-frequency band in a harmonic analysis (the **HARFRQ** command with *LogOpt*).

The octave band is defined by the lower-end frequency:

$$f_{low} = f_0 / (\sqrt{2})^{1/n} \tag{15.88}$$

and upper-end frequency:

$$f_{low} = (\sqrt{2})^{1/n} f_0 \tag{15.89}$$

where the central frequency f_0 of the octave band is specified at 16, 31.6, 63, 125, 250, 500, 1000, 2000, 4000, 8000, 16000 Hz for 11 octave bands, and n = 1 (octave band), 2 (1/2 octave band), 3 (1/3 octave band), 6 (1/6 octave band), 12 (1/12 octave band), and 24 (1/24 octave band).

The recursive relationships between the central frequencies of the specified octave bands are:

$$f_{m+1}^{ctr} = 2^{1/n} f_m^{ctr} \tag{15.90}$$

$$f_{m-1}^{ctr} = f_m^{ctr} / 2^{1/n} \tag{15.91}$$

The logarithm frequency increment is given by:

$$f_{inc} = \left(\log_{10}(f_{up}) - \log_{10}(f_{low})\right) / (NSBSTP - 1) \tag{15.92}$$

Where *NSBSTP* is input by the **NSUBST** command.

The imposed working frequency is calculated by:

$$f_i = 10^{(\log_{10}(f_{low})+(i-1)f_{inc})} \quad \text{where i = 1, 2, ..., NSBSTP} \tag{15.93}$$

The imposed working frequency is f_0, if *NSBSTP* = 1. For a general logarithm frequency span, the beginning frequency f_{low} and ending frequency f_{up} are defined by the FREQB and FREQE parameters of the **HARFRQ** command respectively.

15.4.9. Harmonic Analysis with Rotating Forces on Rotating Structures

If a structure is rotating, forces rotating synchronously or asynchronously with the structure are of interest.

General rotating asynchronous forces are described in General Asynchronous Rotating Force (p. 789). A specific synchronous force: mass unbalance is shown in Specific Synchronous Forces: Mass Unbalance (p. 790).

In both cases, the equation solved for harmonic analysis is the same as (Equation 15.60 (p. 781)) except for the coefficients of the damping matrix [C] which will be a function of the rotational velocity of the structure (see the **CORIOLIS** command). [C] will be updated for each excitation frequency step using the following rotational velocity:

$$\omega = \Omega / s \tag{15.94}$$

where:

ω = rotational velocity of the structure (rd/s)
Ω = frequency of excitation (rd/s)
s = ratio between Ω and ω (s = 1 for synchronous excitations) (input as *RATIO* in the **SYNCHRO** command).

The right-hand term of the equation is given below depending on the force considered.

15.4.9.1. General Asynchronous Rotating Force

If the structure is rotating about X axis, then an asynchronous force having its direction in the plane perpendicular to the spin axis is expressed as:

$$F_y = F\cos\alpha\cos(s\omega t) + F\sin\alpha\sin(s\omega t) \tag{15.95}$$

$$F_z = F\cos\alpha\sin(s\omega t) - F\sin\alpha\cos(s\omega t) \tag{15.96}$$

where:

F = amplitude of force

Using complex notations, the equations become:

$$F_y = (F_a - iF_b)e^{is\omega t} \tag{15.97}$$

$$F_z = (-F_b - iF_a)e^{is\omega t} \tag{15.98}$$

where:

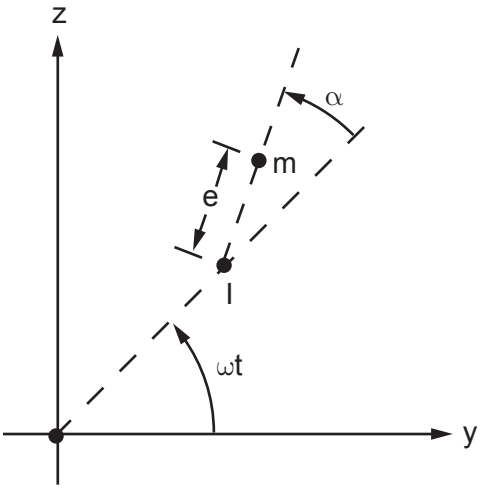
i = square root of -1
F_a = Fcosα = real force in Y-direction; also, negative of imaginary force in Z-direction
(input as *VALUE* on **F** command, label FY; input as *VALUE2* on **F** command, label FZ)
F_b = Fsinα = negative of real force in Z-direction; also, negative of imaginary force in the Y-direction
(input as *VALUE* on **F** command, label FZ; input as *VALUE2* on **F** command, label FY)

The expression of the forces for structures rotating about another direction than X are developed analogously.

15.4.9.2. Specific Synchronous Forces: Mass Unbalance

Consider a structure rotating about the X axis. The mass unbalance m situated at node I with the eccentricity e may be represented as shown in Figure 15.4: Mass Unbalance at Node I (p. 790)

Figure 15.4: Mass Unbalance at Node I



If we only consider the motion in the plane perpendicular to the spin axis (YZ plane), the kinetic energy of the unbalanced mass is written as:

$$E_k^u = \frac{m}{2}(\dot{u}_y^2 + \dot{u}_z^2 - 2\omega e\dot{u}_y \sin(\omega t + \alpha) + 2\omega e\dot{u}_z \cos(\omega t + \alpha) + \omega^2 e^2) \quad (15.99)$$

where:

m = mass unbalance
e = distance from the mass unbalance to the spin axis
ω = amplitude of the rotational velocity vector of the structure (input as **OMEGA** or **CMOMEGA** command). It is equal to the frequency of excitation Ω.
α = phase of the unbalance
$\dot{u}_y, \dot{u}_z$ = instantaneous velocity along Y and Z, respectively

Because the mass unbalance is much smaller than the weight of the structure, the first two terms are neglected. The third term being constant, will have no effect on the final equations.

Applying Lagrange's equations, the force vector is:

$$F_y = \omega^2(F\cos\alpha\cos(\omega t) + F\sin\alpha\sin(\omega t)) \quad (15.100)$$

$$F_z = \omega^2(F\cos\alpha\sin(\omega t) - F\sin\alpha\cos(\omega t)) \quad (15.101)$$

where:

F = me

Using complex notations, it can be written as:

$$F_y = \omega^2(F_a - iF_b)e^{i\omega t} \quad (15.102)$$

$$F_z = \omega^2(-F_b - iF_a)e^{i\omega t} \quad (15.103)$$

Note

The multiplication of the forces by ω^2 is done internally at each frequency step.

15.4.10. Harmonic Ocean Wave Procedure (HOWP)

Standard harmonic analysis with ocean loading includes the added mass of the water outside of line elements (such as LINK180, BEAM188, BEAM189, PIPE288, and PIPE289). The load vector is calculated based on the loads at a given time, but the standard analysis method usually misses some important effects, as all peak wave loads rarely occur at the same time.

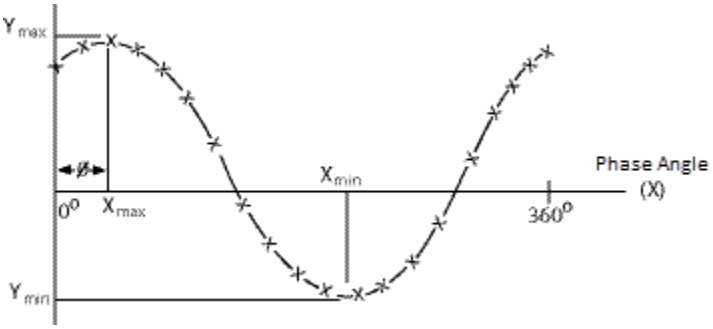
All relevant ocean wave loading effects are accounted for via a specialized variation of the harmonic analysis, the harmonic ocean wave procedure (HOWP).

The HOWP applies to regular waves only (Airy and Wheeler one-component waves, as well as Stokes and Deans Stream Function waves), and works only with the full-solution harmonic analysis method (**HROPT**,FULL).

The HOWP is accessed via the **HROCEAN** command. The frequency is obtained automatically, directly from the ocean information (via **OCDATA** and **OCTABLE** commands). As with a standard harmonic analysis, a damping matrix must be added separately if one is needed. Ocean loads are calculated with the assumption that the structure is stationary.

During the HOWP, special calculations occur to obtain real and imaginary loads for every component of every element. Stated differently, all loads are sinusoidal at the given frequency, but the magnitude and phase angle of every component of every element must be determined. To determine the magnitude and phase angle, a series of static analyses is performed along the phase angle ranging from 0 to 360 degrees. (The number of these analyses is controlled via the **HROCEAN** command.) The result is a roughly sinusoidal force pattern, as shown in this figure:

Figure 15.5: Force History



From this force history, the maximum and minimum forces are calculated by fitting the highest and lowest points. Then:

$$A = (Y_{max} - Y_{min}) / 2 \tag{15.104}$$

$$\varphi = \begin{cases} (X_{max} + X_{min} - 180) / 2 & \text{if } X_{max} < X_{min} \\ (X_{max} + X_{min} + 180) / 2 & \text{if } X_{max} \geq X_{min} \end{cases} \tag{15.105}$$

$$Y = A\cos(X - \varphi) = A\cos\varphi\cos X + A\sin\varphi\sin X \tag{15.106}$$

so that A cos φ is the coefficient on the real load vector and A sin φ is the coefficient on the imaginary load vector.

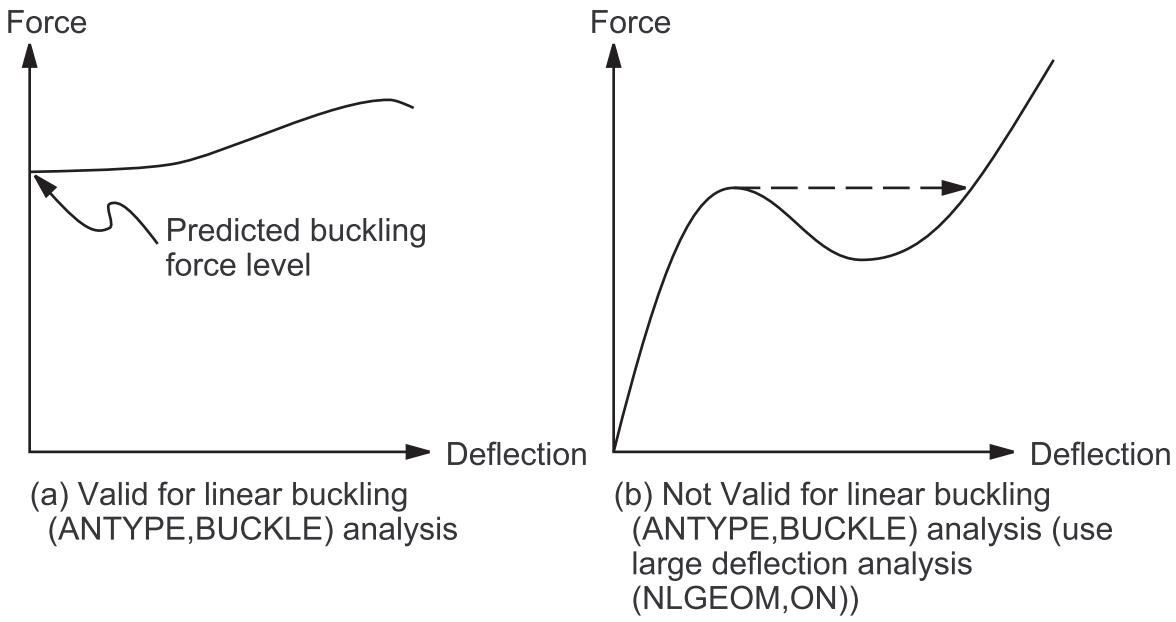
15.5. Buckling Analysis

The following buckling analysis topics are available:

15.5.1. Assumptions and Restrictions

1. Valid for structural degrees of freedom (DOFs) only.
2. The structure fails suddenly, with a horizontal force-deflection curve (see Figure 15.6: Types of Buckling Problems (p. 793)).
3. The structure has constant stiffness effects.
4. A static solution with prestress effects included (**PSTRES**,ON) was run.

Figure 15.6: Types of Buckling Problems



15.5.2. Description of Analysis

This analysis type is for bifurcation buckling using a linearized model of elastic stability. Bifurcation buckling refers to the unbounded growth of a new deformation pattern. A linear structure with a force-deflection curve similar to Figure 15.6: Types of Buckling Problems (p. 793)(a) is well modeled by a linear buckling (**ANTYPE**,BUCKLE) analysis, whereas a structure with a curve like Figure 15.6: Types of Buckling Problems (p. 793)(b) is not (a large deflection analysis (**NLGEOM**,ON) is appropriate, see Large Rotation (p. 35)). The buckling problem is formulated as an eigenvalue problem:

$$([K] + \lambda_i[S])\{\psi\}_i = \{0\} \quad (15.107)$$

where:

[K] = stiffness matrix
[S] = stress stiffness matrix
λ_i = ith eigenvalue (used to multiply the loads which generated [S])
ψ_i = ith eigenvector of displacements

The eigenproblem is solved as discussed in Eigenvalue and Eigenvector Extraction (p. 726). The eigenvectors are normalized so that the largest component is 1.0. Thus, the stresses (when output) may only be interpreted as a relative distribution of stresses.

By default, the Block Lanczos and Subspace Iteration methods find buckling modes in the range of negative infinity to positive infinity. If the first eigenvalue closest to the shift point is negative (indicating that the loads applied in a reverse direction will cause buckling), the program should find this eigenvalue.

15.6. Substructuring Analysis

The substructure analysis (**ANTYPE**,SUBSTR) uses the technique of matrix reduction to reduce the system matrices to a smaller set of DOFs.

The following substructuring analysis topics are available:

15.6.1. Assumptions and Restrictions (within Superelement)

1. Any degree of freedom (DOF) may be used.
2. The elements have constant stiffness, damping, and mass effects (e.g., material properties do not change with temperature).
3. Coupled-field elements using load-vector coupling and elements with Lagrange multipliers cannot be used.

15.6.2. Description of Analysis

A superelement (substructure) may be used in any analysis type. It simply represents a collection of elements that are reduced to act as one element. This one (super) element may then be used in the actual analysis (use pass) or be used to generate more superelements (generation or use pass). To reconstruct the detailed solutions (e.g., displacements and stresses) within the superelement, an expansion pass may be done. See the *Basic Analysis Guide* for loads which are applicable to a substructure analysis.

15.6.3. Statics

Consider the basic form of the static equations (Equation 15.1 (p. 762)):

$$[K]\{u\} = \{F\} \tag{15.108}$$

{F} includes nodal, pressure, and temperature effects. It does not include $\{F^{nr}\}$ (see Newton-Raphson Procedure (p. 711)). The equations may be partitioned into two groups, the master (retained) DOFs, here denoted by the subscript "m", and the slave (removed) DOFs, here denoted by the subscript "s".

$$\begin{bmatrix} [K_{mm}] & [K_{ms}] \\ [K_{sm}] & [K_{ss}] \end{bmatrix} \begin{Bmatrix} \{u_m\} \\ \{u_s\} \end{Bmatrix} = \begin{Bmatrix} \{F_m\} \\ \{F_s\} \end{Bmatrix} \tag{15.109}$$

or expanding:

$$[K_{mm}]\{u_m\} + [K_{ms}]\{u_s\} = \{F_m\} \tag{15.110}$$

$$[K_{sm}]\{u_m\} + [K_{ss}]\{u_s\} = \{F_s\} \tag{15.111}$$

The master DOFs should include all DOFs of all nodes on surfaces that connect to other parts of the structure. If accelerations are to be used in the use pass or if the use pass will be a transient analysis, master DOFs throughout the rest of the structure should also be used to characterize the distributed mass. Solving Equation 15.111 (p. 794) for $\{u_s\}$,

$$\{u_s\} = [K_{ss}]^{-1}\{F_s\} - [K_{ss}]^{-1}[K_{sm}]\{u_m\} \tag{15.112}$$

Substituting $\{u_s\}$ into Equation 15.110 (p. 794)

$$\left[[K_{mm}] - [K_{ms}][K_{ss}]^{-1}[K_{sm}]\right]\{u_m\} = \{F_m\} - [K_{ms}][K_{ss}]^{-1}\{F_s\} \tag{15.113}$$

or,

$$[\hat{K}]\{\hat{u}\} = \{\hat{F}\} \tag{15.114}$$

where:

$$[\hat{K}] = [K_{mm}] - [K_{ms}][K_{ss}]^{-1}[K_{sm}] \tag{15.115}$$

$$\{\hat{F}\} = \{F_m\} - [K_{ms}][K_{ss}]^{-1}\{F_s\} \tag{15.116}$$

$$\{\hat{u}\} = \{u_m\} \tag{15.117}$$

$[\hat{K}]$ and $\{\hat{F}\}$ are the superelement coefficient (e.g., stiffness) matrix and load vector, respectively.

In the preceding development, the load vector for the superelement has been treated as a total load vector. The same derivation may be applied to any number of independent load vectors, which in turn may be individually scaled in the superelement use pass. For example, the analyst may wish to apply thermal, pressure, gravity, and other loading conditions in varying proportions. Expanding the right-hand sides of Equation 15.110 (p. 794) and Equation 15.111 (p. 794) one gets, respectively:

$$\{F_m\} = \sum_{i=1}^{N}\{F_{mi}\} \tag{15.118}$$

$$\{F_s\} = \sum_{i=1}^{N}\{F_{si}\} \tag{15.119}$$

where:

N = number of independent load vectors.

Substituting into Equation 15.116 (p. 795):

$$\{\hat{F}\} = \sum_{i=1}^{N}\{F_{mi}\} - [K_{ms}][K_{ss}]^{-1}\sum_{i=1}^{N}\{F_{si}\} \tag{15.120}$$

To have independently scaled load vectors in the use pass, expand the left-hand side of Equation 15.120 (p. 795)

$$\{\hat{F}\} = \sum_{i=1}^{N}\{\hat{F}_i\} \tag{15.121}$$

Substituting Equation 15.121 (p. 795) into Equation 15.120 (p. 795) :

$$\{\hat{F}_i\} = \{F_{mi}\} - [K_{ms}][K_{ss}]^{-1}\{F_{si}\} \tag{15.122}$$

If the load vectors are scaled in the use pass such that:

$$\{\hat{F}\} = \sum_{i=1}^{N} b_i\{\hat{F}_i\} \tag{15.123}$$

where b_i is the scaling factor (*FACT* on the **LVSCALE** command), then Equation 15.112 (p. 794) becomes:

$$\{u_s\} = [K_{ss}]^{-1}\sum_{i=1}^{N} b_i\{F_{si}\} - [K_{ss}]^{-1}[K_{sm}]\{u_m\} \tag{15.124}$$

Equation 15.124 (p. 796) is used in the expansion pass to obtain the DOF values at the slave DOFs if the backsubstitution method is chosen (**SEOPT** command). If the resolve method is chosen for expansion pass, then the program will use Equation 15.109 (p. 794) to resolve for $\{u_s\}$. In doing so, the program makes $\{u_m\}$ as the internally prescribed displacement boundary conditions since $\{u_m\}$ are known in expansion pass. As the program treats DOFs associated with $\{u_m\}$ as displacement boundary conditions, the reaction forces by resolve method will be different from that computed at those master DOFs by the backsubstitution method. However, they are all in self-equilibrium satisfying Equation 15.109 (p. 794).

The above section Statics (p. 794) is equally applicable at an element level for elements with extra displacement shapes. The master DOFs become the nodal DOFs and the slave DOFs become the nodeless or extra DOFs.

15.6.4. Transients

The general form of the equations for transients is Equation 15.5 (p. 764) and Equation 15.38 (p. 775):

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F\} \tag{15.125}$$

For substructuring, an equation of the form:

$$[\hat{M}]\{\ddot{\hat{u}}\} + [\hat{C}]\{\dot{\hat{u}}\} + [\hat{K}]\{\hat{u}\} = \{\hat{F}\} \tag{15.126}$$

is needed. $[\hat{K}]$ and $\{\hat{F}\}$ are computed as they are for the static case (Equation 15.115 (p. 795) and Equation 15.116 (p. 795)). The computation of the reduced mass matrix is done by:

$$\begin{aligned}[\hat{M}] = {} & [M_{mm}] - [K_{ms}][K_{ss}]^{-1}[M_{sm}] - [M_{ms}][K_{ss}]^{-1}[K_{sm}] \\ & + [K_{ms}][K_{ss}]^{-1}[M_{ss}][K_{ss}]^{-1}[K_{sm}]\end{aligned} \tag{15.127}$$

This simplification was suggested by Guyan([14] (p. 921)) because direct partitioning and condensation are not practical (the condensed matrices would be functions of the time derivatives of displacement and very awkward to implement). The damping matrix is handled similarly:

$$\begin{aligned}[\hat{C}] = {} & [C_{mm}] - [K_{ms}][K_{ss}]^{-1}[C_{sm}] - [C_{ms}][K_{ss}]^{-1}[K_{sm}] \\ & + [K_{ms}][K_{ss}]^{-1}[C_{ss}][K_{ss}]^{-1}[K_{sm}]\end{aligned} \tag{15.128}$$

Equation 15.124 (p. 796) is also used to expand the DOF values to the slave DOFs in the transient case if the backsubstitution method is chosen. If the resolve method is chosen, the program will use Equation 15.109 (p. 794) and make $\{u_m\}$ as displacement boundary conditions the same way as the static expansion method does.

15.6.5. Component Mode Synthesis (CMS)

Component mode synthesis is an option used in substructure analysis (accessed with the **CMSOPT** command). It reduces the system matrices to a smaller set of interface DOFs between substructures and truncated sets of normal mode generalized coordinates (see Craig [344] (p. 940)).

For a damped system, each CMS substructure is defined by a stiffness matrix, a mass matrix, and a damping matrix. The matrix equation of motion is:

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F\} \tag{15.129}$$

Partitioning the matrix equation into interface and interior DOFs:

$$\{u\} = \begin{Bmatrix}\{u_m\}\\ \{u_s\}\end{Bmatrix}, [M] = \begin{bmatrix}[M_{mm}] & [M_{ms}]\\ [M_{sm}] & [M_{ss}]\end{bmatrix}, [C] = \begin{bmatrix}[C_{mm}] & [C_{ms}]\\ [C_{sm}] & [C_{ss}]\end{bmatrix}, [K] = \begin{bmatrix}[K_{mm}] & [K_{ms}]\\ [K_{sm}] & [K_{ss}]\end{bmatrix}, \{F\} = \begin{Bmatrix}\{F_m\}\\ \{F_s\}\end{Bmatrix} \tag{15.130}$$

where subscripts m and s refer to:

m = master DOFs defined only on interface nodes
s = all DOFs that are not master DOFs

The physical displacement vector, {u}, may be represented in terms of component generalized coordinates (see Craig [344] (p. 940)) as in Equation 15.131 (p. 797).

$$\{u\} = \begin{Bmatrix}\{u_m\}\\ \{u_s\}\end{Bmatrix} = [T]\begin{Bmatrix}\{u_m\}\\ \{y_\delta\}\end{Bmatrix} \tag{15.131}$$

where:

y_δ = truncated set of generalized modal coordinates
[T] = transformation matrix.

Fixed-Interface Method

For the fixed-interface method (see Craig and Bampton [345] (p. 940)), the transformation matrix has the form:

$$[T] = \begin{bmatrix}[I] & [0]\\ [G_{sm}] & [\Phi_s]\end{bmatrix} \tag{15.132}$$

where:

$[G_{sm}] = -[K_{ss}]^{-1}[K_{sm}]$ = redundant static constraint modes (see Craig and Bampton [345] (p. 940))
Φ_s = fixed-interface normal modes (eigenvectors obtained with interface nodes fixed)
[I] = identity matrix
[0] = null matrix

Free-Interface Method

For the free-interface method (see Hintz [409] (p. 944), Herting [412] (p. 944)), the transformation matrix has the form:

$$T = \begin{bmatrix}[I] & [0] & [0]\\ [G_{sm}] & [\Phi_{sr}] & [\hat{\Phi}_s]\end{bmatrix} \tag{15.133}$$

where:

$$[\hat{\Phi}_s] = [\Phi_s] - [G_{sm}][\Phi_m]$$

$[\Phi_m]$ = matrix of the master dof partition of the free-interface normal modes (eigenvectors obtained with interface dofs free)
$[\Phi_s]$ = matrix of the slave dof partition of the free-interface normal modes
$[\Phi_{sr}]$ = matrix of inertia relief modes
$[\Phi_{sr}]$ is included only if rigid body modes are present (see **CMSOPT**„„„FBDDEF). Any rigid body modes present are not included in $[\hat{\Phi}_s]$

$$[\Phi_{sr}] = -[K_{ss}]^{-1}[[M_{sm}]+[M_{ss}][G_{sm}]][\Psi_{mr}]$$

where:

$[\Psi_{mr}]$ = matrix of the master DOF partition of the rigid body modes

Residual Flexibility Free Interface Method

For the Residual Flexiblility Free interface (RFFB) method (see Martinez et al. [410] (p. 944)), the transformation matrix has the form:

$$T = \begin{bmatrix} [I] & [0] \\ [R_{sm}][R_{mm}]^{-1} & [\hat{\Phi}_s] \end{bmatrix} \tag{15.134}$$

where:

$[R_{sm}][R_{mm}]^{-1}$ = submatrices of the following residual vectors (see Residual Vector Method (p. 702)):

$$[R] = \begin{bmatrix} [R_{mm}] \\ [R_{sm}] \end{bmatrix}$$

$$[\hat{\Phi}_s] = [\Phi_s] - [R_{sm}][R_{mm}]^{-1}[\Phi_m]$$

Any rigid body modes present are included in $\begin{bmatrix} [\Phi_m] \\ [\Phi_s] \end{bmatrix}$

Transformation

After applying the transformation in Equation 15.131 (p. 797) into the matrix equation of motion Equation 15.129 (p. 797) , the equation of motion in the reduced space is obtained. The reduced stiffness, mass, and damping matrices and the reduced force vector of the CMS substructure will be:

$$[\hat{K}] = [T]^T[K][T] \tag{15.135}$$

$$[\hat{M}] = [T]^T[M][T] \tag{15.136}$$

$$[\hat{C}] = [T]^T[C][T] \tag{15.137}$$

$$\{\hat{F}\} = [T]^T\{F\} \tag{15.138}$$

In the reduced system, master DOFs will be used to couple the CMS superelement to other elements and/or CMS superelements.

For the fixed-interface method, if the fixed-interface normal modes are mass normalized, the reduced stiffness, mass, and damping matrices, and the reduced force vector have the final form:

$$[\hat{K}] = \begin{bmatrix} [K_{mm}] + [K_{ms}][G_{sm}] & [0] \\ [0] & [\Lambda^2] \end{bmatrix} \tag{15.139}$$

$$[\hat{M}] = \begin{bmatrix} [M_{mm}] + [M_{ms}][G_{sm}] + [G_{ms}][[M_{sm}] + [M_{ss}][G_{sm}]] & [[M_{ms}] + [G_{ms}][M_{ss}]][\Phi_s] \\ [\Phi_s]^T [[M_{sm}] + [M_{ss}][G_{sm}]] & [I] \end{bmatrix} \tag{15.140}$$

$$[\hat{C}] = \begin{bmatrix} [C_{mm}] + [C_{ms}][G_{sm}] + [G_{ms}][[C_{sm}] + [C_{ss}][G_{sm}]] & [[C_{ms}] + [G_{ms}][C_{ss}]][\Phi_s] \\ [\Phi_s]^T [[C_{sm}] + [C_{ss}][G_{sm}]] & [\Phi_s]^T [C_{ss}][\Phi_s] \end{bmatrix} \tag{15.141}$$

$$\{\hat{F}\} = \begin{Bmatrix} \{F_m\} + [G_{ms}]\{F_s\} \\ [\Phi_s]^T \{F_s\} \end{Bmatrix} \tag{15.142}$$

Where:

$[\Lambda^2]$ = a diagonal matrix containing the eigenvalues of retained fixed-interface normal modes.

Modal Damping Ratios

Modal damping ratios can be directly included in the reduced damping matrix $[\hat{C}]$ when the fixed-interface method is used (**ANTYPE**,SUBSTR with **CMSOPT**,FIX and **SEOPT**,,,3).The following terms are added to the diagonal of $[\Phi_s]^T[C_{ss}][\Phi_s]$ in Equation 15.141 (p. 799):

$$2\xi_i^m \omega_i$$

where:

ξ_i^m = modal damping ratio of the i^{th} fixed-interface normal mode (input with **MDAMP**)

ω_i = circular natural frequency of the i^{th} fixed-interface normal mode

Unsymmetric Matrices

The reduction of a skew-symmetric damping matrix due to the effect of gyroscopic damping (**CORIOLIS**,ON,,,ON) is supported. The damping matrix must contain the gyroscopic effect only. For all three methods, the modes used in the transformation basis are those of the conservative system. The resulting reduced damping matrix is also skew-symmetric.

15.7. Spectrum Analysis

Two types of spectrum analyses (**ANTYPE**,SPECTR) are supported: the deterministic response spectrum method and the nondeterministic random vibration method. Both excitation at the support and excitation away from the support are allowed. The three response spectrum methods are the single-point, multiple-point and dynamic design analysis method. The random vibration method uses the power spectral density (PSD) approach.

The following spectrum analysis topics are available:

15.7.1. Assumptions and Restrictions

1. The structure is linear.

2. For single-point response spectrum analysis (**SPOPT**,SPRS) and dynamic design analysis method (**SPOPT**,DDAM), the structure is excited by a spectrum of known direction and frequency components, acting uniformly on all support points or on specified unsupported master degrees of freedom (DOFs).

3. For multi-point response spectrum (**SPOPT**,MPRS) and power spectral density (**SPOPT**,PSD) analyses, the structure may be excited by different input spectra at different support points or unsupported nodes. Up to twenty different simultaneous input spectra are allowed.

15.7.2. Description of Analysis

The spectrum analysis capability is a separate analysis type (**ANTYPE**,SPECTR) and it must be preceded by a mode-frequency analysis. If mode combinations are needed, the required modes must also be expanded, as described in Mode-Frequency Analysis (p. 778).

The four options available are the single-point response spectrum method (**SPOPT**,SPRS), the dynamic design analysis method (**SPOPT**,DDAM), the random vibration method (**SPOPT**,PSD) and the multiple-point response spectrum method (**SPOPT**,MPRS). Each option is discussed in detail subsequently.

15.7.3. Single-Point Response Spectrum

Both excitation at the support (base excitation) and excitation away from the support (force excitation) are allowed for the single-point response spectrum analysis (**SPOPT**,SPRS). The table below summarizes these options as well as the input associated with each.

Table 15.3: Types of Spectrum Loading

	Excitation Option	
	Excitation at Support	**Excitation Away From Support**
Spectrum input	Response spectrum table (**FREQ** and **SV** commands)	Amplitude multiplier table (**FREQ** and **SV** commands)

	Excitation Option			
Orientation of load	Direction vector (input on **SED** and **ROCK** commands)		X, Y, Z direction at each node (selected by FX, FY, or FZ on **F** command)	
Distribution of loads	Constant on all support points		Amplitude in X, Y, or Z directions (selected by *VALUE* on **F** command)	
Type of input	Velocity	Acceleration	Displace-ment	Force
Response spec-trum type (KSV on **SVTYP** com-mand)	0	2	3,4	1

15.7.4. Damping

Damping is evaluated for each mode and is defined as in Equation 14.24 (p. 676).

Note that the material dependent damping contribution (**MP**,DMPR) is computed in the modal expansion phase with **MXPAND**,,,,YES, so that this damping contribution must be included there.

15.7.5. Participation Factors and Mode Coefficients

The participation factors for the given excitation direction are defined as:

$$\gamma_i = \{\phi\}_i^T [M]\{D\} \quad \text{for the base excitation option} \tag{15.143}$$

$$\gamma_i = \{\phi\}_i^T \{F\} \quad \text{for the force excitation option} \tag{15.144}$$

where:

γ_i = participation factor for the ith mode
$\{\phi\}_i$ = eigenvector normalized using Equation 15.51 (p. 779) (Nrmkey on the **MODOPT** command has no effect)
{D} = vector describing the excitation direction (see Equation 15.145 (p. 801))
{F} = input force vector

The vector describing the excitation direction has the form:

$$\{D\} = [T]\{e\} \tag{15.145}$$

where:

$$\{D\} = \left[D_1^a D_2^a D_3^a \ldots \right]^T$$

D_j^a = excitation at DOF j in direction a; a may be either X, Y, Z,or rotations about one of these axes

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 & (Z - Z_o) & -(Y - Y_o) \\ 0 & 1 & 0 & -(Z - Z_o) & 0 & (X - X_o) \\ 0 & 0 & 1 & (Y - Y_o) & -(X - X_o) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

X, Y, Z = global Cartesian coordinates of a point on the geometry
X_o, Y_o, Z_o = global Cartesian coordinates of point about which rotations are done (reference point)
{e} = six possible unit vectors

We can calculate the statically equivalent actions at j due to rigid-body displacements of the reference point using the concept of translation of axes [T] (Weaver and Johnston([279] (p. 936))).

For spectrum analysis, the D^a values may be determined in one of two ways:

1. For D values with rocking not included (based on the **SED** command):

$$D_X = \frac{S_X}{B} \tag{15.146}$$

$$D_Y = \frac{S_Y}{B} \tag{15.147}$$

$$D_Z = \frac{S_Z}{B} \tag{15.148}$$

where:

S_X, S_Y, S_Z = components of excitation direction (input as *SEDX*, *SEDY*, and *SEDZ*, respectively, on **SED** command)

$$B = \sqrt{(S_X)^2 + (S_Y)^2 + (S_Z)^2}$$

2. or, for D values with rocking included (based on the **SED** and **ROCK** command):

$$D_X = S_X + R_X \tag{15.149}$$

$$D_Y = S_Y + R_Y \tag{15.150}$$

$$D_Z = S_Z + R_Z \tag{15.151}$$

R is defined by:

$$\begin{Bmatrix} R_X \\ R_Y \\ R_Z \end{Bmatrix} = \begin{Bmatrix} C_X \\ C_Y \\ C_Z \end{Bmatrix} \times \begin{Bmatrix} r_X \\ r_Y \\ r_Z \end{Bmatrix} \tag{15.152}$$

where:

C_X, C_Y, C_Z = components of angular velocity components (input as *OMX*, *OMY*, and *OMZ*, respectively, on **ROCK** command)
x = vector cross product operator

$r_X = X^n - L_X$

$r_Y = Y^n - L_Y$

$r_Z = Z^n - L_Z$

X^n, Y^n, Z^n = coordinate of node n

L_X, L_Y, L_Z = location of center of rotation (input as *CGX*, *CGY*, and *CGZ* on **ROCK** command)

In a modal analysis, the ratio of each participation factor to the largest participation factor (output as RATIO) is printed out.

The displacement, velocity or acceleration vector for each mode is computed from the eigenvector by using a "mode coefficient":

$$\{r\}_i = \omega_i^m A_i \{\phi\}_i \tag{15.153}$$

where:

m = 0, 1, or 2, based on whether the displacements, velocities, or accelerations, respectively, are selected (using *label*, the third field on the mode combination commands **SRSS**, **CQC**, **GRP**, **DSUM**, **NRLSUM**, **ROSE**)

A_i = mode coefficient (see below)

The mode coefficient is computed in five different ways, depending on the type of excitation (**SVTYP** command).

1. For velocity excitation of base (**SVTYP**, 0)

$$A_i = \frac{S_{vi}\gamma_i}{\omega_i} \tag{15.154}$$

where:

S_{vi} = spectral velocity for the i^{th} mode (obtained from the input velocity spectrum at frequency f_i and effective damping ratio ξ_i')

f_i = i^{th} natural frequency (cycles per unit time = $\frac{\omega_i}{2\pi}$

ω_i = i^{th} natural circular frequency (radians per unit time)

2. For force excitation (**SVTYP**, 1)

$$A_i = \frac{S_{fi}\gamma_i}{\omega_i^2} \tag{15.155}$$

where:

S_{fi} = spectral force for the i^{th} mode (obtained from the input amplitude multiplier table at frequency f_i and effective damping ratio ξ_i').

3. For acceleration excitation of base (**SVTYP**, 2)

$$A_i = \frac{S_{ai}\gamma_i}{\omega_i^2} \tag{15.156}$$

where:

S_{ai} = spectral acceleration for the i^{th} mode (obtained from the input acceleration response spectrum at frequency f_i and effective damping ratio ξ_i').

4. For displacement excitation of base (**SVTYP**, 3)

$$A_i = S_{ui}\gamma_i \tag{15.157}$$

where:

S_{ui} = spectral displacement for the i^{th} mode (obtained from the input displacement response spectrum at frequency f_i and effective damping ratio ξ_i').

5. For power spectral density (PSD) (**SVTYP**, 4) (Vanmarcke([34] (p. 922)))

$$A_i = \frac{\gamma_i}{\omega_i^2}\left(S_{pi}\omega_i\left(\frac{\pi}{4\xi} - 1\right) + \int_0^{\omega_i} S_p d\omega\right)^{\frac{1}{2}} \tag{15.158}$$

where:

S_{pi} = power spectral density for the i^{th} mode (obtained from the input PSD spectrum at frequency f_i and effective damping ratio ξ_i')
ξ = damping ratio (input as *RATIO*, **DMPRAT** command, defaults to .01)

The integral in Equation 15.158 (p. 804) is approximated as:

$$\int_0^{\omega_i} S_p d\omega = \sum_{j=1}^{L_i} S_{pj}\Delta f \tag{15.159}$$

where:

$L_i = f_i$ (in integer form)
S_{pj} = power spectral density evaluated at frequency (f) equal to j (in real form)
Δf = effective frequency band for $f_i = 1$.

When S_{vi}, S_{fi}, S_{ai}, S_{ui}, or S_{pi} are needed between input frequencies, log-log interpolation is done in the space as defined.

The spectral values and the mode coefficients output in the "RESPONSE SPECTRUM CALCULATION SUMMARY" table are evaluated at the input curve with the lowest damping ratio, not at the effective damping ratio ξ_i'.

15.7.6. Combination of Modes

The modal displacements, velocity and acceleration (Equation 15.153 (p. 803)) may be combined in different ways to obtain the response of the structure. For all excitations but the PSD this would be the maximum response, and for the PSD excitation, this would be the 1-σ (standard deviation) relative response. The response includes DOF response as well as element results and reaction forces if computed in the expansion operations (*Elcalc* = YES on the **MXPAND** command).

In the case of the single-point response spectrum method (**SPOPT**,SPRS) or the dynamic-design analysis method (**SPOPT**,DDAM) options of the spectrum analysis , it is possible to expand only those modes whose significance factor exceeds the significant threshold value (SIGNIF value on **MXPAND** command). Note that the mode coefficients must be available at the time the modes are expanded.

Only those modes having a significant amplitude (mode coefficient) are chosen for mode combination. A mode having a coefficient of greater than a given value (input as SIGNIF on the mode combination commands **SRSS**, **CQC**, **GRP**, **DSUM**, **NRLSUM**, **ROSE** and **PSDCOM**) of the maximum mode coefficient (all modes are scanned) is considered significant.

The spectrum option provides the following options for the combination of modes. They are:

- Complete Quadratic Combination Method (p. 806)
- Grouping Method (p. 806)
- Double Sum Method (p. 807)
- Square Root of the Sum of the Squares (SRSS) Method (p. 807)
- Naval Research Laboratory Sum (NRL-SUM) Method (p. 807)NRL-SUM Method (NRLSUM)
- Closely Spaced Modes (CSM) Method (p. 808)
- Rosenblueth Method (p. 809)

These methods generate coefficients for the combination of mode shapes. This combination is done by a generalization of the method of the square root of the sum of the squares which has the form:

$$R_a = \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \varepsilon_{ij} R_i R_j \right)^{\frac{1}{2}} \tag{15.160}$$

where:

R_a = total modal response
N = total number of expanded modes
ε_{ij}= coupling coefficient. The value of ε_{ij} = 0.0 implies modes i and j are independent and approaches 1.0 as the dependency increases
$R_i = A_i \Psi_i$ = modal response in the i^{th} mode (Equation 15.153 (p. 803))
$R_j = A_j \Psi_j$ = modal response in the j^{th} mode
A_i = mode coefficient for the i^{th} mode
A_j = mode coefficient for the j^{th} mode
Ψ_i = the i^{th} mode shape

Ψ_j = the j^{th} mode shape

Ψ_i and Ψ_j may be the DOF response, reactions, or stresses. The DOF response, reactions, or stresses may be displacement, velocity or acceleration depending on the user request (*Label* on the mode combination commands **SRSS**, **CQC**, **DSUM**, **GRP**, **ROSE** or **NRLSUM**).

The mode combination instructions are written to `File.MCOM` by the mode combination command. Inputting this file in POST1 automatically performs the mode combination.

15.7.6.1. Complete Quadratic Combination Method

This method (accessed with the **CQC** command), is based on Wilson, et al.([65] (p. 924)).

$$R_a = \left(\left| \sum_{i=1}^{N} \sum_{j=i}^{N} k \varepsilon_{ij} R_i R_j \right| \right)^{\frac{1}{2}} \tag{15.161}$$

where:

$$k = \begin{cases} 1 \text{ if } i = j \\ 2 \text{ if } i \neq j \end{cases}$$

$$\varepsilon_{ij} = \frac{8(\xi_i' \xi_j')^{\frac{1}{2}} (\xi_i' + r\xi_j') r^{3/2}}{(1-r^2)^2 + 4\xi_i' \xi_j' r(1+r^2) + 4(\xi_i'^2 + \xi_j'^2) r^2}$$

$r = \omega_j / \omega_i$

15.7.6.2. Grouping Method

This method (accessed with the **GRP** command), is from the NRC Regulatory Guide([41] (p. 922)). For this case, Equation 15.160 (p. 805) specializes to:

$$R_a = \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \varepsilon_{ij} \left| R_i R_j \right| \right)^{\frac{1}{2}} \tag{15.162}$$

where:

$$\varepsilon_{ij} = \begin{cases} 1.0 \text{ if } \left| \dfrac{\omega_j - \omega_i}{\omega_i} \right| \leq 0.1 \\ 0.0 \text{ if } \left| \dfrac{\omega_j - \omega_i}{\omega_i} \right| > 0.1 \end{cases}$$

Closely spaced modes are divided into groups that include all modes having frequencies lying between the lowest frequency in the group and a frequency 10% higher. No one frequency is to be in more than one group.

15.7.6.3. Double Sum Method

The Double Sum Method (accessed with the **DSUM** command) also is from the NRC Regulatory Guide([41] (p. 922)). For this case, Equation 15.160 (p. 805) specializes to:

$$R_a = \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \varepsilon_{ij} \left| R_i R_j \right| \right)^{\frac{1}{2}} \tag{15.163}$$

where:

$$\varepsilon_{ij} = \frac{1}{1 + \left(\frac{\omega_i' - \omega_j'}{\xi_i'' \omega_i + \xi_j'' \omega_j} \right)^2}$$

ω_i' = damped natural circular frequency of the ith mode

ω_i = undamped natural circular frequency of the i^{th} mode

ξ_i'' = modified damping ratio of the i^{th} mode

The damped natural frequency is computed as:

$$\omega_i' = \omega_i (1 - (\xi_i')^2)^{\frac{1}{2}} \tag{15.164}$$

The modified damping ratio ξ_i'' is defined to account for the earthquake duration time:

$$\xi_i'' = \xi_i' + \frac{2}{t_d \omega_i} \tag{15.165}$$

where:

t_d = earthquake duration time, fixed at 10 units of time

15.7.6.4. Square Root of the Sum of the Squares (SRSS) Method

The SRSS method, accessed with the **SRSS** command, is taken from the NRC Regulatory Guide([41] (p. 922)). For this case, Equation 15.160 (p. 805) reduces to:

$$R_a = \left(\sum_{i=1}^{N} (R_i)^2 \right)^{\frac{1}{2}} \tag{15.166}$$

15.7.6.5. Naval Research Laboratory Sum (NRL-SUM) Method

The NRL-SUM method (O'Hara and Belsheim([107] (p. 926))), accessed with the **NRLSUM** command, calculates the maximum modal response as:

$$R_a = |R_{a1}| + \left(\sum_{i=2}^{N} (R_{ai})^2 \right)^{\frac{1}{2}} \tag{15.167}$$

where:

$|R_{a1}|$ = absolute value of the largest modal displacement, stress or reaction at the point
R_{ai} = displacement, stress or reaction contributions of the same point from other modes.

15.7.6.6. Closely Spaced Modes (CSM) Method

The CSM method, accessed with the **NRLSUM**,,,CSM command, calculates the combined effect of the modal responses of pairs of closely spaced modes (NAVSEA [411] (p. 944)). Modes are close if their frequencies are within 10% of the mean frequency. Once each CSM pair combination is determined, it is used in the NRL-SUM as a single effective mode. The modified NRL (or CSM) sum is:

$$R_a = \max\left(\left| R_i \right|_{i \neq j,k}, E_{\max} \right) + \sqrt{E_{\max}^2 + \sum_{i \neq j,k}^{N} R_i^2 - \left(\max\left(\left| R_i \right|_{i \neq j,k}, E_{\max} \right) \right)^2} \tag{15.168}$$

where

i ranges from 1 to N, excluding the closely spaced modes j and k.

E_{max} is the maximum amplitude of a mode pair response. It is computed as follows.

This procedure first requires a modal response correction to account for the effect of damping during the first quarter cycle:

$$C_i = R_i e^{\xi \frac{\pi}{2}} \tag{15.169}$$

where:

C_i = modal response of the i^{th} mode including the quarter cycle correction.
ξ = damping ratio (constant for all modes)
The closed form treatment can be expressed as the envelope of the sum of two decaying sinusoids (modes j and k). This closed form is a slightly conservative approach, as it is based on determining the peak of the envelope rather than the peak of the superposed values.

$$E(t) = e^{-at} \sqrt{\left(C_j + C_k \right)^2 - 4 C_j C_k \sin^2(dt)} \tag{15.170}$$

where:

E(*t*) = combined response of two closely spaced modes j and k

$$a = \xi \omega_m$$

$$\omega_m = \left(\frac{\omega_j + \omega_k}{2} \right)$$

$$d = 0.5 \sqrt{1 - \xi^2} \left| \omega_j - \omega_k \right|$$

The times at which this function reaches a minimum or a maximum are $t = 0$, and:

$$t_n = \frac{\sin^{-1}[S] - \theta}{2d} \tag{15.171}$$

where:

$$S = \frac{-a\left(C_j^2 + C_k^2\right)}{2C_j C_k \sqrt{a^2 + d^2}}$$

$$\theta = \tan^{-1}\left(a/d\right) \text{ and } 0 \le \theta \le \pi/2$$

Equation Equation 15.171 (p. 809) has multiple solutions only if S is smaller or equal to 1. If S is greater than 1, the CSM cannot be used to reduce the NRL sum.

15.7.6.7. Rosenblueth Method

The Rosenblueth Method ([375] (p. 942)) is accessed with the **ROSE** command.

The equations for the Double Sum method (above) apply, except for Equation 15.163 (p. 807). For the Rosenblueth Method, the sign of the modal responses is retained:

$$R_a = \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \varepsilon_{ij} R_i R_j \right)^{\frac{1}{2}} \tag{15.172}$$

15.7.7. Effective Mass and Cumulative Mass Fraction

The effective mass (output as EFFECTIVE MASS) for the i^{th} mode (which is a function of excitation direction) is (Clough and Penzien([80] (p. 925))):

$$M_{ei} = \frac{\gamma_i^2}{\{\phi\}_i^T [M]_i \{\phi\}_i} \tag{15.173}$$

Note from Equation 15.51 (p. 779) that

$$\{\phi\}_i^T [M] \{\phi\}_i = 1 \tag{15.174}$$

so that the effective mass reduces to γ_i^2. This does not apply to the force spectrum, for which the excitation is independent of the mass distribution.

The cumulative mass fraction for the *i*th mode is:

$$\widehat{M}_{ei} = \frac{\sum_{j=1}^{i} M_{ej}}{\sum_{j=1}^{N} M_{ej}} \tag{15.175}$$

where N is the total number of modes.

15.7.8. Dynamic Design Analysis Method

For the DDAM (Dynamic Design Analysis Method) procedure (**SPOPT**,DDAM) (O'Hara and Belsheim([107] (p. 926))), modal weights in thousands of pounds (kips) are computed from the participation factor:

$$w_i = \frac{g\gamma_i^2}{1000} \tag{15.176}$$

with

$$\gamma_i = \frac{\{\phi\}_i^T [M]\{D\}}{\bar{M}_i} \tag{15.177}$$

and

$$\bar{M}_i = \{\phi\}_i^T [M]\{\phi\}_i \tag{15.178}$$

where:

w_i = modal weight in kips

$\bar{M}_i$ = generalized mass of the i^{th} mode

g = acceleration due to gravity (386 in/sec^2)

$\{\phi\}_i$ = i^{th} mode shape normalized to unity

The mode coefficients are computed by:

$$A_i = \frac{S_{ai}\gamma_i}{\omega_i^2} \tag{15.179}$$

where:

S_{ai} = the greater of A_m or S_x

A_m = minimum acceleration (input as *AMIN* on the **ADDAM** command) defaults to 6g = 2316.0)

S_x = the lesser of Ag or $\omega_i V$

A = spectral acceleration

$$= \begin{cases} A_f A_a \dfrac{(A_b + w_i)(A_c + w_i)}{(A_d + w_i)^2} & \text{if } A_d \neq 0 \\ A_f A_a \dfrac{(A_b + w_i)}{(A_c + w_i)} & \text{if } A_d = 0 \end{cases}$$

V = spectral velocity

$$= V_f V_a \frac{(V_b + w_i)}{(V_c + w_i)}$$

A_f, A_a, A_b, A_c, A_d = acceleration spectrum computation constants (input as AF, AA, AB, AC, AD on the **ADDAM** command)

V_f, V_a, V_b, V_c = velocity spectrum computation constants (input as VF, VA, VB, VC on the **VDDAM** command)

The acceleration response, which is required for the nodal acceleration check (refer to Mode Selection Based on DDAM Procedure), is computed by:

$$\{\bar{R}_a\} = S_{ai}\gamma_i \{\Phi\}_i \tag{15.180}$$

where:

$\{\bar{R}_a\}$ = acceleration response for the i^{th} mode

The DDAM procedure is generally used with the NRL-SUM method of mode combination, which was described in Single-Point Response Spectrum (p. 800). Note that unlike Equation 15.51 (p. 779), O'Hara and Belsheim ([107] (p. 926)) normalize the mode shapes to the largest modal displacements. As a result, the NRL-1396 participation factors γ_i and mode coefficients A_i will be different.

15.7.9. Random Vibration Method

The random vibration method (**SPOPT**,PSD) allows multiple power spectral density (PSD) inputs (up to two hundred) in which these inputs can be:

1. full correlated,
2. uncorrelated, or
3. partially correlated.

The procedure is based on computing statistics of each modal response and then combining them. It is assumed that the excitations are stationary random processes.

15.7.10. Description of Method

For partially correlated nodal and base excitations, the complete equations of motions are segregated into the free and the restrained (support) DOF as:

$$\begin{bmatrix}[M_{ff}] & [M_{fr}]\\ [M_{rf}] & [M_{rr}]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}_f\}\\ \{\ddot{u}_r\}\end{Bmatrix}+\begin{bmatrix}[C_{ff}] & [C_{fr}]\\ [C_{rf}] & [C_{rr}]\end{bmatrix}\begin{Bmatrix}\{\dot{u}_f\}\\ \{\dot{u}_r\}\end{Bmatrix}+\begin{bmatrix}[K_{ff}] & [K_{fr}]\\ [K_{rf}] & [K_{rr}]\end{bmatrix}\begin{Bmatrix}\{u_f\}\\ \{u_r\}\end{Bmatrix}=\begin{Bmatrix}\{F\}\\ \{0\}\end{Bmatrix} \tag{15.181}$$

where $\{u_f\}$ are the free DOF and $\{u_r\}$ are the restrained DOF that are excited by random loading (unit value of displacement on **D** command). Note that the restrained DOF that are not excited are not included in Equation 15.181 (p. 811) (zero displacement on **D** command). {F} is the nodal force excitation activated by a nonzero value of force (on the **F** command). The value of force can be other than unity, allowing for scaling of the participation factors.

The free displacements can be decomposed into pseudo-static and dynamic parts as:

$$\{u_f\} = \{u_s\} + \{u_d\} \tag{15.182}$$

The pseudo-static displacements may be obtained from Equation 15.181 (p. 811) by excluding the first two terms on the left-hand side of the equation and by replacing $\{u_f\}$ by $\{u_s\}$:

$$\{u_s\} = -[K_{ff}]^{-1}[K_{fr}]\{u_r\} = [A]\{u_r\} \tag{15.183}$$

in which [A] = - $[K_{ff}]$-1$[K_{fr}]$. Physically, the elements along the i^{th} column of [A] are the pseudo-static displacements due to a unit displacement of the support DOFs excited by the ith base PSD. These dis-

placements are written as load step 2 on the .rst file. Substituting Equation 15.183 (p. 811) and Equation 15.182 (p. 811) into Equation 15.181 (p. 811) and assuming light damping yields:

$$[M_{ff}]\{\ddot{u}_d\} + [C_{ff}]\{\dot{u}_d\} + [K_{ff}]\{u_d\} \simeq \{F\} - ([M_{ff}][A] + [M_{fr}])\{\ddot{u}_r\} \quad (15.184)$$

The second term on the right-hand side of the above equation represents the equivalent forces due to support excitations.

Using the mode superposition analysis of Mode Superposition Method (p. 698) and rewriting Equation 14.100 (p. 698)) as:

$$\{u_d(t)\} = [\phi]\{y(t)\} \quad (15.185)$$

the above equations are decoupled yielding:

$$\ddot{y}_j + 2\xi_j\omega_j\dot{y}_j + \omega_j^2 y_j = G_j, \qquad (j = 1,2,3,\ldots,n) \quad (15.186)$$

where:

n = number of mode shapes chosen for evaluation (input as NMODE on **SPOPT** command)
y_j = generalized displacements
ω_j and ξ_j = natural circular frequencies and modal damping ratios

The modal loads G_j are defined by:

$$G_j = \{\Gamma_j\}^T\{\ddot{u}_r\} + \gamma_j \quad (15.187)$$

The modal participation factors corresponding to support excitation are given by:

$$\{\Gamma_j\} = -([M_{ff}][A] + [M_{fr}])^T\{\phi_j\} \quad (15.188)$$

and for nodal excitation:

$$\gamma_j = \{\phi_j\}^T\{F\} \quad (15.189)$$

Note that, for simplicity, equations for nodal excitation problems are developed for a single PSD table. Multiple nodal excitation PSD tables are, however, allowed in the program.

These factors are calculated (as a result of the **PFACT** action command) when defining base or nodal excitation cases and are written to the .psd file. Mode shapes $\{\phi_j\}$ should be normalized with respect to the mass matrix as in Equation 15.51 (p. 779).

The relationship between multiple input spectra are described in the later subsection, "Cross Spectral Terms for Partially Correlated Input PSD's".

15.7.11. Response Power Spectral Densities and Mean Square Response

Using the theory of random vibrations, the response PSD's can be computed from the input PSD's with the help of transfer functions for single DOF systems H(ω) and by using mode superposition techniques (**RPSD** command in POST26). The response PSD's for ith DOF are given by:

15.7.11.1. Dynamic Part

$$S_{d_i}(\omega) = \sum_{j=1}^{n}\sum_{k=1}^{n}\phi_{ij}\phi_{ik}\left(\sum_{\ell=1}^{r_1}\sum_{m=1}^{r_1}\gamma_{\ell j}\gamma_{mk}H_j^*(\omega)H_k(\omega)\bar{S}_{\ell m}(\omega) + \sum_{\ell=1}^{r_2}\sum_{m=1}^{r_2}\Gamma_{\ell j}\Gamma_{mk}H_j^*(\omega)H_k(\omega)\hat{S}_{\ell m}(\omega)\right) \quad (15.190)$$

15.7.11.2. Pseudo-Static Part

$$S_{s_i}(\omega) = \sum_{\ell=1}^{r_2}\sum_{m=1}^{r_2}A_{i\ell}A_{im}\left(\frac{1}{\omega^4}\hat{S}_{\ell m}(\omega)\right) \quad (15.191)$$

15.7.11.3. Covariance Part

$$S_{sd_i}(\omega) = \sum_{j=1}^{n}\sum_{\ell=1}^{r_2}\sum_{m=1}^{r_2}\phi_{ij}A_{i\ell}\left(-\frac{1}{\omega^2}\Gamma_{mj}H_j(\omega)\hat{S}_{\ell m}(\omega)\right) \quad (15.192)$$

where:

n = number of mode shapes chosen for evaluation (input as NMODE on **SPOPT** command)
r_1 and r_2 = number of nodal (away from support) and base PSD tables, respectively

The transfer functions for the single DOF system assume different forms depending on the type (*Type* on the **PSDUNIT** command) of the input PSD and the type of response desired (*Lab* and *Relkey* on the **PSDRES** command). The forms of the transfer functions for displacement as the output are listed below for different inputs.

1. Input = force or acceleration (FORC, ACEL, or ACCG on **PSDUNIT** command):

$$H_j(\omega) = \frac{1}{\omega_j^2 - \omega^2 + i(2\xi_j\omega_j\omega)} \quad (15.193)$$

2. Input = displacement (DISP on **PSDUNIT** command):

$$H_j(\omega) = \frac{\omega^2}{\omega_j^2 - \omega^2 + i(2\xi_j\omega_j\omega)} \quad (15.194)$$

3. Input = velocity (VELO on **PSDUNIT** command):

$$H_j(\omega) = \frac{i\omega}{\omega_j^2 - \omega^2 + i(2\xi_j\omega_j\omega)} \quad (15.195)$$

where:

ω = forcing frequency
ω_j = natural circular frequency for jth mode
$i = \sqrt{-1}$

Now, random vibration analysis can be used to show that the absolute value of the mean square response of the i^{th} free displacement (ABS option on the **PSDRES** command) is:

$$\sigma_{f_i}^2 = \int_0^\infty S_{d_i}(\omega)d\omega + \int_0^\infty S_{s_i}(\omega)d\omega + 2\left|\int_0^\infty S_{sd_i}(\omega)d\omega\right|_{Re} \quad (15.196)$$
$$= \sigma_{d_i}^2 + \sigma_{s_i}^2 + 2C_v(u_{s_i}, u_{d_i})$$

where:

| |Re = denotes the real part of the argument

$\sigma_{d_i}^2$ = variance of the ith relative (dynamic) free displacements (REL option on the **PSDRES** command)

$\sigma_{s_i}^2$ = variance of the ith pseudo-static displacements

C_v (u_{s_i} , u_{d_i}) = covariance between the static and dynamic displacements

The general formulation described above gives simplified equations for several situations commonly encountered in practice. For fully correlated nodal excitations and identical support motions, the subscripts ℓ and m would drop out from the Equation 15.190 (p. 813) thru Equation 15.192 (p. 813). When only nodal excitations exist, the last two terms in Equation 15.196 (p. 814) do not apply, and only the first term within the large parentheses in Equation 15.190 (p. 813) needs to be evaluated. For uncorrelated nodal force and base excitations, the cross PSD's (i.e. $\ell \neq$ m) are zero, and only the terms for which ℓ = m in Equation 15.190 (p. 813) thru Equation 15.192 (p. 813) need to be considered.

Equation 15.190 (p. 813) thru Equation 15.192 (p. 813) can be rewritten as:

$$S_{d_i}(\omega) = \sum_{j=1}^{n}\sum_{k=1}^{n}\phi_{ij}\phi_{ik}R_{jk}(\omega) \quad (15.197)$$

$$S_{s_i}(\omega) = \sum_{\ell=1}^{r_2}\sum_{m=1}^{r_2}A_{i\ell}A_{im}\bar{R}_{\ell m}(\omega) \quad (15.198)$$

$$S_{sd_i}(\omega) = \sum_{j=1}^{n}\sum_{\ell=1}^{r_2}\phi_{ij}A_{i\ell}\hat{R}_{j\ell}(\omega) \quad (15.199)$$

where:

$R_{jk}(\omega), \bar{R}_{\ell m}(\omega), \hat{R}_{j\ell}(\omega)$ = modal PSD's, terms within large parentheses of Equation 15.190 (p. 813) thru Equation 15.192 (p. 813)

Closed-form solutions for piecewise linear PSD in log-log scale are employed to compute each integration in Equation 15.196 (p. 814) (Chen and Ali([193] (p. 931)) and Harichandran([194] (p. 931))) .

Subsequently, the variances become:

$$\sigma_{d_i}^2 = \sum_{j=1}^{n}\sum_{k=1}^{n}\phi_{ij}\phi_{ik}Q_{jk} \quad (15.200)$$

$$\sigma_{s_i}^2 = \sum_{\ell=1}^{r_2}\sum_{m=1}^{r_2} A_{i\ell}A_{im}\bar{Q}_{\ell m} \tag{15.201}$$

$$\sigma_{sd_i}^2 = \sum_{j=1}^{n}\sum_{\ell=1}^{r_2} \phi_{ij}A_{i\ell}\hat{Q}_{j\ell} \tag{15.202}$$

The modal covariance matrices $Q_{jk}, \bar{Q}_{\ell m}$, and $\hat{Q}_{j\ell}$ are available in the .psd file. Note that Equation 15.200 (p. 814) thru Equation 15.202 (p. 815) represent mode combination (**PSDCOM** command) for random vibration analysis.

The variance for stresses, nodal forces or reactions can be computed (*Elcalc* = YES on **SPOPT** (if *Elcalc* = YES on **MXPAND**)) from equations similar to Equation 15.200 (p. 814) thru Equation 15.202 (p. 815). If the stress variance is desired, replace the mode shapes (ϕ_{ij}) and static displacements $(A_{i\ell})$ with mode stresses $(\bar{\phi}_{ij})$ and static stresses $(\bar{A}_{i\ell})$. Similarly, if the node force variance is desired, replace the mode shapes and static displacements with mode nodal forces $(\hat{\phi}_{ij})$ and static nodal forces $(\hat{A}_{i\ell})$. Finally, if reaction variances are desired, replace the mode shapes and static displacements with mode reaction $(\tilde{\phi}_{ij})$ and static reactions $(\tilde{A}_{i\ell})$. Furthermore, the variances of the first and second time derivatives (VELO and ACEL options respectively on the **PSDRES** command) of all the quantities mentioned above can be computed using the following relations:

$$S_{\dot{u}}(\omega) = \omega^2 S_u(\omega) \tag{15.203}$$

$$S_{\ddot{u}}(\omega) = \omega^4 S_u(\omega) \tag{15.204}$$

Finally, the square root of Equations Equation 15.200 (p. 814), Equation 15.201 (p. 815), and Equation 15.202 (p. 815) are taken to yield the 1-σ results (i.e., the standard deviation) for the i^{th} displacement (or strain, stress, or force) quantity, and the they are written to the `.rst` file.

15.7.11.4. Equivalent Stress Mean Square Response

The equivalent stress (SEQV) mean square response is computed as suggested by Segalman et al([354] (p. 940)) as:

$$(\bar{\sigma}_d^2)_{nd} = \sum_{j=1}^{n}\sum_{k=1}^{n} (\Psi)_{nd}^T A (\Psi_k)_{nd} Q_{jk} \tag{15.205}$$

where:

$(\Psi_j)_{nd}$ = vector of component "stress shapes" for mode j at node nd

$$[A] = \begin{bmatrix} 1 & -1/2 & -1/2 & 0 & 0 & 0 \\ -1/2 & 1 & -1/2 & 0 & 0 & 0 \\ -1/2 & -1/2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix} = \text{quadratic operator}$$

Note that the probability distribution for the equivalent stress is neither Gaussian nor is the mean value zero. However, the"3-σ" rule (multiplying the RMS value by 3) yields a conservative estimate on the upper bound of the equivalent stress (Reese et al([355] (p. 941))). Since no information on the distribution of the principal stresses or stress intensity (S1, S2, S3, and SINT) is known, these values are set to zero.

15.7.12. Cross Spectral Terms for Partially Correlated Input PSDs

For excitation defined by more than a single input PSD, cross terms which determine the degree of correlation between the various PSDs are defined as:

$$[S(\omega)] = \begin{bmatrix} S_{11}(\omega) & C_{12}(\omega)+iQ_{12}(\omega) & C_{13}(\omega)+iQ_{13}(\omega) \\ C_{12}(\omega)-iQ_{12}(\omega) & S_{22}(\omega) & C_{23}(\omega)+iQ_{23}(\omega) \\ C_{13}(\omega)-iQ_{13}(\omega) & C_{23}(\omega)-iQ_{23}(\omega) & S_{33}(\omega) \end{bmatrix} \quad (15.206)$$

where:

$S_{nn}(\omega)$ = input PSD spectra which are related. (Defined by the **PSDVAL** command and located as table number (*TBLNO*) n)
$C_{nm}(\omega)$ = cospectra which make up the real part of the cross terms. (Defined by the **COVAL** command where n and m (*TBLNO1* and *TBLNO2*) identify the matrix location of the cross term)
$Q_{nm}(\omega)$ = quadspectra which make up the imaginary part of the cross terms. (Defined by the **QDVAL** command where n and m (*TBLNO1* and *TBLNO2*) identify the matrix location of the cross term)

The normalized cross PSD function is called the coherence function and is defined as:

$$\gamma^2_{nm}(\omega) = \frac{\left| C_{nm}(\omega) - iQ_{nm}(\omega) \right|^2}{S_{nn}(\omega) S_{mm}(\omega)} \quad (15.207)$$

where: $0 \le \gamma^2_{nm}(\omega) \le 1$

Although the above example demonstrates the cross correlation for 3 input spectra, this matrix may range in size from 2 x 2 to 200 x 200 (i.e., maximum number of tables is 200).

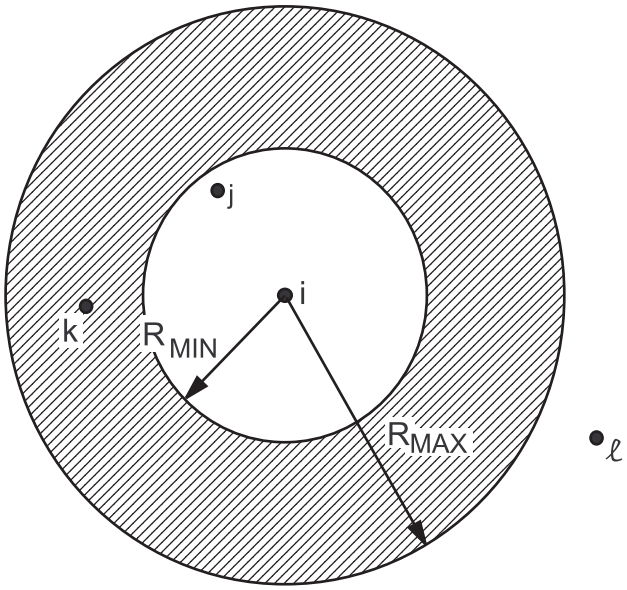
For the special case in which all cross terms are zero, the input spectra are said to be uncorrelated. Note that correlation between nodal and base excitations is not allowed.

15.7.13. Spatial Correlation

The degree of correlation between excited nodes may also be controlled. Depending upon the distance between excited nodes and the values of R_{MIN} and R_{MAX} (input as RMIN and RMAX on the **PSDSPL** command), an overall excitation PSD can be constructed such that excitation at the nodes may be un-

correlated, partially correlated or fully correlated. If the distance between excited nodes is less than R_{MIN}, then the two nodes are fully correlated; if the distance is greater than R_{MAX}, then the two nodes are uncorrelated; if the distance lies between R_{MIN} and R_{MAX}, excitation is partially correlated based on the actual distance between nodes. The following figure indicates how R_{MIN}, R_{MAX} and the correlation are related. Spatial correlation between excited nodes is not allowed for a pressure PSD analysis (**PSDUNIT**,PRES).

Figure 15.7: Sphere of Influence Relating Spatially Correlated PSD Excitation



Node i excitation is fully correlated with node j excitation
Node i excitation is partially correlated with node k excitation
Node i excitation is uncorrelated with node ℓ excitation

For two excitation points 1 and 2, the PSD would be:

$$[S(\omega)] = S_o(\omega)\begin{bmatrix} 1 & \alpha_{12} \\ \alpha_{12} & 1 \end{bmatrix} \tag{15.208}$$

where:

$$\alpha_{12} = \begin{cases} \dfrac{R_{MAX} - D_{12}}{R_{MAX} - R_{MIN}} & \text{if } R_{MIN} < D_{12} < R_{MAX} \\ 1 & \text{if } D_{12} \le R_{MIN} \\ 0 & \text{if } D_{12} \ge R_{MAX} \end{cases}$$

D_{12} = distance between the two excitation points 1 and 2
$S_o(\omega)$ = basic input PSD (**PSDVAL** and **PSDFRQ** commands)

15.7.14. Wave Propagation

To include wave propagation effects of a random loading, the excitation PSD is constructed as:

$$S_{\ell m}(\omega) = S_o(\omega)(e^{-i\omega d_{\ell m}}) \tag{15.209}$$

where:

$d_{\ell m} = \frac{\lfloor D_{\ell m} \rfloor \{V\}}{V^2}$ = delay

$\{D_{\ell m}\} = \{x_m\} - \{x_\ell\}$ = separation vector between excitations points ℓ and m

{V} = velocity of propagation of the wave (input as VX, VY and VZ on **PSDWAV** command)

$\{x_\ell\}$ = nodal coordinates of excitation point ℓ

More than one simultaneous wave or spatially correlated PSD inputs are permitted, in which case the input excitation [S(ω)] reflects the influence of two or more uncorrelated input spectra. In this case, partial correlation among the basic input PSD's is not currently permitted. Wave propagation effects are not allowed for a pressure PSD analysis (**PSDUNIT**,PRES).

15.7.15. Multi-Point Response Spectrum Method

The response spectrum analysis due to multi-point support and nodal excitations (**SPOPT**,MPRS) allows up to three hundred different excitations (**PFACT** command) based on up to two hundred different spectrum tables (**SPUNIT**, **SPFREQ** and **SPVAL** commands). The input spectrum are assumed to be uncorrelated to each other.

Most of the ingredients for performing multi-point response spectrum analysis are already developed in the previous subsection of the random vibration method. As with the PSD analysis, the static shapes corresponding to equation Equation 15.183 (p. 811) for base excitation are written as load step #2 on the `*.rst` file, Assuming that the participation factors, $\Gamma_{j\ell}$, for the ℓ^{th} input spectrum table have already been computed (by Equation 15.188 (p. 812), for example), the mode coefficients for the ℓ^{th} table are obtained as:

$$B_{j\ell} = \Gamma_{j\ell} S_{j\ell} \tag{15.210}$$

where:

$S_{j\ell}$ = interpolated input response spectrum for the ℓ^{th} table at the j^{th} natural frequency (defined by the **SPFREQ**, **SPVAL** and **SPUNIT** commands)

For each input spectrum, the mode shapes, mode stresses, etc. are multiplied by the mode coefficients to compute modal quantities, which can then be combined with the help of any of the available mode combination techniques (SRSS, CQC, Double Sum, Grouping, NRL-SUM, or Rosenblueth method), as described in the previous section on the single-point response spectrum method. The Absolute Sum method (AbsSumKey = yes on the SRSS command) can also be used as described in the section that follows.

Finally, the response of the structure is obtained by combining the responses to each spectrum using the SRSS method.

The mode combination instructions are written to the file `Jobname.MCOM` by the mode combination command. Inputting the file in POST1 (**/INPUT** command) automatically performs the mode combination.

15.7.15.1. Absolute Sum Combination Method

In a multi-point response spectrum analysis with base excitation, the square root of sum of square (SRSS) combination method does not take into account the phase relationship between the modal responses when same direction of excitation is applied at each support. The Absolute Sum combination method [[420] (p. 944)] takes into account that the peak values for one excitation direction are occurring at the same time so that the absolute values are summed. The remaining combination of modes and directions is performed with the SRSS method.

The total response with missing mass included is expressed as:

$$R_a = \sqrt{\sum_{n_d=1}^{ND}\left\{\sum_{j=1}^{N}\left(\sum_{n_s=1}^{NS}\left|R_{jn_dn_s}\right|\right)^2 + \left(\sum_{n_s=1}^{NS}\left|R_{Mn_dn_s}\right|\right)^2\right\}} \quad (15.211)$$

Where:

n_d is the excitation direction index. The excitation direction is specified using *SEDX*, *SEDY*, or *SEDZ* on the **SED** command.

ND is the total number of excitation directions.

n_s is the support (node group) index. The node group is defined with a component based on nodes (*Cname* on the **SED** command).

NS is the total number of supports.

$R_{jn_dn_s}$ is the modal response for the jth mode, the n_dth direction, and the n_sth support.

$R_{Mn_dn_s}$ is the missing mass response for the n_dth direction, and the n_sth support (**MMASS** command). It is optional.

Note

All supports are not necessarily excited in all directions. For example, if the n_sth support is not excited in the n_dth direction then Rjn_dn_s is zero for all modes j.

15.7.16. Missing-Mass Response

The spectrum analysis is based on a mode-superposition approach where the responses of the higher modes are neglected; therefore, part of the mass of the structure is missing in the dynamic analysis. The missing-mass response method ([374] (p. 942)) permits inclusion of the missing mass effect in a single-point response spectrum (**SPOPT**, SPRS) or multiple-point response spectrum analysis (**SPOPT**,MPRS) when base excitation is considered

Considering a rigid structure, the inertia force due to ground acceleration is:

$$\{F_T\} = -[M]\{D\}S_{a0} \quad (15.212)$$

where:

$\{F_T\}$ = total inertia force vector

S_{a0} = spectrum acceleration at zero period (also called the ZPA value), input as ZPA on the **MMASS** command.

Mode superposition can be used to determine the inertia force. For mode j, the modal inertia force is:

$$\{F_j\} = -[M]\{\phi\}_j \ddot{y}_j \tag{15.213}$$

where:

$\{F_j\}$ = modal inertia force for mode *j*.

Using equations Equation 15.153 (p. 803) and Equation 15.156 (p. 804), this force can be rewritten:

$$\{F_j\} = -[M]\{\phi\}_j \gamma_j S_{a0} \tag{15.214}$$

The missing inertia force vector is then the difference between the total inertia force given by Equation 15.212 (p. 819) and the sum of the modal inertia forces defined by Equation 15.214 (p. 820):

$$\{F_M\} = \{F_T\} - \sum_{j=1}^{N}\{F_j\} = [M]\left(\sum_{j=1}^{N}\{\phi\}_j \gamma_j - \{D\}\right)S_{a0} \tag{15.215}$$

The expression within the parentheses in the equation above is the fraction of degree of freedom mass missing:

$$\{e\} = \sum_{j=1}^{N}\{\phi\}_j \gamma_j - \{D\} \tag{15.216}$$

The missing mass response is the static shape due to the inertia forces defined by equation :

$$\{R_M\} = [K]^{-1}\{F_M\} \tag{15.217}$$

where:

$\{R_M\}$ is the missing mass response

The application of these equations can be extended to flexible structures because the higher truncated modes are supposed to be mostly rigid and exhibit pseudo-static responses to an acceleration base excitation.

In a single-point response spectrum analysis, the missing mass response is written as load step 2 in the `*.rst` file. In a multiple-point response spectrum analysis, it is written as load step 3.

Combination Method

Because the missing-mass response is a pseudo-static response, it is in phase with the imposed acceleration but out of phase with the modal responses; therefore, the missing-mass response and the modal responses defined in Equation 15.160 (p. 805) are combined using the square root of sum of the squares (SRSS) method.

The total response including the missing mass effect is:

$$R_a = \sqrt{\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\varepsilon_{ij}R_iR_j\right) + (R_M)^2} \quad (15.218)$$

15.7.17. Rigid Responses

For frequencies higher than the amplified acceleration region of the spectrum, the modal responses consist of both periodic and rigid components. The rigid components are considered separately because the corresponding responses are all in phase. The combination methods listed in Combination of Modes (p. 805) do not apply

The rigid component of a modal response is expressed as:

$$R_{ri} = \alpha_i R_i \quad (15.219)$$

where:

R_{ri} = the rigid component of the modal response of mode *i*

α_i = rigid response coefficient in the range of values 0 through 1. See the Gupta and Lindley-Yow methods below.

R_i = modal response of mode *i*

The corresponding periodic component is then:

$$R_{pi} = \sqrt{(1-(\alpha_i)^2}R_i \quad (15.220)$$

where:

R_{pi} = periodic component of the modal response of mode *i*

Two methods ([375] (p. 942)) can be used to separate the periodic and the rigid components in each modal response. Each one has a different definition of the rigid response coefficients α_i.

Gupta Method

$$\alpha_i = \frac{\log\left(\frac{F_i}{F_1}\right)}{\log\left(\frac{F_2}{F_1}\right)} \quad (15.221)$$

$\alpha_i = 0$ for $F_i \leq F_1$

$\alpha_i = 1$ for $F_i \geq F_2$

where:

Fi = ith frequency value.

F1 and F2 = key frequencies. F1 is input as Val1 and F2 is input as Val2 on **RIGRESP** command with Method = GUPTA.

Lindley-Yow Method

$$\alpha_i = \frac{S_{a0}}{S_{ai}} \tag{15.222}$$

where:

S_{a0} = spectrum acceleration at zero period (ZPA). It is input as ZPA on **RIGRESP** command with Method = LINDLEY

S_{ai} = spectrum acceleration corresponding to the ith frequency

Combination Method

The periodic components are combined using the square root of sum of squares (SRSS), the complete quadratic (CQC) or the Rosenblueth (ROSE) combination methods.

Since the rigid components are all in phase, they are summed algebraically. When the missing mass response (accessed with **MMASS** command) is included in the analysis, since it is a rigid response as well, it is summed with those components. Finally, periodic and rigid responses are combined using the SRSS method.

The total response with the rigid responses and the missing mass response included is expressed as:

$$R_a = \sqrt{\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\varepsilon_{ij}\sqrt{1-(\alpha_i)^2}R_i\sqrt{1-(\alpha_j)^2}R_j\right)+\left(\sum_{i=1}^{N}\alpha_i R_i + R_M\right)^2} \tag{15.223}$$

15.8. Linear Perturbation Analysis

The following linear perturbation analysis topics are available:

15.8.1. Assumptions and Restrictions

The following assumptions and restrictions apply to the linear perturbation method:

- Valid for structural degrees of freedom (DOFs) only.
- Supports only certain structural element types. See Elements Under Linear Perturbation in the *Element Reference* for a complete list.
- Performed by first running a linear or nonlinear static or full (**TRNOPT**,FULL) transient analysis, then restarting the analysis with the appropriate linear perturbation analysis type.

- For a linear perturbation modal analysis, the mass density and damping information should be supplied in the analysis phase of the static or full transient procedure (if they are needed later for the linear perturbation analysis).

15.8.2. Description of Analysis

In many engineering applications, the linear behavior of a structure based on a prior linear or nonlinear preloaded status is of interest. The linear perturbation analysis procedure is designed to solve a linear problem from this preloaded case. Typically, the Newton-Raphson procedure is used in the nonlinear analysis (see Figure 14.9: Newton-Raphson Solution - One Iteration (p. 713)). Without loss of generality, the nonlinear static problem is discussed here. A similar process can apply to linear static and linear or nonlinear full transient analyses.

In the case of a nonlinear static analysis, the mode frequency of the structure based on load level $\{F^a\}$ is needed. The tangent matrix $\left[K_i^T\right]$ can be used in the linear perturbation analysis in order to obtain the effect of preload, as the linear stiffness matrix without preloading does not give the correct solution.

Equation 14.147 (p. 711) at iteration *i* is recast here for convenience of discussion:

$$[K_i^T]\{\Delta u_i\} = \{F^a\} - \{F_i^{nr}\} \tag{15.224}$$

Here, $\left[K_i^T\right]$ is the global tangent matrix which can be symbolically segregated into other matrices as follows:

$$\left[K_i^T\right]=\left[K_i^M\right]+\left[S_i^N\right]+\left[K_i^{LD}\right]+\left[K_i^C\right]+\left[K_i^{\mathrm{SP}}\right] \tag{15.225}$$

where:

$\left[K_i^M\right]$ = the part of the tangent stiffness contributed from the material property

$\left[S_i^N\right]$ = the stress stiffening matrix introduced by non-zero stresses from the structure; superscript N indicates nonlinear (that is, the stress stiffening matrix is obtained from a nonlinear analysis – not to be confused with the linear stress stiffening matrix obtained from the first phase of a linear perturbation buckling analysis)

$\left[K_i^{LD}\right]$ = the load stiffening matrix introduced by external pressure loads or by follower force effect (element FOLLW201)

$\left[K_i^C\right]$ = the total stiffness matrix contributed from contact elements of the model

$\left[K_i^{\mathrm{SP}}\right]$ = the spin-softening matrix introduced by rotational velocities

In the preload (or base) analysis, $\left[K_i^M\right]$ could be generally nonlinear. In the subsequent linear perturbation analysis, the material behavior must be linear. Depending on the material option specified on the **PERTURB** command and the material properties used, the program handles materials in the linear perturbation analysis differently. The program uses the consistent material Jacobian of hyperelastic materials at iteration *i* as the material behavior in the linear perturbation analysis. For other nonlinear materials, the program uses the linear portion of the material in the linear perturbation analysis by default.

For any linear materials, the same behaviors are assumed in both the base and the linear perturbation analyses. For nonlinear materials, an option is available to use the consistent material tangent Jacobian (see the **PERTURB** command for more information).

In the case of contact analysis from the static or full transient runs, $\left[K_i^C\right]$ is by default the consistent tangent (total) stiffness matrix from the contact elements based on their status at iteration *i*. However, you are optionally allowed to change contact behavior and, hence, contact stiffness for all contact pairs by using the **PERTURB** command in the linear perturbation analysis or the **CNKMOD** command for individual contact pairs. All contact behavior can be changed into the following form (similar to KEYOPT(12) in the contact element descriptions; for example, CONTA174) from whatever behavior it was in the previous static or full transient analysis to:

1 -- Rough
2 -- No separation (sliding permitted)
3 -- Bonded
4 -- No separation (always)

15.8.3. Static Analysis Based on Linear Perturbation

A static analysis can be performed by using the linear perturbation analysis procedure. The effect of the structure from the previous static or full transient analysis is included.

The theoretical steps can be summarized as follows:

1. Perform a static or full transient base analysis; use the **RESCONTROL** command to create the restart files for the load points of interest (i.e., at various substeps of a load step).
2. Restart the analysis at the load point of interest and issue the commands **PERTURB**,STATIC and **SOLVE**,ELFORM. The program regenerates the total tangent stiffness matrix $\left[K_i^T\right]$ from the restart point.
3. For the static analysis, the perturbation load is required and must be defined or modified at this step (see Application of Perturbation Loads (p. 828) and also the **PERTURB** command description).
4. Upon the onset of the second **SOLVE** command, the nodal coordinates are updated automatically if the base analysis includes large-deflection effects (**NLGEOM**,ON). The program then calculates the static solution using the following the equation (similar to Equation 15.1 (p. 762)):
$$\left[K_i^T\right]\left\{U_{perturbed}\right\}=\left\{F_{perturbed}\right\} \tag{15.226}$$
5. Along with the static solution phase, a stress calculation is carried out based on the material definition specified via the **PERTURB** command. (See Interpretation of Structural Element Results After a Linear Perturbation Analysis in the *Element Reference* for details.)

15.8.4. Modal Analysis Based on Linear Perturbation

A modal analysis can be performed by using the linear perturbation analysis procedure. The effect of the structure from the previous static or full transient analysis is included. The theoretical steps can be summarized as follows:

1. Perform a static or full transient base analysis; use the **RESCONTROL** command to create the restart files for the load points of interest (i.e., at various substeps of a load step).

2. Restart the analysis at the load point of interest and issue the commands **PERTURB**,MODAL and **SOLVE**,ELFORM. The program regenerates the total tangent stiffness matrix $\left[K_i^T\right]$ from the restart point.

3. If a downstream analysis is desired using the load vector from the modal analysis, the perturbation load can be defined or modified here (see Application of Perturbation Loads (p. 828) and also the **PERTURB** command description) for the downstream analyses.

4. Upon the onset of the second **SOLVE** command, the nodal coordinates are updated automatically if large-deflection was included in the base analysis (**NLGEOM**,ON). The program then calculates the eigensolution using the following equation (similar to Equation 14.192 (p. 727)):

$$\left[K_i^T\right]\left\{\phi_j\right\}=\lambda_j[M]\left\{\phi_j\right\} \tag{15.227}$$

   where:

   $\left[K_i^T\right]$ = total tangent matrix from the load point of interest
   $\{\phi_j\}$ = eigenvector
   [M] = structural mass matrix

   The structural material behavior in a linear perturbation analysis is discussed above and can be controlled by the **PERTURB** command. The UNSYM and DAMP eigensolvers can be used as well. In the case of the DAMP eigensolution, the following equation (similar to Equation 14.204 (p. 732)) is solved:

$$\left[K_i^T\right]\left\{\phi_j\right\}+\lambda_j[C]\left\{\phi_j\right\}=-\lambda_j^2[M]\left\{\phi_j\right\} \tag{15.228}$$

   where [C] = structural damping matrix.

5. Along with the modal solution phase, a stress expansion pass is carried out. (See Interpretation of Structural Element Results After a Linear Perturbation Analysis in the *Element Reference* for details.) To use the appropriate material property and to obtain the total sum of elastic strain/stress due to the linear perturbation analysis and the base analysis, a stress expansion pass must be done along with the modal analysis. A separate expansion pass (**EXPASS** command) is not allowed after the linear perturbation analysis.

15.8.5. Eigenvalue Buckling Analysis Based on Linear Perturbation

An eigenvalue buckling analysis can be performed by using the linear perturbation analysis procedure. Unlike the preceding modal analysis description, this procedure is more complicated since the perturbation load is a prerequisite for calculating the linearly perturbed stresses; then the perturbed stresses are used to calculate the linearly perturbed stress stiffening matrix before the buckling analysis can be started. Physically, the total perturbation load multiplied by the buckling load factor plus the total loads available from the restart load point is used to determine the ultimate buckling loads from the analysis.

If the base analysis is linear, however, the total perturbation load multiplied by the buckling load factor represents the ultimate buckling load. This procedure is similar to the linear eigenvalue buckling analysis outlined in Buckling Analysis (p. 792). It can be viewed as restarting the analysis with the tangent

matrix $\left[K_i^T\right]$ equal to the linear stiffness matrix, and load vector $\{F_{restart}\} = \{0\}$. The prediction of the buckling load can be written:

$$\left\{F_{buckling}\right\} = \{0\} + \lambda_j \left\{F_{perturbed}\right\} \tag{15.229}$$

The theoretical steps for a linear perturbation buckling analysis with a nonlinear base analysis can be summarized as follows:

1. Perform a static or full transient base analysis; use the **RESCONTROL** command to create the restart files for the load points of interest (i.e., at various substeps of a load step).
2. Restart the analysis at the load point of interest and issue the commands **PERTURB**,BUCKLE and **SOLVE**,ELFORM. The program regenerates the total tangent stiffness matrix $\left[K_i^T\right]$ from the restart point.
3. The perturbation load is required and must be defined or modified at this step (see Application of Perturbation Loads (p. 828) and also the **PERTURB** command description) for the calculation of the linearly perturbed displacements and stresses.
4. Upon the onset of the second **SOLVE** command, $\{F_{perturbed}\}$ is used to calculate the linearly perturbed displacement $\{U_{perturbed}\}$ following the equation:

   $$\left[K_i^T\right]\left\{U_{perturbed}\right\} = \left\{F_{perturbed}\right\} \tag{15.230}$$

   Then, internally, $\{U_{perturbed}\}$ is used to calculate the linearly perturbed stresses $\{\sigma\}$, and the linearly perturbed stress stiffening matrix $[S_{perturbed}]$ is calculated:

   $$\left[S_{perturbed}\right] = \int_{vol} \left[S_g\right]^T [\sigma] \left[S_g\right] d(vol) \tag{15.231}$$

   following a formula similar to Equation 3.61 (p. 45).

   Next, the nodal coordinates are updated automatically if large-deflection was included in the base analysis (**NLGEOM**,ON). The program then calculates the eigensolution using the following equation (similar to Equation 15.107 (p. 793)):

   $$\left[K_i^T\right]\left\{\phi_j\right\} = \lambda_j \left[S_{perturbed}\right]\left\{\phi_j\right\} \tag{15.232}$$

   where $\left[K_i^T\right]$ is the total tangent matrix from the load point of interest. $[S_{perturbed}]$ is the linearly perturbed stress stiffening matrix generated by Equation 15.230 (p. 826) and Equation 15.231 (p. 826).
5. Along with the eigenvalue buckling solution phase, a stress expansion pass is carried out. (See Interpretation of Structural Element Results After a Linear Perturbation Analysis in the *Element Reference* for details.) To use the appropriate material property and to obtain the total sum of elastic strain/stress due to the linear perturbation analysis and the base analysis, a stress expansion pass must be done along with the buckling analysis. A separate expansion pass (**EXPASS** command) is not allowed after the linear perturbation analysis.

6. *You must perform this step manually.* This step determines the total buckling load using the loads from the restart load point and the additional perturbation load $\{F_{perturbed}\}$. The user should manually calculate the total buckling load from the following equation:

$$\{F_{buckling}\} = \{F_{restart}\} + \lambda_j \{F_{perturbed}\} \quad (15.233)$$

where:

$\{F_{restart}\}$ = total loads at the current restart load point (load applications are from the `.LDHI` file)
λ_j = eigenvalues from Equation 15.232 (p. 826). Normally, the lowest eigenvalue (λ_j when j = 1) of Equation 15.232 (p. 826) is of interest.

Equation 15.233 (p. 827) is useful in a way that combines linearly or nonlinearly prestressed loads with a linearly perturbed load. For example, if the nonlinear static analysis cannot proceed further due to convergence issues, the buckling loads can still be predicted with this process under the assumption of linear perturbation.

Generally, eigensolutions for equations Equation 15.107 (p. 793) (purely linear eigenvalue buckling) and Equation 15.232 (p. 826) (perturbed eigenvalue buckling) can be performed by using the Block Lanczos eigensolver (**BUCOPT**,LANB) if stiffness matrix [K] or $\left[K_i^T\right]$ is non-singular (i.e., no rigid body mechanism is introduced into the model). In the case of purely linear eigenvalue buckling, [K] is positive definite. However, in linear perturbation, $\left[K_i^T\right]$ can be indefinite (i.e., it contains both positive and negative pivots if it is factorized by a direct sparse equation solver). The following conditions will likely invoke an indefinite $\left[K_i^T\right]$ matrix from the base analysis:

1. In the base analysis, the applied load is large enough to trigger nonlinear buckling points in the nonlinear solution process.
2. In the base analysis, the status of contact elements changes to be open or sliding.
3. In the base and perturbed eigenvalue buckling analyses, mixed u-P formulation (Lagrange multiplier) elements have been used in the model (see General Element Formulations (p. 50)).

When $\left[K_i^T\right]$ becomes indefinite (i.e, one of the above conditions occurs), the Block Lanczos eigensolver (**BUCOPT**,LANB) could fail to produce an eigensolution due to the mathematical limitation of this solver. In such a case, it is recommended that you use the subspace eigensolver (**BUCOPT**,SUBSP) to achieve a successful solution. The subspace eigensolver can still give a converged solution even when both the $\left[K_i^T\right]$ and $[S_{pertrubed}]$ matrices are indefinite. However, if the base nonlinear analysis introduces a singular $\left[K_i^T\right]$ matrix, then no buckling eigensolutions are possible. This sometimes happens when the stiffness matrix $\left[K_i^T\right]$ is highly ill-conditioned.

15.8.6. Full Harmonic Analysis Based on Linear Perturbation

A full harmonic analysis can be performed by using the linear perturbation analysis procedure. The effect of the structure from the previous static or full transient analysis is included. The theoretical steps can be summarized as follows:

1. Perform a static or full transient base analysis; use the **RESCONTROL** command to create the restart files for the load points of interest (i.e., at various substeps of a load step).

2. Restart the analysis at the load point of interest and issue the commands **PERTURB**,HARMONIC and **SOLVE**,ELFORM. The program regenerates the total tangent stiffness matrix $\left[K_i^T\right]$ from the restart point.

3. For the full harmonic analysis, the perturbation load is required and must be defined or modified here (see Application of Perturbation Loads (p. 828) and also the **PERTURB** command description).

4. Upon the onset of the second **SOLVE** command, the nodal coordinates are updated automatically if large-deflection is included in the base analysis (**NLGEOM**,ON). The program then calculates the harmonic solution using the following equation (similar to Equation 15.52 (p. 780)):

$$[M]\left\{\ddot{u}\right\}+[C]\left\{\dot{u}\right\}+\left[K_i^T\right]\{u\}=\left\{F_{perturbed}\right\} \tag{15.234}$$

   where:

   $\left[K_i^T\right]$ = total tangent matrix from the load point of interest
   [M] = structural mass matrix
   [C] = structural damping matrix

   The perturbation load $\{F_{perturbed}\}$ is generally a complex number load vector with frequency dependency. Also, in the second phase of the linear perturbation analysis, the number of harmonic solutions (or substeps) is specified by the **NSUBST** or **DELTIM** command.

5. Along with the harmonic solution phase, a stress calculation is carried out based on the material definition specified via the **PERTURB** command. (See Interpretation of Structural Element Results After a Linear Perturbation Analysis in the *Element Reference* for details.)

15.8.7. Application of Perturbation Loads

The total perturbed loads are required for static, eigenvalue buckling, and harmonic analyses and are optional for modal analyses. One of the obvious reasons they would be needed for a modal analysis is because a downstream mode superposition (MSUP) analysis needs loads.

The total perturbed loads are calculated as follows:

$$\left\{F_{perturbed}\right\}=\{F_{end}\}+\{F_{add}\} \tag{15.235}$$

where:

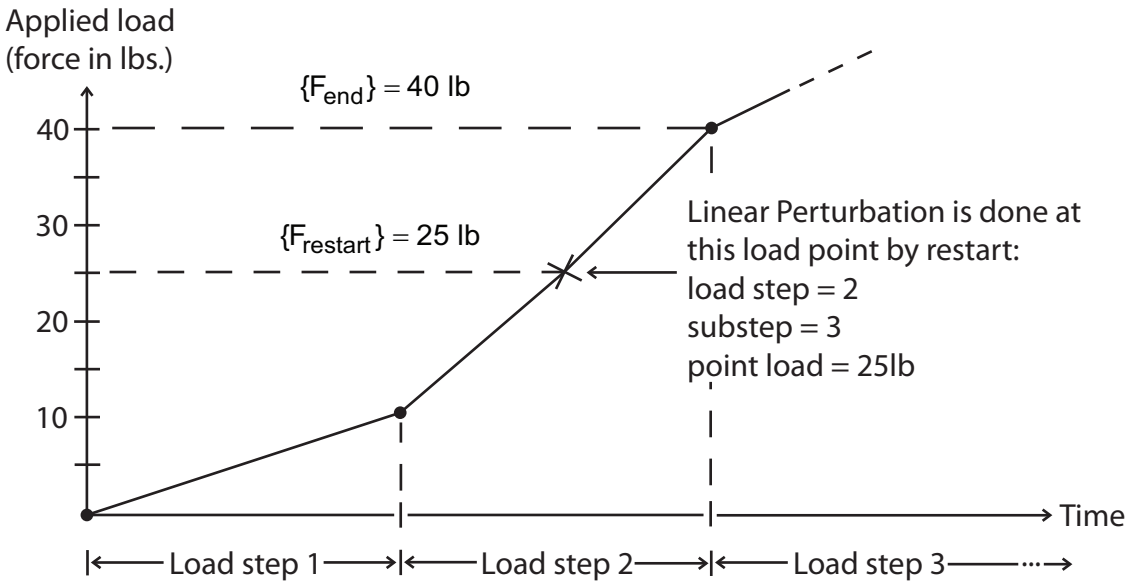
$\{F_{end}\}$ = total loads at the *end* of the load step of the current restart load point (load applications are read from the `.LDHI` file, see Figure 15.8: Linear Perturbation Started from Loadstep = 2,

Substep = 3 (p. 829)). By default, $\{F_{end}\}$ only contains loads contributed by displacement boundary conditions, inertia loads, and thermal loads (see the **PERTURB** command).
$\{F_{add}\}$ = additional (new) loads prescribed by the user in the second phase of the linear perturbation analysis (after the first **SOLVE**,ELFORM command is invoked). This additional loading is optional.

Special consideration is required if extra displacement boundary conditions are added or if constraint equations are added or deleted during the linear perturbation analysis because these modifications could introduce discontinuities into the structure.

Figure 15.8: Linear Perturbation Started from Loadstep = 2, Substep = 3 (p. 829) provides a graphic example of the definitions for the different load vectors. Two ramped load steps are shown in the figure. Since a linear perturbation analysis is started at load step = 2, substep = 3, the load vectors by definition are $\{F_{end}\}$ = 40 lb and $\{F_{restart}\}$ = 25 lb.

Figure 15.8: Linear Perturbation Started from Loadstep = 2, Substep = 3



$\{F_{restart}\}$ is the total loads at the current restart point. $\{F_{restart}\}$ may be useful for some types of linear perturbation analysis; for example, to calculate the ultimate buckling loads in an eigenvalue buckling analysis.

In the first phase of a linear perturbation analysis, the **ANTYPE**,,RESTART and **SOLVE**,ELFORM commands resume the `Jobname.RDB` database and read in the `.LDHI` file to establish the $\{F_{end}\}$ load.

$\{F_{perturbed}\}$ has a different usage and implication for each perturbed analysis type:

- For *Type* = STATIC, $\{F_{perturbed}\}$ is the actual external load for the perturbed static analysis. $\{F_{perturbed}\}$ must be supplied by the user.
- For *Type* = MODAL, $\{F_{perturbed}\}$ is calculated and stored in the `.FULL` and `.MODE` files for a subsequent mode-superposition, PSD, or other type of modal-based linear dynamic analysis.
- For *Type* = BUCKLE, $\{F_{perturbed}\}$ is the actual linear buckling load which is used to generate the linearly perturbed stress stiffening matrix for the eigenvalue buckling analysis. $\{F_{perturbed}\}$ must be supplied by the user.

- For *Type* = HARMONIC, $\{F_{perturbed}\}$ is the actual external load for the perturbed full harmonic analysis. $\{F_{perturbed}\}$, which must be supplied by the user, can be frequency dependent and can use complex input.

15.8.8. Downstream Analysis Using the Solution of a Linear Perturbation Analysis

Following the linear perturbation analysis, other analysis types can be performed by using the information from the linear perturbation analysis.

If the linear perturbation analysis is a modal analysis, the following analysis types are possible by using the `.MODE` file generated by the linear perturbation analysis and the database of the model:

- harmonic or transient analysis of the mode superposition (MSUP) method
- response spectrum analysis
- random vibration analysis

Note that the deformed mesh due to the prior static or full transient analysis is used in the linear perturbation analysis and in the downstream analysis.

In all the above listed analyses, the first load vector used is $\{F_{perturbed}\}$ from Equation 15.235 (p. 828). If more loading cases are required, it is required to regenerate a new $\{F_{perturbed}\}$ load vector by using the **MODCONT** command. Only linear material properties are supported in these analyses. The program assumes the analyses are purely linear.

Chapter 16: Preprocessing and Postprocessing Tools

The following topics concerning preprocessing and postprocessing tools are available:

16.1. Integration and Differentiation Procedures

The following integration and differentiation topics are available:

16.1.1. Single Integration Procedure

(accessed with ***VOPER** command, INT1 operation; similar capability is in POST26, INT1 command)

Given two vectors Y (parameter *Par1*) and X (parameter *Par2*), and an integration constant C1 (input as *CON1*), Y* (parameter *ParR*) is replaced by the accumulated integral of Y over X as follows:

$$\text{Set } Y_1^* = C_1 \quad \text{(for example, this would be the initial displacement of X represents time and Y represents velocity)} \tag{16.1}$$

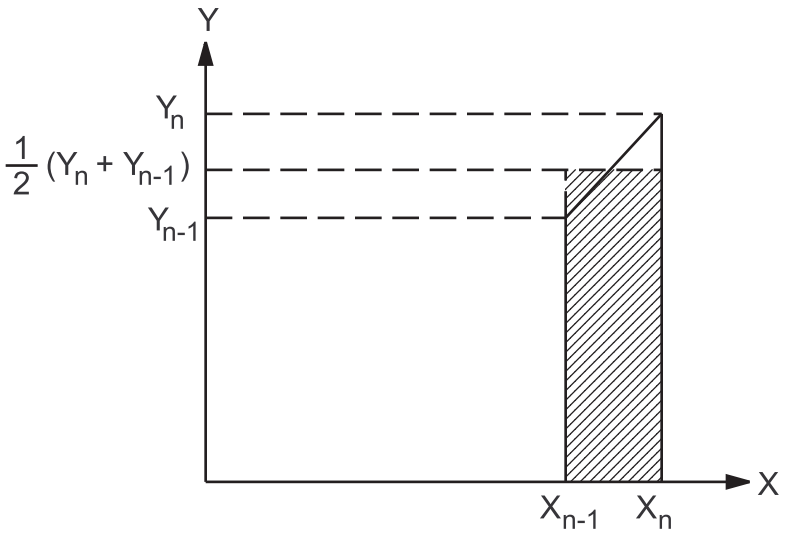
Then for each remaining point in the vector, set:

$$Y_n^* = Y_{n-1}^* + \frac{1}{2}(Y_n + Y_{n-1})(X_n - X_{n-1}) n = 2, L \tag{16.2}$$

where:

Y_n^* = integrated value of Y up to point n in the vector
L = length of the vectors

Figure 16.1: Integration Procedure



16.1.2. Double Integration Procedure

(accessed with ***VOPER** command, INT2 operation)

Given two vectors Y (parameter *Par1*) and X (parameter *Par2*), integration constants C_1 and C_2, (input as *CON1* and *CON2*) set:

$$Y_1^* = C_1 \quad \text{(for example, this would be the initial velocity if X represents time and Y represents acceleration)} \tag{16.3}$$

$$Y_1^{**} = C_2 \quad \text{(for example, this would be the initial displacement if X represents time and Y represents acceleration)} \tag{16.4}$$

Then, for each remaining point in the vector,

$$Y_n^* = Y_{n-1}^* + \frac{1}{2}(Y_n + Y_{n-1})(X_n - X_{n-1}) \tag{16.5}$$

$$Y_n^{**} = Y_{n-1}^{**} + (X_n - X_{n-1})Y_{n-1}^* + \frac{1}{6}(X_n - X_{n-1})^2(2Y_{n-1} + Y_n) \tag{16.6}$$

16.1.3. Differentiation Procedure

(accessed with ***VOPER** Command, DER1 Operation; similar capability is in POST26 **DERIV** command)

Given two vectors Y (parameter *Par1*) and X (parameter *Par2*), the derivative is found by averaging the slopes of two adjacent intervals (central difference procedure):

$$\dot{Y}_{n+1} = \frac{\frac{Y_{n+2} - Y_{n+1}}{X_{n+2} - X_{n+1}}(X_{n+1} - X_n) + \frac{Y_{n+1} - Y_n}{X_{n+1} - X_n}(X_{n+2} - X_{n+1})}{X_{n+2} - X_n} \tag{16.7}$$

A constant second derivative is assumed for the starting and ending intervals.

$$\dot{Y}_1 = \frac{Y_2 - Y_1}{X_2 - X_1} \tag{16.8}$$

$$\dot{Y}_L = \frac{Y_L - Y_{L-1}}{X_L - X_{L-1}} \tag{16.9}$$

For **DERIV** calculation, the first and last terms may differ slightly from that calculated with ***VOPER** because **DERIV** linearly extrapolates these terms from adjacent values.

16.1.4. Double Differentiation Procedure

(accessed by ***VOPER** command, DER2 Operation)

This is performed by simply repeating the differentiation procedure reported above.

16.2. Fourier Coefficient Evaluation

Fourier coefficients can be evaluated using the ***MFOURI** command. Given two vectors defining data points to be fit (parameters *CURVE* and *THETA*) and two more vectors defining which terms of the trigonometric series are desired to be computed (parameters *MODE* and *ISYM*), the desired coefficients can be computed (parameter *COEFF*). The curve fitting cannot be perfect, as there are more data than unknowns. Thus, an error R_i will exist at each data point:

$$\begin{aligned}
R_1 &= A_1 + A_2\cos\theta_1 + A_3\sin\theta_1 + A_4\cos 2\theta_1 + A_5\sin 2\theta_1 \\
&\quad + A_6\cos 3\theta_1 + A_7\sin 3\theta_1 + \ldots + A_L F(M\theta_1) - C_1 \\
R_2 &= A_1 + A_2\cos\theta_2 + A_3\sin\theta_2 + A_4\cos 2\theta_2 + A_5\sin 2\theta_2 \\
&\quad + A_6\cos 3\theta_2 + A_7\sin 3\theta_2 + \ldots + A_L F(M\theta_2) - C_2 \\
&\vdots \\
R_i &= A_1 + A_2\cos\theta_i + A_3\sin\theta_i + A_4\cos 2\theta_i + A_5\sin 2\theta_i \\
&\quad + A_6\cos 3\theta_i + A_7\sin 3\theta_i + \ldots + A_L F(M\theta_i) - C_i \\
&\vdots \\
R_m &= A_1 + A_2\cos\theta_m + A_3\sin\theta_m + A_4\cos 2\theta_m + A_5\sin 2\theta_m \\
&\quad + A_6\cos 3\theta_m + A_7\sin 3\theta_m + \ldots + A_L F(M\theta_m) - C_m
\end{aligned} \tag{16.10}$$

where:

R_i = error term (residual) associated with data point i
A = desired coefficients of Fourier series (parameter *COEFF*)
θ_i = angular location of data points i (parameter *THETA*)
L = number of terms in Fourier series
F = sine or cosine, depending on ISYM (parameter *ISYM*)
M = multiplier on θ_i (parameter *MODE*)
C_i = value of data point i (parameter *CURVE*)
m = number of data points (length of *CURVE* parameter array)

Equation 16.10 (p. 833) can be reduced to matrix form as:

$$\{R\}_{m,1} = [G]_{m,L}\{A\}_{L,1} - \{C\}_{L,1} \tag{16.11}$$

where:

{R} = vector of error terms
{G} = matrix of sines and cosines, evaluated at the different data points
{A} = vector of desired coefficients
{C} = vector of data points

Note that m > L. If m = L, the coefficients would be uniquely determined with {R} = {O} and Equation 16.11 (p. 833) being solved for {A} by direct inversion.

The method of least squares is used to determine the coefficients {A}. This means that $\sum_{i=1}^{m}(R_i)^2$ is to be minimized. The minimization is represented by

$$\frac{\partial \sum_{i=1}^{m}(R_i)^2}{\partial A_j} = 0 \tag{16.12}$$

where A_j is the jth component of {A}. Note that

$$\{R\}^T\{R\} = \sum_{i=1}^{m}(R_i)^2 \tag{16.13}$$

The form on the left-hand side of Equation 16.13 (p. 834) is the more convenient to use. Performing this operation on Equation 16.11 (p. 833),

$$\{R\}^T\{R\} = \{A\}^T[G]^T[G]\{A\} - 2\{A\}^T[G]^T\{C\} + \{C\}^T\{C\} \tag{16.14}$$

Minimizing this with respect to $\{A\}^T$ (Equation 16.12 (p. 834)), it may be shown that:

$$\{0\} = 2[G]^T[G]\{A\} - 2[G]^T\{C\} \tag{16.15}$$

or

$$[G]^T[G]\{A\} = [G]^T\{C\} \tag{16.16}$$

Equation 16.16 (p. 834) is known as the "normal equations" used in statistics. Finally,

$$\{A\} = ([G]^T[G])^{-1}[G]^T\{C\} \tag{16.17}$$

$[G^T]$ could not have been "cancelled out" of Equation 16.16 (p. 834) because it is not a square matrix. However, $[G]^T[G]$ is square.

In spite of the orthogonal nature of a trigonometric series, the value of each computed coefficient is dependent on the number of terms requested because of the least squares fitting procedure which takes place at the input data points. Terms of a true Fourier series are evaluated not by a least squares fitting procedure, but rather by the integration of a continuous function (e.g., Euler formulas, p. 469 of Kreyszig([23] (p. 922))).

16.3. Statistical Procedures

The following statistical procedures topics are available:

16.3.1. Mean, Covariance, Correlation Coefficient

The mean, variance, covariance, and correlation coefficients of a multiple subscripted parameter are computed (using the ***MOPER** command). Refer to Kreyszig([162] (p. 930)) for the basis of the following formulas. All operations are performed on columns to conform to the database structure. The covariance is assumed to be a measure of the association between columns.

The following notation is used:

where:

[x] = starting matrix
i = row index of first array parameter matrix
j = column index of first array parameter matrix
m = number of rows in first array parameter matrix
n = number of columns in first array parameter matrix
subscripts s, t = selected column indices
[S] = covariance matrix n x n
[c] = correlation matrix n x n
σ_s^2 = variance

The mean of a column is:

$$\bar{x}_j = \sum_{i=1}^{m} \frac{x_{ij}}{m} \tag{16.18}$$

The covariance of the columns s and t is:

$$S_{st} = \sum_{i=1}^{m} \frac{(x_{is} - \bar{x}_s)(x_{it} - \bar{x}_t)}{m-1} \tag{16.19}$$

The variance, σ_s^2, of column s is the diagonal term S_{ss} of the covariance matrix [S]. The equivalent common definition of variance is:

$$\sigma_s^2 = \sum_{i=1}^{m} \frac{(x_{is} - \bar{x}_s)^2}{m-1} \tag{16.20}$$

The correlation coefficient is a measure of the independence or dependence of one column to the next. The correlation and mean operations are based on Hoel([163] (p. 930)) (and initiated when CORR is inserted in the *Oper* field of the ***MOPER** command).

Correlation coefficient:

$$C_{st} = \frac{S_{st}}{\sqrt{S_{ss}}\sqrt{S_{tt}}} \tag{16.21}$$

value S of the terms of the coefficient matrix range from -1.0 to 1.0 where:

-1.0 = fully inversely related
0.0 = fully independent
1.0 = fully directly related

16.3.2. Random Samples of a Uniform Distribution

A vector can be filled with a random sample of real numbers based on a uniform distribution with given lower and upper bounds (using RAND in the *Func* field on the ***VFILL** command) (see Figure 16.2: Uniform Density (p. 836)):

$$f(x) = 1.0 \qquad a \le x \le b \tag{16.22}$$

where:

a = lower bound (input as CON1 on ***VFILL** command)
b = upper bound (input as CON2 on ***VFILL** command)

Figure 16.2: Uniform Density

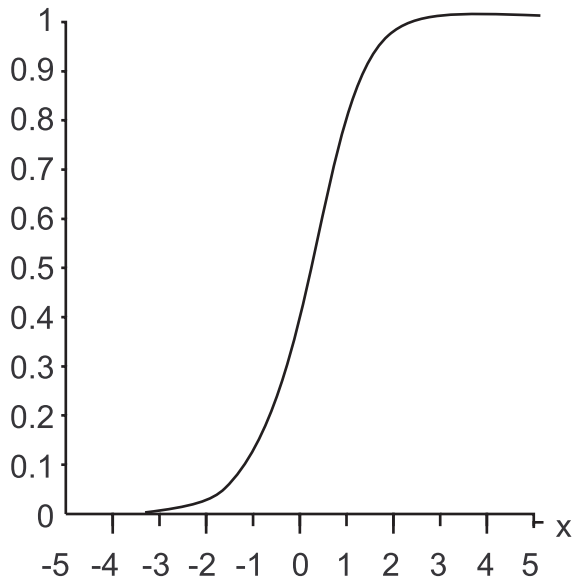
The numbers are generated using the URN algorithm of Swain and Swain([161] (p. 930)). The initial seed numbers are hard coded into the routine.

16.3.3. Random Samples of a Gaussian Distribution

A vector may be filled with a random sample of real numbers based on a Gaussian distribution with a known mean and standard deviation (using GDIS in the *Func* field on the ***VFILL** command).

First, random numbers P(x), with a uniform distribution from 0.0 to 1.0, are generated using a random number generator. These numbers are used as probabilities to enter a cumulative standard normal probability distribution table (Abramowitz and Stegun([160] (p. 929))), which can be represented by Figure 16.3: Cumulative Probability Function (p. 837) or the Gaussian distribution function:

Figure 16.3: Cumulative Probability Function



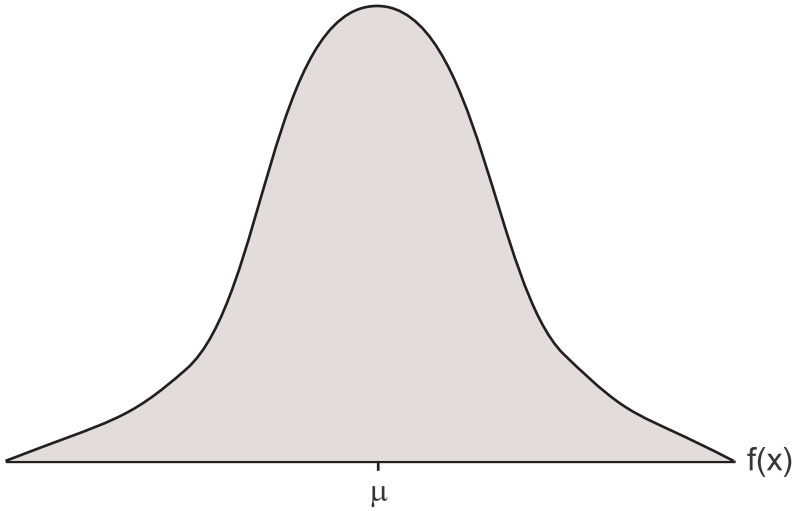
$$P(x) = \int_{-\infty}^{x} f(t)dt \qquad (16.23)$$
$$= \text{no closed form}$$

where:

f(t) = Gaussian density function

The table maps values of P(x) into values of x, which are standard Gaussian distributed random numbers from -5.0 to 5.0, and satisfy the Gaussian density function (Figure 16.4: Gaussian Density (p. 837)):

Figure 16.4: Gaussian Density



$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \qquad -\infty < x < \infty \tag{16.24}$$

where:

μ = mean (input as CON1 on ***VFILL** command)
σ = standard deviation (input as CON2 on ***VFILL** command)

The x values are transformed into the final Gaussian distributed set of random numbers, with the given mean and standard deviation, by the transformation equation:

$$z = \sigma x + \mu \tag{16.25}$$

16.3.4. Random Samples of a Triangular Distribution

A vector may be filled with a random sample of real numbers based on a triangular distribution with a known lower bound, peak value location, and upper bound (using TRIA in the *Func* field on the ***VFILL** command).

First, random numbers P(x) are generated as in the Gaussian example. These P(x) values (probabilities) are substituted into the triangular cumulative probability distribution function:

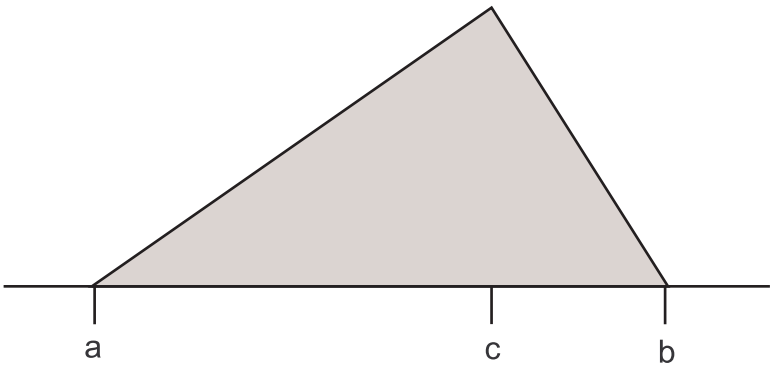
$$P(x) = \begin{cases} 0 & \text{if } x < a \\ \dfrac{(x-a)^2}{(b-a)(c-a)} & \text{if } a \le x \le c \\ 1\dfrac{(b-x)^2}{(b-a)(b-c)} & \text{if } c < x \le b \\ 1 & \text{if } b < x \end{cases} \tag{16.26}$$

where:

a = lower bound (input as CON1 on ***VFILL** command)
c = peak location (input as CON2 on ***VFILL** command)
b = upper bound (input as CON3 on ***VFILL** command)

which is solved for values of x. These x values are random numbers with a triangular distribution, and satisfy the triangular density function (Figure 16.5: Triangular Density (p. 838)):

Figure 16.5: Triangular Density



$$f(x) = \begin{cases} \dfrac{2(x-a)}{(b-c)(c-a)} & \text{if } a \le x \le c \\ \dfrac{2(b-x)}{(b-a)(b-c)} & \text{if } c < x \le b \\ 0 & \text{otherwise} \end{cases} \tag{16.27}$$

16.3.5. Random Samples of a Beta Distribution

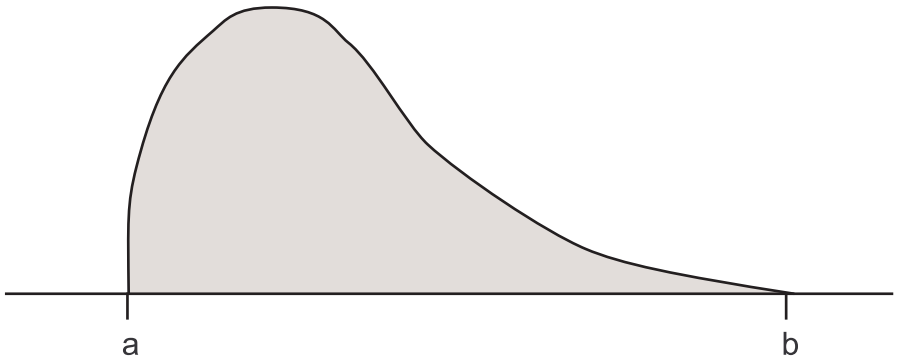
A vector may be filled with a random sample of real numbers based on a beta distribution with known lower and upper bounds and α and β parameters (using BETA in the *Func* field on the ***VFILL** command).

First, random numbers P(x) are generated as in the Gaussian example. These random values are used as probabilities to enter a cumulative beta probability distribution table, generated by the program. This table can be represented by a curve similar to (Figure 16.3: Cumulative Probability Function (p. 837)), or the beta cumulative probability distribution function:

$$P(x) = \int_{-\infty}^{x} f(t)dt = \text{no closed form} \tag{16.28}$$

The table maps values of P(x) into x values which are random numbers from 0.0 to 1.0. The values of x have a beta distribution with given α and β values, and satisfy the beta density function (Figure 16.6: Beta Density (p. 839)):

Figure 16.6: Beta Density



$$f(x) = \begin{cases} \dfrac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} & \text{if } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{16.29}$$

where:

a = lower bound (input as CON1 on ***VFILL** command)
b = upper bound (input as CON2 on ***VFILL** command)
α = alpha parameter (input as CON3 on ***VFILL** command)
β = beta parameter (input as CON4 on ***VFILL** command)

$B(\alpha, \beta)$ = beta function

$$= \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \qquad \text{for } \alpha > 0,\ \beta > 0$$

f(t) = beta density function

The x values are transformed into the final beta distributed set of random numbers, with given lower and upper bounds, by the transformation equation:

$$z = a + (b-a)x \tag{16.30}$$

16.3.6. Random Samples of a Gamma Distribution

A vector may be filled with a random sample of real numbers based on a gamma distribution with a known lower bound for α and β parameters (using GAMM in the *Func* field on the ***VFILL** command).

First, random numbers P(x) are generated as in the Gaussian example. These random values are used as probabilities to enter a cumulative gamma probability distribution table, generated by the program. This table can be represented by a curve similar to Figure 16.7: Gamma Density (p. 840), or the gamma cumulative probability distribution function:

$$P(x) = \int_{-\infty}^{x} f(t)dt \tag{16.31}$$

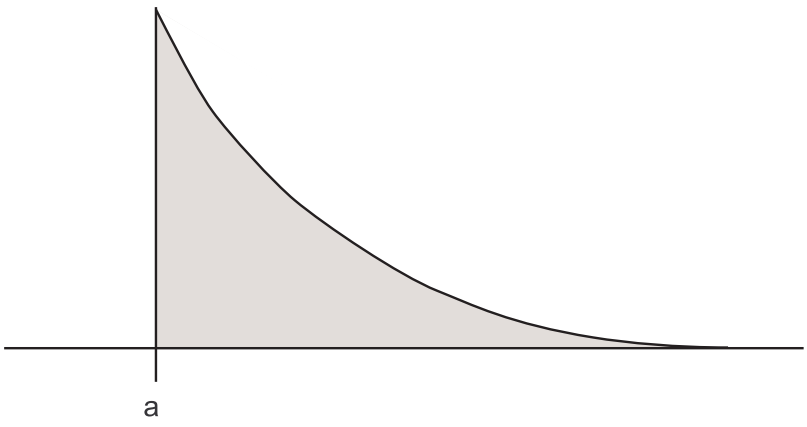
$$= \text{no closed form}$$

where:

f(t) = gamma density function.

The table maps values of P(x) into values of x, which are random numbers having a gamma distribution with given α and β values, and satisfy the gamma distribution density function (Figure 16.7: Gamma Density (p. 840)):

Figure 16.7: Gamma Density



$$f(x) = \begin{cases} \dfrac{\beta^{-\alpha} x^{\alpha-1} e^{-x/\beta}}{\Gamma(\alpha)} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{16.32}$$

where:

$$\Gamma(\alpha+1) = \int_0^{\infty} t^{\alpha} e^{-t} dt \qquad \alpha \geq 0$$

α = alpha parameter of gamma function (input as CON2 on ***VFILL** command)
β = beta parameter of gamma density function (input as CON3 on ***VFILL** command)
a = lower bound (input as CON1 on ***VFILL** command)

The x values are relocated relative to the given lower bound by the transformation equation:

$$z = a + x \tag{16.33}$$

Chapter 17: Postprocessing

The following postprocessing topics are available:

17.1. POST1 - Derived Nodal Data Processing

17.1.1. Derived Nodal Data Computation

The computation of derived data (data derived from nodal unknowns) is discussed in Structures with Geometric Nonlinearities (p. 29) through Acoustics (p. 253). Derived nodal data is available for solid and shell elements (except SHELL61). Available data include stresses, strains, thermal gradients, thermal fluxes, pressure gradients, electric fields, electric flux densities, magnetic field intensities, magnetic flux densities, and magnetic forces. Structural nonlinear data is processed in a similar fashion and includes equivalent stress, stress state ratio, hydrostatic pressure, accumulated equivalent plastic strain, plastic state variable, and plastic work.

POST1 averages the component tensor or vector data at corner nodes used by more than one element.

$$\sigma_{ik} = \frac{\sum_{j=1}^{N_k} \sigma_{ijk}}{N_k} \tag{17.1}$$

where:

σ_{ik} = average derived data component i at node k
σ_{ijk} = derived data component i of element j at node k
N_k = number of elements connecting to node k

For higher-order elements, component tensor or vector data at midside nodes are calculated by directly averaging the averaged corner node values, so Equation 17.1 (p. 843) is not used for midside nodes. Midside node values are printed or plotted only via PowerGraphics (/GRAPHICS,POWER) and /EFACET,2.

Combining principal tensor data (principal stress, principal strain) or vector magnitudes at the nodes may either be computed using the averaged component data (*KEY* = 0, **AVPRIN** command):

$$\sigma_{ck} = f(\sigma_{ik}) \tag{17.2}$$

where:

$f(\sigma_{ik})$ = function to compute principal data from component data as given in Structures with Geometric Nonlinearities (p. 29) through Acoustics (p. 253).

or be directly averaged (*KEY* = 1, **AVPRIN** command):

$$\sigma_{ck} = \frac{\sum_{j=1}^{k} \sigma_{cjk}}{N_k} \tag{17.3}$$

where:

σ_{ck} = averaged combined principal data at node k
σ_{cjk} = combined principal data of element j at node k

17.2. POST1 - Vector and Surface Operations

17.2.1. Vector Operations

The dot product of two vectors $\{A\}(= A_x \hat{i} + A_y \hat{j} + A_z \hat{k})$ and $\{B\}(= B_x \hat{i} + B_y \hat{j} + B_z \hat{k})$ is provided (with the **VDOT** command) as:

$$\{A\} \cdot \{B\} = A_x B_x + A_y B_y + A_z B_z \tag{17.4}$$

The cross product of two vectors {A} and {B} is also provided (with the **VCROSS** command) as:

$$\{A\} \times \{B\} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix} \tag{17.5}$$

In both operations, the components of vectors {A} and {B} are transformed to global Cartesian coordinates before the calculations. The results of the cross product are also in global Cartesian coordinates.

17.2.2. Surface Operations

(Integration of Values Across a Free Surface)

Nodal values across a free surface can be integrated (using the **INTSRF** command). The free surface is determined by a selected set of nodes which must lie on an external surface of the selected set of elements.

Only pressure values can be integrated (for purposes of lift and drag calculations in fluid flow analyses). As a result of the integration, force and moment components in the global Cartesian coordinate system are:

$$\{F_t\} = \int_{area} \{p\} d(area) \tag{17.6}$$

$$\{F_r\} = \int_{area} \{r\} \times \{p\} d(area) \tag{17.7}$$

where:

$\{F_t\}$ = force components
$\{F_r\}$ = moment components
$\{r\}$ = position vector = $\lfloor X\,Y\,Z \rfloor^T$
$\{p\}$ = distributed pressure vector
area = surface area

In the finite element implementation, the position vector {r} is taken with respect to the origin.

17.3. POST1 - Path Operations

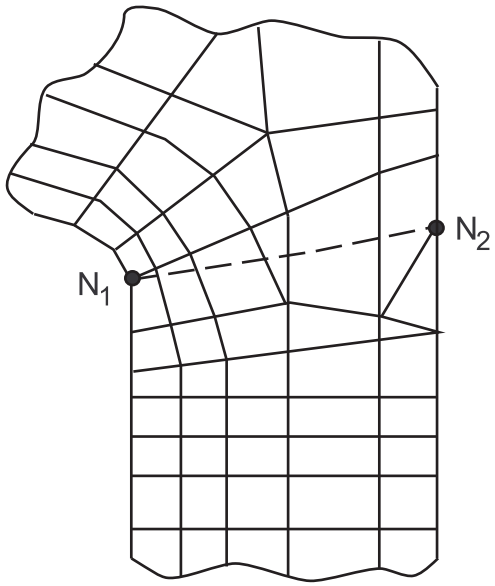
General vector calculus may be performed along any arbitrary 2-D or 3-D path through a solid element model. Nodal data, element data, and data stored with element output tables (**ETABLE** command) may be mapped onto the path and operated on as described below.

17.3.1. Defining the Path

A path is defined by first establishing path parameters (**PATH** command) and then defining path points which create the path (**PPATH** command). The path points may be nodes, or arbitrary points defined by geometry coordinates. A segment is a line connecting two path points. The number of path points used to create a path and the number of divisions used to discretize the path are input (using *Npts* and the *nDiv* parameter on the **PATH** command). The discretized path divisions are interpolated between path points in the currently active coordinate system (**CSYS** command), or as directly input (on the **PPATH** command). A typical segment is shown in Figure 17.1: Typical Path Segment (p. 846) as going from points N_1 to N_2, for the first segment.

The geometry of each point along the path is stored. The geometry consists of the global Cartesian coordinates (output label XG, YG, ZG) and the length from the first path point along the path (output label S). The geometry is available for subsequent operations.

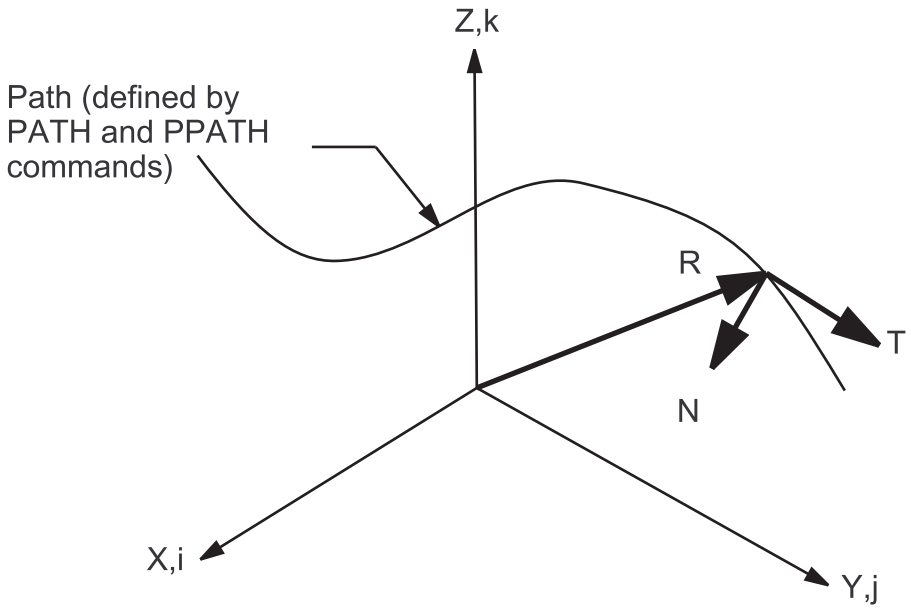
Figure 17.1: Typical Path Segment



17.3.2. Defining Orientation Vectors of the Path

In addition, position (R), unit tangent (T), and unit normal (N) vectors to a path point are available as shown in Figure 17.2: Position and Unit Vectors of a Path (p. 846). These three vectors are defined in the active Cartesian coordinate system.

Figure 17.2: Position and Unit Vectors of a Path



The position vector R (stored with **PVECT**,RADI command) is defined as:

$$\{R\} = \begin{Bmatrix} x_n \\ y_n \\ z_n \end{Bmatrix} \tag{17.8}$$

where:

x_n = x coordinate in the active Cartesian system of path point n, etc.

The unit tangent vector T (stored with **PVECT**,TANG command) is defined as:

$$\{T_1\} = C \begin{Bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{Bmatrix} \quad \text{(for first path point)} \tag{17.9}$$

$$\{T_n\} = C \begin{Bmatrix} x_{n+1} - x_{n-1} \\ y_{n+1} - y_{n-1} \\ z_{n+1} - z_{n-1} \end{Bmatrix} \quad \text{(for intermediate path point)} \tag{17.10}$$

$$\{T_L\} = C \begin{Bmatrix} x_L - x_{L-1} \\ y_L - y_{L-1} \\ z_L - z_{L-1} \end{Bmatrix} \quad \text{(for last path point)} \tag{17.11}$$

where:

x, y, z = coordinate of a path point in the active Cartesian system n = 2 to (L-1)
L = number of points on the path
C = scaling factor so that {T} is a unit vector

The unit normal vector N (**PVECT**,NORM command) is defined as:

$$\{N\} = \{T\} \times \{k\} / \left| \{T\} \times \{k\} \right| \tag{17.12}$$

where:

x = cross product operator

$$\{k\} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$$

{N} is not defined if {T} is parallel to {k}.

17.3.3. Mapping Nodal and Element Data onto the Path

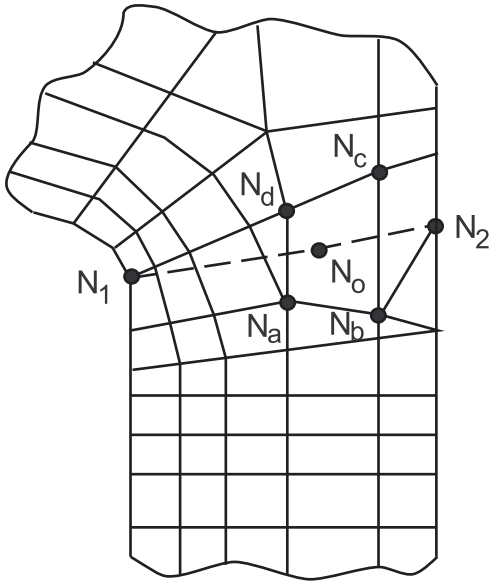
Having defined the path, the nodal or element data (as requested by *Item*,*Comp* on the **PDEF** command) may be mapped onto the path. For each path point, the selected elements are searched to find an element containing that geometric location. In the lower order finite element example of Figure 17.3: Mapping Data (p. 848), point N_o has been found to be contained by the element described by nodes N_a, N_b, N_c and N_d. Nodal degree of freedom data is directly available at nodes N_a, N_b, N_c and N_d. Element result data may be interpreted either as averaged data over all elements connected to a node (as described in the Nodal Data Computation topic, see POST1 - Derived Nodal Data Processing (p. 843)) or as unaver-

aged data taken only from the element containing the path interpolation point (using the *Avglab* option on the **PDEF** command). When using the material discontinuity option (MAT option on the **PMAP** command) unaveraged data is mapped automatically.

Caution should be used when defining a path for use with the unaveraged data option. Avoid defining a path (**PPATH** command) directly along element boundaries since the choice of element for data interpolation may be unpredictable. Path values at nodes use the element from the immediate preceding path point for data interpolation.

The value at the point being studied (i.e., point N_o) is determined by using the element shape functions together with these nodal values. Principal results data (principal stresses, strains, flux density magnitude, etc.) are mapped onto a path by first interpolating item components to the path and then calculating the principal value from the interpolated components.

Figure 17.3: Mapping Data



Higher order elements include midside nodal (DOF) data for interpolation. Element data at the midside nodes are averaged from corner node values before interpolation.

17.3.4. Operating on Path Data

Once nodal or element data are defined as a path item, its associated path data may be operated on in several ways. Path items may be combined by addition, multiplication, division, or exponentiation (**PCALC** command). Path items may be differentiated or integrated with respect to any other path item (**PCALC** command). Differentiation is based on a central difference method without weighting:

$$\dot{A}_1 = \frac{A_2 - A_1}{B_2 - B_1} \times S \qquad \text{(for first path point)} \tag{17.13}$$

$$\dot{A}_n = \frac{A_{n+1} - A_{n-1}}{B_{n+1} - B_{n-1}} \times S \qquad \text{(for intermediate path points)} \tag{17.14}$$

$$\dot{A}_L = \frac{A_L - A_{L-1}}{B_L - B_{L-1}} \times S \qquad \text{(for last path point)} \tag{17.15}$$

where:

A = values associated with the first labeled path in the operation (*LAB1*, on the **PCALC**,DERI command)
B = values associated with the second labeled path in the operation (*LAB2*, on the **PCALC**,DERI command)
n = 2 to (L-1)
L = number of points on the path
S = scale factor (input as FACT1, on the **PCALC**,DERI command)

If the denominator is zero for Equation 17.13 (p. 848) through Equation 17.15 (p. 848), then the derivative is set to zero.

Integration is based on the rectangular rule (see Figure 16.1: Integration Procedure (p. 832) for an illustration):

$$A_1^* = 0.0 \tag{17.16}$$

$$A_n^* = A_{n-1}^* + \frac{1}{2}(A_n + A_{n+1})(B_n - B_{n-1}) \times S \tag{17.17}$$

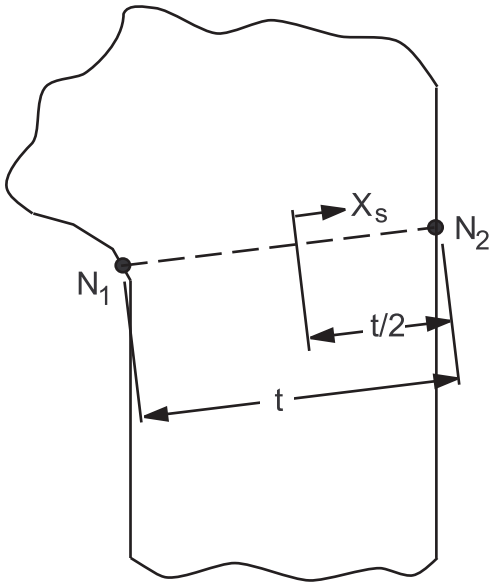
Path items may also be used in vector dot (**PDOT** command) or cross (**PCROSS** command) products. The calculation is the same as the one described in the Vector Dot and Cross Products Topic, above. The only difference is that the results are not transformed to be in the global Cartesian coordinate system.

17.4. POST1 - Stress Linearization

An option is available to allow a separation of stresses through a section into constant (membrane) and linear (bending) stresses. An approach similar to the one used here is reported by Gordon([63] (p. 924)). The stress linearization option (accessed using the **PRSECT**, **PLSECT**, or **FSSECT** commands) uses a path defined by two nodes (with the **PPATH** command). The section is defined by a path consisting of two end points (nodes N_1 and N_2) as shown in Figure 17.4: Coordinates of Cross Section (p. 850) (nodes) and 47 intermediate points (automatically determined by linear interpolation in the active display coordinate system (**DSYS**). Nodes N_1 and N_2 are normally both presumed to be at free surfaces.

Initially, a path must be defined and the results mapped onto that path as defined above. The logic for most of the remainder of the stress linearization calculation depends on whether the structure is axisymmetric or not, as indicated by the value of ρ (input as *RHO* on **PRSECT**, **PLSECT**, or **FSSECT** commands). For ρ = 0.0, the structure is not axisymmetric (Cartesian case); and for nonzero values of ρ, the structure is axisymmetric. The explicit definition of ρ, as well as the discussion of the treatment of axisymmetric structures, is discussed later.

Figure 17.4: Coordinates of Cross Section



17.4.1. Cartesian Case

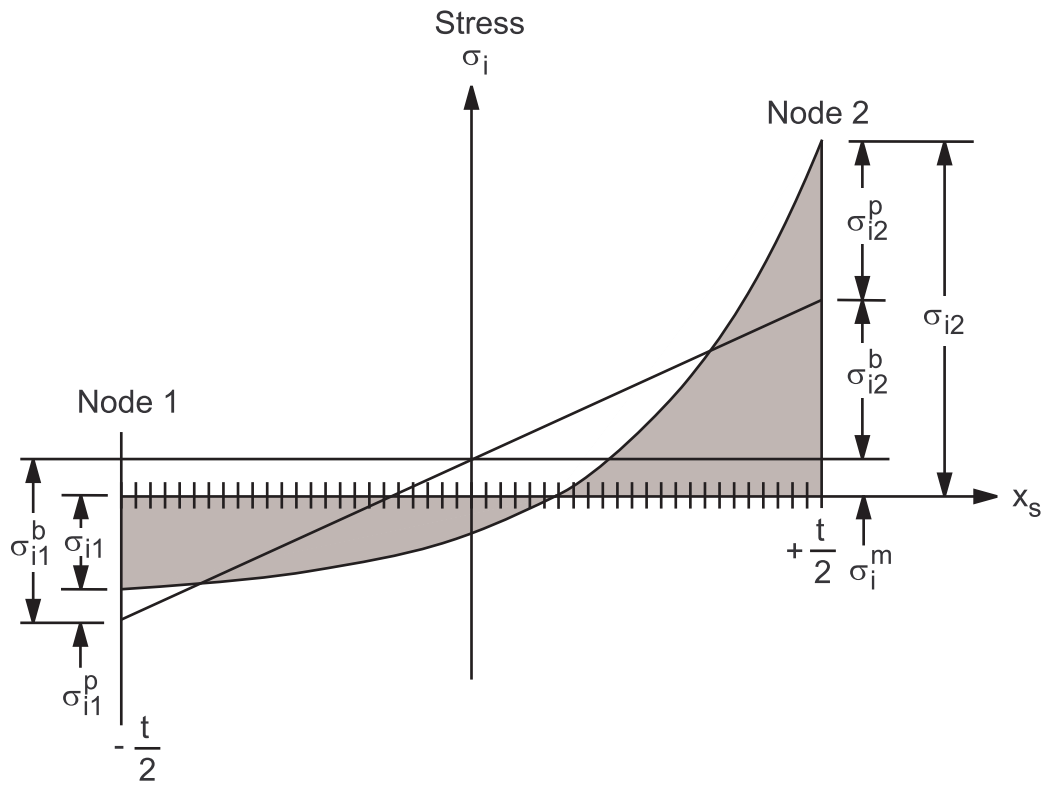
Refer to Figure 17.5: Typical Stress Distribution (p. 851) for a graphical representation of stresses. The membrane values of the stress components are computed from:

$$\sigma_i^m = \frac{1}{t}\int_{-t/2}^{t/2} \sigma_i \, dx_s \tag{17.18}$$

where:

σ_i^m = membrane value of stress component i
t = thickness of section, as shown in Figure 17.4: Coordinates of Cross Section (p. 850)
σ_i = stress component i along path from results file (`total' stress)
x_s = coordinate along path, as shown in Figure 17.4: Coordinates of Cross Section (p. 850)

Figure 17.5: Typical Stress Distribution



The subscript i is allowed to vary from 1 to 6, representing σ_x, σ_y, σ_z, σ_{xy}, σ_{yz}, and σ_{xz}, respectively. These stresses are in global Cartesian coordinates. Strictly speaking, the integrals such as the one above are not literally performed; rather it is evaluated by numerical integration:

$$\sigma_i^m = \frac{1}{48}\left(\frac{\sigma_{i,1}}{2} + \frac{\sigma_{i,49}}{2} + \sum_{j=2}^{47}\sigma_{i,j}\right) \tag{17.19}$$

where:

$\sigma_{i,j}$ = total stress component i at point j along path

The integral notation will continue to be used, for ease of reading.

The "bending" values of the stress components at node N_1 are computed from:

$$\sigma_{i1}^b = \frac{-6}{t^2}\int_{-t/2}^{t/2}\sigma_i\, x_s\, dx_s \tag{17.20}$$

where:

σ_{i1}^b = bending value of stress component i at node N_1

The bending values of the stress components at node N_2 are simply

$$\sigma_{i2}^b = -\sigma_{i1}^b \tag{17.21}$$

where:

σ_{i2}^{b} = bending value of the stress component i at node N_2

The "peak" value of stress at a point is the difference between the total stress and the sum of the membrane and bending stresses. Thus, the peak stress at node N_1 is:

$$\sigma_{i1}^{p} = \sigma_{i1} - \sigma_{i}^{m} - \sigma_{i1}^{b} \tag{17.22}$$

where:

σ_{i1}^{p} = peak value of stress component i at node N_1
σ_{i1} = value of total stress component i at node N_1

Similarly, for node N_2,

$$\sigma_{i2}^{p} = \sigma_{i2} - \sigma_{i}^{m} - \sigma_{i2}^{b} \tag{17.23}$$

At the center point (x = 0.0)

$$\sigma_{ic}^{p} = \sigma_{ic} - \sigma_{i}^{m} \tag{17.24}$$

where:

σ_{ic}^{p} = peak value of stress component i at center
σ_{ic} = computed (total) value of stress component i at center

17.4.2. Axisymmetric Case (General)

The axisymmetric case is the same, in principle, as the Cartesian case, except for the fact that there is more material at a greater radius than at a smaller radius. Thus, the neutral axis is shifted radially outward a distance x_f, as shown in Figure 17.6: Axisymmetric Cross-Section (p. 853). The axes shown in Figure 17.6: Axisymmetric Cross-Section (p. 853) are Cartesian, i.e., the logic presented here is only valid for structures axisymmetric in the global cylindrical system. As stated above, the axisymmetric case is selected if $\rho \neq 0.0$. ρ is defined as the radius of curvature of the midsurface in the X-Y plane, as shown in Figure 17.7: Geometry Used for Axisymmetric Evaluations (p. 853). A point on the centerplane of the torus has its curvatures defined by two radii: ρ and the radial position R_c. Both of these radii will be used in the forthcoming development. In the case of an axisymmetric straight section such as a cylinder, cone, or disk, $\rho = \infty$, so that the input must be a large number (or -1).

Figure 17.6: Axisymmetric Cross-Section

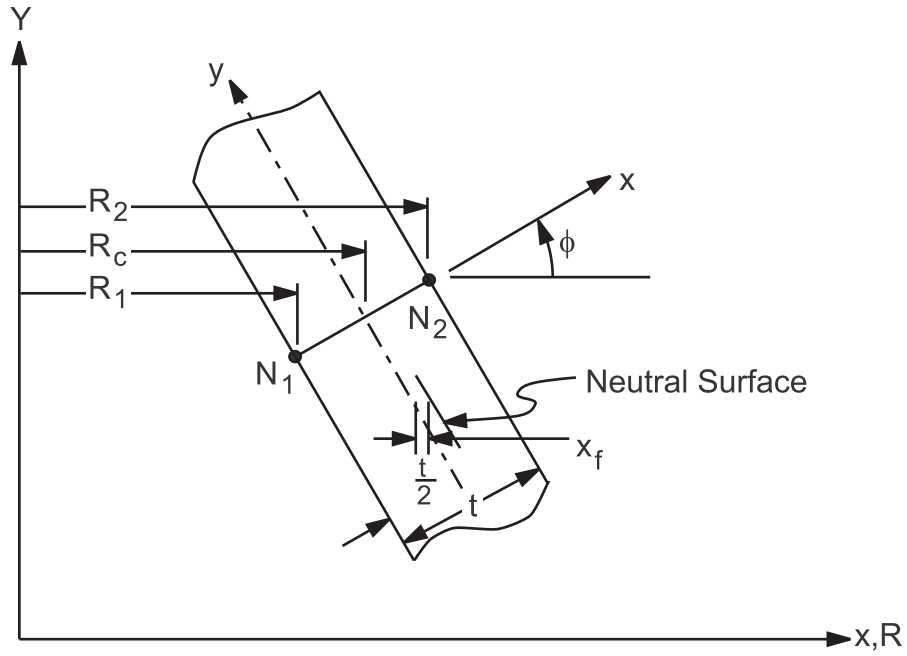
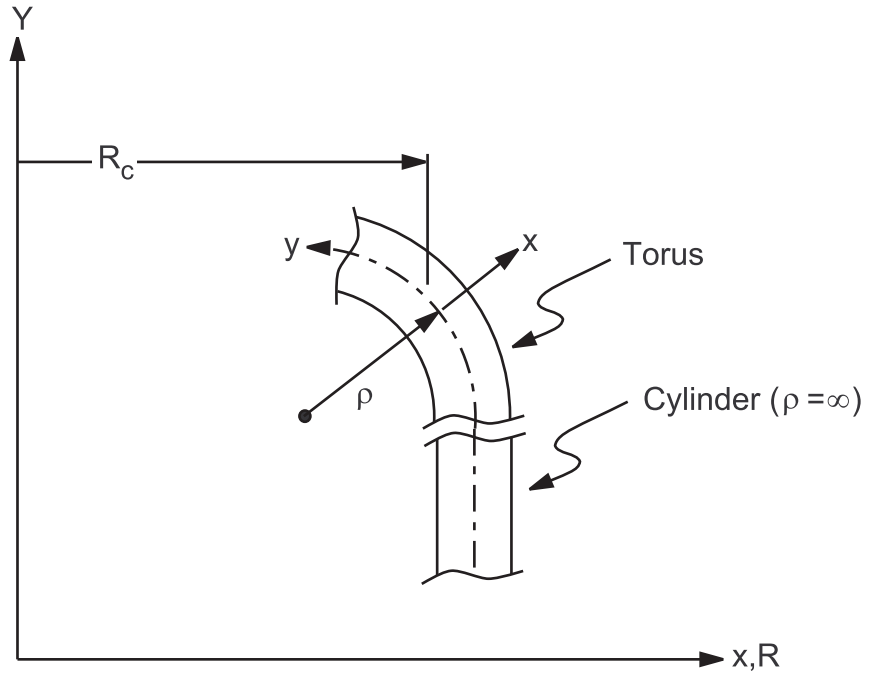


Figure 17.7: Geometry Used for Axisymmetric Evaluations



Each of the components for the axisymmetric case needs to be treated separately. For this case, the stress components are rotated into section coordinates, so that x stresses are parallel to the path and y stresses are normal to the path.

Starting with the y direction membrane stress, the force over a small sector is:

$$F_y = \int_{-t/2}^{t/2} \sigma_y R \Delta\theta \, dx \tag{17.25}$$

where:

F_y = total force over small sector
σ_y = actual stress in y (meridional) direction
R = radius to point being integrated
$\Delta\theta$ = angle over a small sector in the hoop direction
t = thickness of section (distance between nodes N_1 and N_2)

The area over which the force acts is:

$$A_y = R_c \Delta\theta t \tag{17.26}$$

where:

A_y = area of small sector

$$R_c = \frac{R_1 + R_2}{2}$$

R_1 = radius to node N_1
R_2 = radius to node N_2

Thus, the average membrane stress is:

$$\sigma_y^m = \frac{F_y}{A_y} = \frac{\int_{-t/2}^{t/2} \sigma_y R dx}{R_c t} \tag{17.27}$$

where:

σ_y^m = y membrane stress

To process the bending stresses, the distance from the center surface to the neutral surface is needed. This distance is shown in Figure 17.6: Axisymmetric Cross-Section (p. 853) and is:

$$x_f = \frac{t^2 \cos\phi}{12 R_c} \tag{17.28}$$

The derivation of Equation 17.28 (p. 854) is the same as for y_f given at the end of SHELL61 - Axisymmetric-Harmonic Structural Shell (p. 464). Thus, the bending moment may be given by:

$$M = \int_{-t/2}^{t/2} (x - x_f) dF \tag{17.29}$$

or

$$M = \int_{-t/2}^{t/2} (x - x_f) \sigma_y R \Delta\theta \, dx \tag{17.30}$$

The moment of inertia is:

$$I = \frac{1}{12} R_c \Delta\theta t^3 - R_c \Delta\theta t x_f^2 \tag{17.31}$$

The bending stresses are:

$$\sigma^b = \frac{Mc}{I} \tag{17.32}$$

where:

c = distance from the neutral axis to the extreme fiber

Combining the above three equations,

$$\sigma^b_{y1} = \frac{M(x_1 - x_f)}{I} \tag{17.33}$$

or

$$\sigma^b_{y1} = \frac{x_1 - x_f}{R_c t\left(\frac{t_2}{12} - x_f^2\right)} \int_{-t/2}^{t/2} (x - x_f)\sigma_y R\, dx \tag{17.34}$$

where:

σ^b_{y1} = y bending stress at node N_1

Also,

$$\sigma^b_{y2} = \frac{M(x_2 - x_f)}{I} \tag{17.35}$$

or

$$\sigma^b_{y2} = \frac{x_2 - x_f}{R_c t\left(\frac{t_2}{12} - x_f^2\right)} \int_{-t/2}^{t/2} (x - x_f)\sigma_y R\, dx \tag{17.36}$$

where:

σ^b_{y2} = y bending stress at node N_2

σ_x represents the stress in the direction of the thickness. Thus, σ_{x1} and σ_{x2} are the negative of the pressure (if any) at the free surface at nodes N_1 and N_2, respectively. A membrane stress is computed as:

$$\sigma^m_x = \frac{1}{t} \int_{-t/2}^{t/2} \sigma_x dx \tag{17.37}$$

where:

σ^m_x = the x membrane stress

The treatment of the thickness-direction "bending" stresses is controlled by K_B (input as *KBR* on **PRSECT**, **PLSECT**, or **FSSECT** commands). When the thickness-direction bending stresses are to be ignored (K_B = 1), bending stresses are equated to zero:

$$\sigma_{x1}^{b} = 0 \tag{17.38}$$

$$\sigma_{x2}^{b} = 0 \tag{17.39}$$

When the bending stresses are to be included ($K_B = 0$), bending stresses are computed as:

$$\sigma_{x1}^{b} = \sigma_{x1} - \sigma_{x}^{m} \tag{17.40}$$

$$\sigma_{x2}^{b} = \sigma_{x2} - \sigma_{x}^{m} \tag{17.41}$$

where:

σ_{x1}^{b} = x bending stress at node N_1
σ_{x1} = total x stress at node N_1
σ_{x2}^{b} = x bending stress at node N_2
σ_{x2} = total x stress at node N_2

and when $K_B = 2$, membrane and bending stresses are computed using Equation 17.27 (p. 854), Equation 17.34 (p. 855), and Equation 17.36 (p. 855) substituting σ_x for σ_y.

The hoop stresses are processed next.

$$\sigma_{h}^{m} = \frac{F_h}{A_h} = \frac{\Delta\phi \int_{-t/2}^{t/2} \sigma_h(\rho + x)dx}{\Delta\phi\rho t} \tag{17.42}$$

where:

σ_{h}^{m} = hoop membrane stress
F_h = total force over small sector
$\Delta\phi$ = angle over small sector in the meridional (y) direction
σ_h = hoop stress
A_h = area of small sector in the x-y plane
r = radius of curvature of the midsurface of the section (input as *RHO*)
x = coordinate thru cross-section
t = thickness of cross-section

Equation 17.42 (p. 856) can be reduced to:

$$\sigma_{h}^{m} = \frac{1}{t}\int_{-t/2}^{t/2} \sigma_h \left(1 + \frac{x}{\rho}\right) dx \tag{17.43}$$

Using logic analogous to that needed to derive Equation 17.34 (p. 855) and Equation 17.36 (p. 855), the hoop bending stresses are computed by:

$$\sigma_{h1}^{b} = \frac{x_1 - x_h}{t\left(\frac{t^2}{12} - x_h^2\right)} \int_{-t/2}^{t/2} (x - x_h)\sigma_h \left(1 + \frac{x}{\rho}\right) dx \tag{17.44}$$

and

$$\sigma_{h1}^{b} = \frac{x_1 - x_h}{t\left(\frac{t^2}{12} - x_h^2\right)} \int_{-t/2}^{t/2} (x - x_h)\sigma_h \left(1 + \frac{x}{\rho}\right) dx \tag{17.45}$$

where:

$$x_h = \frac{t^2}{12\rho} \tag{17.46}$$

for hoop-related calculations of Equation 17.44 (p. 856) and Equation 17.45 (p. 857).

An xy membrane shear stress is computed as:

$$\sigma_{xy}^{m} = \frac{1}{R_c t} \int_{-t/2}^{t/2} \sigma_{xy} R \, dx \tag{17.47}$$

where:

σ_{xy}^{m} = xy membrane shear stress
σ_{xy} = xy shear stress

Since the shear stress distribution is assumed to be parabolic and equal to zero at the ends, the xy bending shear stress is set to 0.0. The other two shear stresses (σ_{xz}, σ_{yz}) are assumed to be zero if K_B = 0 or 1. If K_B = 2, the shear membrane and bending stresses are computing using Equation 17.27 (p. 854), Equation 17.34 (p. 855), and Equation 17.36 (p. 855) substituting σ_{xy} for σ_y

All peak stresses are computed from

$$\sigma_i^{P} = \sigma_i - \sigma_i^{m} - \sigma_i^{b} \tag{17.48}$$

where:

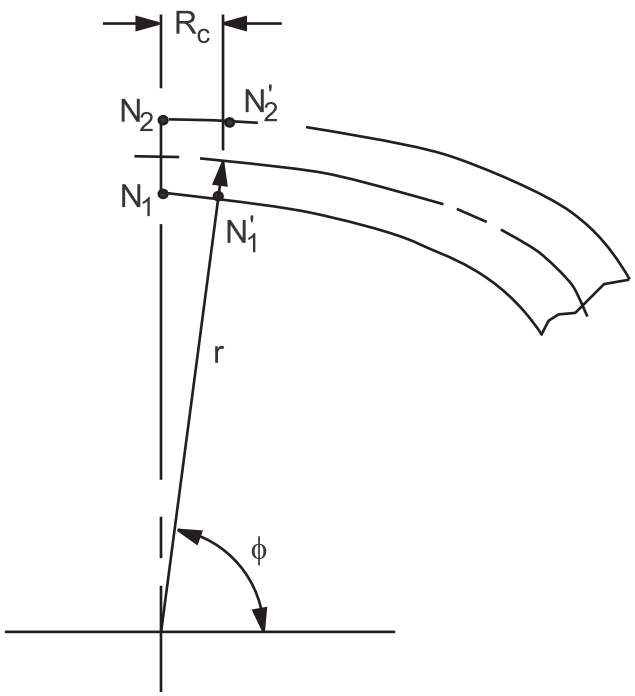
σ_i^{P} = peak value of stress component i
σ_i = total value of stress of component i

17.4.3. Axisymmetric Case

(Specializations for Centerline)

At this point it is important to mention one exceptional configuration related to the y-direction membrane and bending stress calculations above. For paths defined on the centerline (X = 0), R_c = 0 and cosΦ = 0, and therefore Equation 17.27 (p. 854), Equation 17.28 (p. 854), Equation 17.34 (p. 855), and Equation 17.36 (p. 855) are undefined. Since centerline paths are also vertical (ϕ = 90°), it follows that R = R_c, and R_c is directly cancelled from stress Equation 17.27 (p. 854), Equation 17.34 (p. 855), and Equation 17.35 (p. 855). However, x_f remains undefined. Figure 17.8: Centerline Sections (p. 858) shows a centerline path from N_1 to N_2 in which the inside and outside wall surfaces form perpendicular intersections with the centerline.

Figure 17.8: Centerline Sections

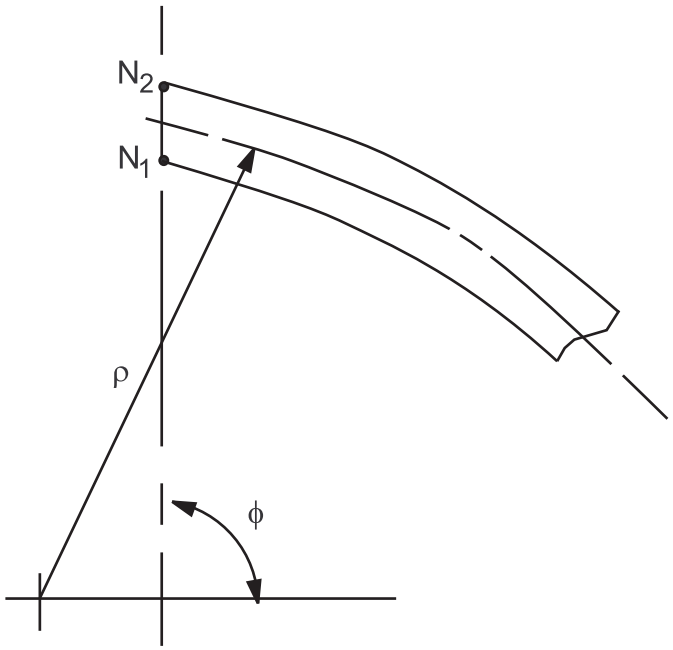


For this configuration it is evident that cos $\phi = R_c/\rho$ as ϕ approaches 90° (or as $N_1' - N_2'$ approaches $N_1 - N_2$). Thus for any paths very near or exactly on the centerline, Equation 17.28 (p. 854) is generalized to be:

$$x_f = \begin{cases} \frac{t^2 \cos\phi}{12R_c} \text{ if } R_c \geq \frac{t}{1000} \\ \frac{t^2}{12\rho} \text{ if } R_c < \frac{t}{1000} \end{cases} \quad (17.49)$$

The second option of Equation 17.49 (p. 858) applied to centerline paths is an accurate representation for spherical/elliptical heads and flat plates. It is incorrect for axisymmetric shapes that do not form perpendicular intersections with the centerline (e.g., conical heads). For such shapes (as shown in Figure 17.9: Non-Perpendicular Intersections (p. 859)) centerline paths must not be selected.

Figure 17.9: Non-Perpendicular Intersections



17.5. POST1 - Fatigue Module

The FATIGUE module of POST1 combines the effects of stress cycling over many cycles involving all stress components at a point in the structure. The procedure is explained in the *Structural Analysis Guide*.

The module automatically calculates all possible stress ranges and keeps track of their number of occurrences, using a technique commonly known as the "rain flow" range-counting method. At a selected nodal location, a search is made throughout all of the events for the pair of loadings (stress vectors) that produces the most severe stress-intensity range. The number of repetitions possible for this range is recorded, and the remaining number of repetitions for the events containing these loadings is decreased accordingly. At least one of the source events will be "used up" at this point; remaining occurrences of stress conditions belonging to that event will subsequently be ignored. This process continues until all ranges and numbers of occurrences have been considered.

The fatigue calculations rely on the ASME Boiler and Pressure Vessel Code, Section III (and Section VIII, Division 2)([60] (p. 923)) for guidelines on range counting, simplified elastic-plastic adaptations, and cumulative fatigue summations by Miner's rule.

The following steps are performed for the fatigue calculations (initiated by the **FTCALC** command).

1. Each loading is compared to each other loading to compute a maximum alternating shear stress:

   A. First, a vector of stress differences is computed:

   $$\{\sigma\}_{i,j} = \{\sigma\}_i - \{\sigma\}_j \tag{17.50}$$

   where:

   $\{\sigma\}_i$ = stress vector for loading ℓ_i

   $\{\sigma\}_j$ = stress vector for loading ℓ_j

B. Second, a stress intensity (σ_I (i,j)) is computed based on $\{\sigma\}_{i,j}$, using

$$\sigma_I = \text{MAX}(|\sigma_1 - \sigma_2|\,|\sigma_2 - \sigma_3|\,|\sigma_3 - \sigma_1|)$$

C. Then, the interim maximum alternating shear stress is:

$$\sigma^d_{i,j} = \frac{\sigma_I(i,j)}{2} \tag{17.51}$$

D. The maximum alternating shear stress is calculated as:

$$\sigma^c_{i,j} = K_e \sigma^d_{i,j} \tag{17.52}$$

where K_e is determined by:

Analysis Type	Range	K_e
ELASTIC (based on peak stresses)	All	1.0
SIMPLIFIED ELASTIC PLASTIC (based on linearized stress components)	$\sigma_n < 3\ S_m$	1.0
	$3\ S_m < \sigma_n < 3\ m\ S_m$	$1.0 + \frac{(1-n)}{n(m-1)}\left(\frac{\sigma_n}{3S_m} - 1\right)$
	$3\ m\ S_m < \sigma_n$	$\frac{1.0}{n}$

where:

σ_n = a stress intensity equivalent of 2 σ^d_{ij} except that it is based on linearized stresses (based on the output of the **FSSECT** command), not actual stresses. (Note that nomenclature is not the same in POST1 - Stress Linearization (p. 849) as in this section.)
S_m = design stress-intensity obtained from the S_m versus temperature table. (The table is input using the **FP** commands inputting *Sm1* to *Sm10* and *T1* to *T10*).
m = first elastic-plastic material parameter (input as *M* on **FP** command) (m >1.0)
n = second elastic-plastic material parameter (input as *N* on **FP** command) (0.0 < n < 1.0)

2. There are a total of (L/2) (L-1) loading case combinations, where L is the number of loadings. These loadings are then sorted (the rain flow method), with the highest value of $\sigma^c_{i,j}$ first.

3. Designate the highest value of $\sigma^c_{i,j}$ as occurring with loading ℓ_i, event k_i together with loading ℓ_j, event k_j. Let MT be the minimum number of times that either event k_i or event k_j is expected to occur. Compute a usage factor following Miner's rule as:

$$f_u = \frac{M_T}{M_A} \tag{17.53}$$

where:

f_u = usage factor (output as PARTIAL USAGE)

M_A = number of allowable cycles at this stress amplitude level. Obtained by entering the allowable alternating stress amplitude (S_a) versus cycles (N) table from the S_a axis and reading the allowable number of cycles M_A corresponding to $\sigma_{i,j}^c$. (The table is input using the **FP** commands inputting *S1* to *S20* for S_a and *N1* to *N20* for N).

Next, cumulatively add f_u to f_u^c where f_u^c = output as CUMULATIVE FATIGUE USAGE. Then decrease the number of possible occurrences of both event k_i and event k_j by M_T (so that one of them becomes zero).

4. Repeat step 3, using the next highest value of $\sigma_{i,j}^c$ until all of the $\sigma_{i,j}^c$ values have been exhausted. It may be seen that the number of times this cycle is performed is equal to the number of events (or less).

17.6. POST1 - Electromagnetic Macros

Electromagnetic macros are macro files created to perform specific postprocessing operations for electromagnetic field analysis. Macros performing computational analysis are detailed in this section.

17.6.1. Flux Passing Thru a Closed Contour

The flux passing through a surface defined by a closed line contour (**PPATH** command) is computed (using the **FLUXV** command macro). The macro is applicable to 2-D and 3-D magnetic field analysis employing the magnetic vector potential A. For 2-D planar analyses, the flux value is per unit depth.

The flux passing through a surface S can be calculated as:

$$\phi = \int_{\text{area}} \{B\} \cdot \{n\} d(\text{area}) \tag{17.54}$$

where:

ϕ = flux enclosed by the bounding surface S
{B} = flux density vector
{n} = unit normal vector
area = area of the bounding surface S

Equation 17.54 (p. 861) can be rewritten in terms of the definition of the vector potential as:

$$\phi = \int_{\text{area}} (\nabla \times \{A\}) \cdot \{n\} d(\text{area}) \tag{17.55}$$

where:

{A} = magnetic vector potential

By applying Stokes theorem, the surface integral reduces to a line integral of A around a closed contour;

$$\phi = \int_{\ell} \{A\} \cdot d\ell \tag{17.56}$$

where:

ℓ = length of the bounding contour line

The macro interpolates values of the vector potential, A, to the closed contour path (defined by the **PPATH** command) and integrates to obtain the flux using Equation 17.56 (p. 861). In the axisymmetric case, the vector potential is multiplied by $2\pi r$ to obtain the total flux for a full circumferential surface (where "r" is the x-coordinate location of the interpolation point).

17.6.2. Force on a Body

The force on a body is evaluated using the Maxwell stress tensor([77] (p. 924)) (with the command macro **FOR2D**). The Maxwell stress approach computes local stress at all points of a bounding surface and then sums the local stresses by means of a surface integral to find the net force on a body. The force can be expressed as:

$$\{F^{mx}\} = \frac{1}{\mu}\int_{area} [T]\cdot\{n\}d(area) \tag{17.57}$$

where:

$\{F^{mx}\}$ = total force vector on the body
[T] = Maxwell stress tensor (see equation 5.126)
μ = permeability of the bounding region

In 2-D planar analyses the surface integral reduces to a line integral and the resulting force is per unit depth. The macro requires a pre-specified path (**PPATH** command) to create the bounding surface. The bounding surface (or line path) should encompass the body for which the force is to be calculated. In principle, the bounding surface (line) is the surface of the body itself. However, in practice it is common to place the path within the air domain surrounding the body. This is perfectly satisfactory and does not violate the principle of the Maxwell stress tensor since the air carries no current and has no magnetic properties different from free space.

The macro interpolates values of flux density, B, to the path (defined by the **PPATH** command) and integrates to obtain the force on the body as in Equation 17.57 (p. 862).

17.6.3. Magnetomotive Forces

The magnetomotive force (current) along a contour or path (defined by the **PPATH** command) is calculated (using the **MMF** command macro) according to Amperes' theorem:

$$I_{mmf} = \int_{\ell}\{H\}\cdot d\ell \tag{17.58}$$

where:

I_{mmf} = magnetomotive force
{H} = magnetic field intensity vector

The macro interpolates values of magnetic field intensity, H, to the path and integrates to obtain the I_{mmf} as in Equation 17.58 (p. 862). In a static analysis or transverse electromagnetic (TEM) and transverse electric (TE) wave guide mode computation, I_{mmf} can be interpreted as a current passing the surface bounded by the closed contour.

17.6.4. Power Loss

The power dissipated in a conducting solid body under the influence of a time-harmonic electromagnetic field is computed (using the **POWERH** command macro). The r.m.s. power loss is calculated from the equation (see Harmonic Analysis Using Complex Formalism (p. 187) for further details):

$$P_{rms} = \frac{1}{2}\int_{vol} \rho \left|\tilde{J}_t\right|^2 d(vol) \tag{17.59}$$

where:

P_{rms} = rms power loss
r = material resistivity
J_t = total current density
~ = complex quantity

The macro evaluates Equation 17.59 (p. 863) by integrating over the selected element set according to:

$$P_{rms} = \frac{1}{2} Re\left\{\sum_{i=1}^{n} (([\rho_i]\{\tilde{J}_{ti}\})^* \cdot \{\tilde{J}_{ti}\}) vol_i\right\} \tag{17.60}$$

where:

n = number of elements
Re{ } = real component of a complex quantity
$[\rho_i]$ = resistivity tensor (matrix)
$\{\tilde{J}_{ti}\}$ = total eddy current density vector for element i
vol_i = element volume
* = complex conjugate operator

For 2-D planar analyses, the resulting power loss is per unit depth.

17.6.5. Terminal Parameters for a Stranded Coil

The terminal parameter quantities for a stranded coil with a d.c. current are computed (using the command macro SRCS). The macro is applicable to linear magnetostatic analysis. In addition, the far-field boundary of the model must be treated with either a flux-normal (Neumann condition), flux-parallel (Dirichlet condition), or modelled with infinite elements.

17.6.6. Energy Supplied

The energy supplied to the coil for a linear system is calculated as:

$$W = \frac{1}{2}\int_{vol} \{A\} \cdot \{J_s\} d(vol) \tag{17.61}$$

where:

W = energy input to coil
{A} = nodal vector potential
$\{J_s\}$ = d.c. source current density
vol = volume of the coil

17.6.7. Terminal Inductance

The inductance as seen by the terminal leads of the coil is calculated as:

$$L = \frac{2W}{i^2} \tag{17.62}$$

where:

L = terminal inductance
i = coil current (per turn)

17.6.8. Flux Linkage

The total flux linkage of a coil can be calculated from the terminal inductance and coil current,

$$\lambda = Li \tag{17.63}$$

where:

λ = flux linkage

17.6.9. Terminal Voltage

For a coil operating with an a.c. current at frequency ω (Hz), a voltage will appear at the terminal leads. Neglecting skin effects and saturation, a static analysis gives the correct field distribution. For the assumed operating frequency, the terminal voltage can be found. From Faraday's law,

$$u = \frac{d\lambda}{dt} \tag{17.64}$$

where:

u = terminal voltage

Under a sinusoidal current at an operating frequency ω, the flux linkage will vary sinusoidally

$$\lambda = \lambda_m \sin \omega t \tag{17.65}$$

where:

λ_m = zero-to-peak magnitude of the flux linkage

The terminal voltage is therefore:

$$u = U \cos \omega t \tag{17.66}$$

where:

$U = \omega \lambda_m$ = zero-to-peak magnitude of the terminal voltage (parameter VLTG returned by the macro)

For 2-D planar analyses, the results are per unit depth.

17.6.10. Torque on a Body

The torque on a body for a 2-D planar analysis is computed by making use of the Maxwell stress tensor (Coulomb([168] (p. 930))) (using the **TORQ2D** and **TORQC2D** command macros) . The torque integrand is evaluated at all points of a bounding surface about the body, and then summed to find the net torque on the body. The torque can be expressed as:

$$\{T\} = \frac{1}{\mu}\int_{area}\left[(\{B\}\cdot\{n\})(\{R\}\times\{B\}) - \frac{|B|^2}{2}(\{R\}\times\{n\})\right]d(area) \tag{17.67}$$

where:

{T} = total torque on a body
μ = permeability of the bounding region
{B} = flux density vector
{n} = unit normal vector to the path
{R} = position vector
area = area of the bounding surface

In 2-D planar analyses, the surface integral reduces to a line integral and the torque results are per unit depth. When a pre-specified path (using the **PPATH** command) is needed to create the bounding surface, a general procedure is used (using the **TORQ2D** command macro). The bounding surface (or line path) should encompass the body for which the torque, about the global origin, is to be calculated.

In principle the bounding surface (line) is the surface of the body itself. However, in practice, it is common to place the path within the air domain surrounding the body. This is perfectly satisfactory and does not violate the principle of the Maxwell stress tensor since the air carries no current and has no magnetic properties different from free space.

A simplified procedure (using the command macro **TORQC2D**) is available when a circular bounding surface (line) about the global origin can be used. This macro creates its own path for evaluation. For the case of a circular path, Equation 17.67 (p. 865) reduces to:

$$\{T\} = \frac{1}{\mu}\int_{area}[M(\{B\}\cdot\{n\})(\{R\}\times\{B\})]d(area) \tag{17.68}$$

The macro **TORQC2D** makes use of Equation 17.68 (p. 865) to evaluate torque.

For both torque macros, flux density, B, is interpolated to the path and integrated according to Equation 17.67 (p. 865) or Equation 17.68 (p. 865) to obtain the torque on a body.

17.6.11. Energy in a Magnetic Field

The stored energy and co-energy in a magnetic field are calculated (by the **SENERGY** command macro). For the static or transient analysis, the stored magnetic energy is calculated as:

$$W_s = \int_0^B \{H\}\cdot\{dB\} \tag{17.69}$$

where:

W_s = stored magnetic energy

The magnetic co-energy is calculated as:

$$W_c = \int_{-H_c}^{H} \{B\} \cdot \{dH\} \tag{17.70}$$

where:

W_c = stored magnetic co-energy
H_c = coercive force

For time-harmonic analysis, the r.m.s. stored magnetic energy is calculated as:

$$W_{rms} = \mathrm{Re} \frac{1}{4} \int \{\tilde{B}\} \cdot \{\tilde{H}\}^* d(vol) \tag{17.71}$$

where:

W_{rms} = r.m.s. stored energy

For 2-D planar analyses, the results are per unit depth.

17.6.12. Relative Error in Electrostatic or Electromagnetic Field Analysis

The relative error in an electrostatic or electromagnetic field analysis is computed (by the **EMAGERR** command macro). The relative error measure is based on the difference in calculated fields between a nodal-averaged continuous field representation and a discontinuous field represented by each individual element's-nodal field values. An average error for each element is calculated. Within a material, the relative error is calculated as:

17.6.12.1. Electrostatics

17.6.12.1.1. Electric Field

$$E_{ei} = \frac{1}{n} \sum_{j=1}^{n} \left| E_j - E_{ij} \right| \tag{17.72}$$

where:

E_{ei} = relative error for the electric field (magnitude) for element i
E_j = nodal averaged electric field (magnitude)
E_{ij} = electric field (magnitude) of element i at node j
n = number of vertex nodes in element i

17.6.12.1.2. Electric Flux Density

$$D_{ei} = \frac{1}{n} \sum_{j=1}^{n} \left| D_j - D_{ij} \right| \tag{17.73}$$

where:

D_{ei} = relative error for the electric flux density (magnitude) for element i
D_j = nodal averaged electric flux density (magnitude)
D_{ij} = electric flux density (magnitude) of element i at node j

A normalized relative error norm measure is also calculated based on the maximum element nodal calculated field value in the currently selected element set.

$$E_{nei} = E_{ei} / E_{max} \quad (17.74)$$

where:

E_{max} = maximum element nodal electric field (magnitude)

$$D_{nei} = D_{ei} / D_{max} \quad (17.75)$$

where:

D_{max} = maximum element nodal electric flux density (magnitude)

17.6.12.2. Electromagnetics

17.6.12.2.1. Magnetic Field Intensity

$$H_{ei} = \frac{1}{n} \sum_{j=1}^{n} \left| H_j - H_{ij} \right| \quad (17.76)$$

where:

H_{ei} = relative error for the magnetic field intensity (magnitude) for element i
H_j = nodal averaged magnetic field intensity (magnitude)
H_{ij} = magnetic field intensity (magnitude) of element i at node j

17.6.12.2.2. Magnetic Flux Density

$$B_{ei} = \frac{1}{n} \sum_{j=1}^{n} \left| B_j - B_{ij} \right| \quad (17.77)$$

where:

B_{ei} = relative error for the magnetic flux density (magnitude) for element i
B_j = nodal averaged magnetic flux density (magnitude)
B_{ij} = magnetic flue density (magnitude) of element i at node j

A normalized relative error measure is also calculated based on the maximum element nodal calculated field value in the currently selected element set.

$$H_{nei} = H_{ei} / H_{max} \quad (17.78)$$

where:

H_{max} = maximum element nodal magnetic field intensity (magnitude)

$$B_{nei} = B_{ei} / B_{max} \quad (17.79)$$

where:

B_{max} = maximum nodal averaged magnetic flux density (magnitude)

17.6.13. Electromotive Force

The electromotive force (voltage drop) between two conductors defined along a path contour (**PATH** command) is computed (using the **EMF** command macro):

$$V_{emf} = \int_{\ell} \{E\} \cdot d\ell \tag{17.80}$$

where:

V_{emf} = electromotive force (voltage drop)
{E} = electric field vector

The macro interpolates values of the electric field, E, to the path (defined by the **PATH** command) and integrates to obtain the electromotive force (voltage drop). The path may span multiple materials of differing permittivity. At least one path point should reside in each material transversed by the path. In static analysis or transverse electromagnetic (TEM) and transverse magnetic (TM) wave guide mode computation, V_{emf} can be interpreted as a voltage drop.

17.6.14. Impedance of a Device

The impedance of a device from the calculated V_{emf} and I_{mmf} values is calculated (using the **IMPD** macro). Impedance calculations are valid for transverse electromagnetic (TEM) waves in coaxial waveguide structures. The impedance is calculated as:

$$Z = \frac{V_{emf}^{Re} + jV_{emf}^{Im}}{I_{mmf}^{Re} + jI_{mmf}^{Im}} \tag{17.81}$$

where:

V and I = voltage drop and current, respectively
Re and Im = represent real and imaginary parts of complex terms
V_{emf} = voltage drop (computed with the **EMF** macro)
I_{mmf} = current (computed by the **MMF** macro)

17.6.15. Computation of Equivalent Transmission-line Parameters

The equivalent transmission-line parameters for a guiding wave structure are calculated. For a lossless guiding structure, the total mode voltage, V(Z), and mode current, I(Z), associated with a +Z propagating field take on the form:

$$V(Z) = Ae^{-j\beta Z} + Be^{j\beta Z} \tag{17.82}$$

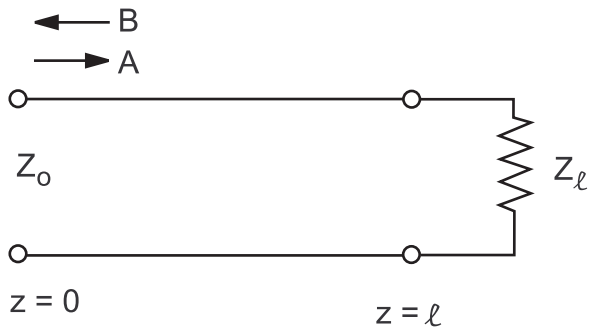
$$I(Z) = \frac{A}{Z_o} e^{-j\beta Z} - \frac{B}{Z_o} e^{j\beta Z} \tag{17.83}$$

where:

Z_o = characteristic impedance for any mode
A = amplitude of the incident voltage wave (see below)
B = amplitude of the backscattered voltage wave (see below)

We can consider the propagating waves in terms on an equivalent two-wire transmission line terminated at Z = ℓ by a load impedance Z_ℓ.

Figure 17.10: Equivalent Two-Wire Transmission Line



The voltage term "A" in Equation 17.82 (p. 868) can be considered as the amplitude of the incident wave, and voltage term "B" as the amplitude of the mode voltage wave backscattered off the load impedance Z_ℓ.

Thus,

$$Z_\ell = Z_o \frac{Ae^{-j\beta\ell} + Be^{j\beta\ell}}{Ae^{-j\beta\ell} - Be^{j\beta\ell}} \tag{17.84}$$

Rearranging we have,

$$\Gamma = \frac{B}{A} = \frac{(Z_\ell/Z_o) - 1}{(Z_\ell/Z_o) + 1} \tag{17.85}$$

where:

Γ = voltage reflection coefficient

The voltage standing-wave ratio is calculated as:

$$S = \frac{1 + |\Gamma|}{1 - |\Gamma|} \tag{17.86}$$

where:

S = voltage standing-wave ratio (output as VSWR)

For a matched load ($Z_\ell = Z_o$) there is no reflection ($\Gamma = 0$) and the S = 1. If Z_ℓ is a short circuit, B = -A, Γ = -1, and the S is infinite. If Z_ℓ is an open circuit, B = A, Γ = +1, and the S once again is infinite.

The reflection coefficient is frequently expressed in dB form by introducing the concept of return loss defined by:

$$L_R = -20 \log_{10} |\Gamma| \tag{17.87}$$

where:

L_R = return loss in dB (output as RL)

The macro calculates the above transmission line parameters in terms of the incident, reference and total voltage.

17.7. POST1 - Error Approximation Technique

17.7.1. Error Approximation Technique for Displacement-Based Problems

The error approximation technique used by POST1 (**PRERR** command) for displacement-based problems is similar to that given by Zienkiewicz and Zhu([102] (p. 926)). The essentials of the method are summarized below.

The usual continuity assumption used in many displacement based finite element formulations results in a continuous displacement field from element to element, but a discontinuous stress field. To obtain more acceptable stresses, averaging of the element nodal stresses is done. Then, returning to the element level, the stresses at each node of the element are processed to yield:

$$\{\Delta\sigma_n^i\} = \{\sigma_n^a\} - \{\sigma_n^i\} \tag{17.88}$$

where:

$\{\Delta\sigma_n^i\}$ = stress error vector at node n of element i

$\{\sigma_n^a\}$ = averaged stress vector at node n = $\dfrac{\sum_{i=1}^{N_e^n}\{\sigma_n^i\}}{N_e^n}$

N_e^n = number of elements connecting to node n

$\{\sigma_n^i\}$ = stress vector of node n of element i

Then, for each element

$$e_i = \frac{1}{2}\int_{vol}\{\Delta\sigma\}^T[D]^{-1}\{\Delta\sigma\}d(vol) \tag{17.89}$$

where:

e_i = energy error for element i (accessed with **ETABLE** (SERR item) command)
vol = volume of the element (accessed with **ETABLE** (VOLU item) command)
[D] = stress-strain matrix evaluated at reference temperature
$\{\Delta\sigma\}$ = stress error vector at points as needed (evaluated from all $\{\Delta\sigma_n\}$ of this element)

The energy error over the model is:

$$e = \sum_{i=1}^{N_r} e_i \tag{17.90}$$

where:

e = energy error over the entire (or part of the) model (accessed with ***GET** (SERSM item) command)
N_r = number of elements in model or part of model

The energy error can be normalized against the strain energy.

$$E = 100\left(\frac{e}{U+e}\right)^{\frac{1}{2}} \tag{17.91}$$

where:

E = percentage error in energy norm (accessed with **PRERR**, **PLDISP**, **PLNSOL** (U item), ***GET** (SEPC item) commands)
U = strain energy over the entire (or part of the) model (accessed with ***GET** (SENSM item) command)

$$= \sum_{i=1}^{N_r} E_{ei}^{po}$$

E_{ei}^{po} = strain energy of element i (accessed with **ETABLE** (SENE item) command) (see Energies)

The e_i values can be used for adaptive mesh refinement. It has been shown by Babuska and Rheinboldt([103] (p. 926)) that if e_i is equal for all elements, then the model using the given number of elements is the most efficient one. This concept is also referred to as "error equilibration".

At the bottom of all printed nodal stresses (the **PRNSOL** or **PRESOL** command), which may consist of the 6 component stresses, the 5 combined stresses, or both, a summary printout labeled: ESTIMATED BOUNDS CONSIDERING THE EFFECT OF DISCRETIZATION ERROR gives minimum nodal values and maximum nodal values. These are:

$$\sigma_j^{mnb} = \min(\sigma_{j,n}^{a} - \Delta\sigma_n) \tag{17.92}$$

$$\sigma_j^{mxb} = \max(\sigma_{j,n}^{a} + \Delta\sigma_n) \tag{17.93}$$

where min and max are over the selected nodes, and

where:

σ_j^{mnb} = nodal minimum of stress quantity (output as VALUE (printout) or SMNB (plot))
σ_j^{mxb} = nodal maximum of stress quantity (output as VALUE (printout) or SMXB (plot))
j = subscript to refer to either a particular stress component or a particular combined stress

$$\sigma_{j,n}^{a} = \begin{cases} \sigma_{j,n}^{avg} & \text{if nodal quantities (}\textbf{PLNSOL}\text{ or }\textbf{PRNSOL}\text{ command) are used} \\ \sigma_{j,n}^{max} & \text{if element quantities (}\textbf{PLESOL}\text{ command) are used} \end{cases}$$

$\sigma_{j,n}^{avg}$ = average of stress quantity j at node n of element attached to node n
$\sigma_{j,n}^{max}$ = maximum of stress quantity j at node n of element attached to node n
$\Delta\sigma_n$ = root mean square of all $\Delta\sigma_i$ from elements connecting to node n

$\Delta\sigma_i$ = maximum absolute value of any component of $\{\Delta\sigma_n^i\}$ for all nodes connecting to element (accessed with **ETABLE** (SDSG item) command)

17.7.2. Error Approximation Technique for Temperature-Based Problems

The error approximation technique used by POST1 (**PRERR** command) for temperature based problems is similar to that given by Huang and Lewis([126] (p. 927)). The essentials of the method are summarized below.

The usual continuity assumption results in a continuous temperature field from element to element, but a discontinuous thermal flux field. To obtain more acceptable fluxes, averaging of the element nodal thermal fluxes is done. Then, returning to the element level, the thermal fluxes at each node of the element are processed to yield:

$$\{\Delta q_n^i\} = \{q_n^a\} - \{q_n^i\} \tag{17.94}$$

where:

$\{\Delta q_n^i\}$ = thermal flux error vector at node n of element i

$\{q_n^a\}$ = averaged thermal flux vector at node n = $\dfrac{\sum_{i=1}^{N_e^n}\{q_n^i\}}{N_e^n}$

N_e^n = number of elements connecting to node n

$\{q_n^i\}$ = thermal flux vector of node n of element

Then, for each element

$$e_i = \frac{1}{2}\int_{vol}\{\Delta q\}^T[D]^{-1}\{\Delta q\}d(vol) \tag{17.95}$$

where:

e_i = energy error for element i (accessed with **ETABLE** (TERR item) command)
vol = volume of the element (accessed with **ETABLE** (VOLU item) command)
[D] = conductivity matrix evaluated at reference temperature
$\{\Delta q\}$ = thermal flux error vector at points as needed (evaluated from all $\{\Delta q_n\}$ of this element)

The energy error over the model is:

$$e = \sum_{i=1}^{N_r} e_i \tag{17.96}$$

where:

e = energy error over the entire (or part of the) model (accessed with ***GET** (TERSM item) command)
N_r = number of elements in model or part of model

The energy error can be normalized against the thermal dissipation energy.

$$E = 100\left(\frac{e}{U+e}\right)^{\frac{1}{2}} \tag{17.97}$$

where:

E = percentage error in energy norm (accessed with **PRERR**, **PLNSOL**, (TEMP item) or ***GET** (TEPC item) commands)
U = thermal dissipation energy over the entire (or part of the) model (accessed with ***GET** (TENSM item) command)

$$= \sum_{i=1}^{N_r} E_{ei}^{po}$$

E_{ei}^{po} = thermal dissipation energy of element i (accessed with **ETABLE** (TENE item) command) (see Energies)

The e_i values can be used for adaptive mesh refinement. It has been shown by Babuska and Rheinboldt([103] (p. 926)) that if e_i is equal for all elements, then the model using the given number of elements is the most efficient one. This concept is also referred to as "error equilibration".

At the bottom of all printed fluxes (with the **PRNSOL** command), which consists of the 3 thermal fluxes, a summary printout labeled: ESTIMATED BOUNDS CONSIDERING THE EFFECT OF DISCRETIZATION ERROR gives minimum nodal values and maximum nodal values. These are:

$$q_j^{mnb} = \min(q_{j,n}^a - \Delta q_n) \tag{17.98}$$

$$q_j^{mxb} = \max(q_{j,n}^a + \Delta q_n) \tag{17.99}$$

where min and max are over the selected nodes, and

where:

q_j^{mnb} = nodal minimum of thermal flux quantity (output as VALUE (printout) or SMNB (plot))
q_j^{mxb} = nodal maximum of thermal flux quantity (output as VALUE (printout) or SMXB (plot))
j = subscript to refer to either a particular thermal flux component or a particular combined thermal flux

$$q_{j,n}^a = \begin{cases} q_{j,n}^{avg} & \text{if nodal quantities (}\textbf{PLNSOL}\text{ or }\textbf{PRNSOL}\text{ command) are used} \\ q_{j,n}^{max} & \text{if element quantities (}\textbf{PLESOL}\text{ command) are used} \end{cases}$$

$q_{j,n}^{avg}$ = average of thermal flux quantity j at node n of element attached to node n
$q_{j,n}^{max}$ = maximum of thermal flux quantity j at node n of element attached to node n
Δq_n = maximum of all Δq_i from elements connecting to node n

Δq_i = maximum absolute value of any component of $\{\Delta q_n^i\}$ for all nodes connecting to element (accessed with **ETABLE** (TDSG item) command)

17.7.3. Error Approximation Technique for Magnetics-Based Problems

The error approximation technique used by POST1 (**PRERR** command) for magnetics- based problems is similar to that given by Zienkiewicz and Zhu ([102] (p. 926)) and Huang and Lewis ([126] (p. 927)). The essentials of the method are summarized below.

The usual continuity assumption results in a continuous temperature field from element to element, but a discontinuous magnetic flux field. To obtain more acceptable fluxes, averaging of the element nodal magnetic fluxes is done. Then, returning to the element level, the magnetic fluxes at each node of the element are processed to yield:

$$\{\Delta B_n^i\} = \{B_n^a\} - \{B_n^i\} \tag{17.100}$$

where:

$\{\Delta B_n^i\}$ = magnetic flux error vector at node n of element i

$\{B_n^a\}$ = averaged magnetic flux vector at node n = $\dfrac{\sum_{i=1}^{N_e^n} \{B_n^i\}}{N_e^n}$

N_e^n = number of elements connecting to node n

$\{B_n^i\}$ = magnetic flux vector of node n of element

Then, for each element

$$e_i = \frac{1}{2}\int_{vol} \{\Delta B\}^T [D]^{-1} \{\Delta B\} d(vol) \tag{17.101}$$

where:

e_i = energy error for element i (accessed with **ETABLE** (BERR item) command)
vol = volume of the element (accessed with **ETABLE** (VOLU item) command)
[D] = magnetic conductivity matrix evaluated at reference temperature
$\{\Delta B\}$ = magnetic flux error vector at points as needed (evaluated from all $\{\Delta B_n\}$ of this element)

The energy error over the model is:

$$e = \sum_{i=1}^{N_r} e_i \tag{17.102}$$

where:

e = energy error over the entire (or part of the) model (accessed with ***GET** (BERSM item) command)
N_r = number of elements in model or part of model

The energy error can be normalized against the magnetic energy.

$$E = 100\left(\frac{e}{U+e}\right)^{\frac{1}{2}} \tag{17.103}$$

where:

E = percentage error in energy norm (accessed with **PRERR**, **PLNSOL**, (TEMP item) or ***GET** (BEPC item) commands)
U = magnetic energy over the entire (or part of the) model (accessed with ***GET** (BENSM item) command)

$= \sum_{i=1}^{N_r} E_{ei}^{po}$

E_{ei}^{po} = magnetic energy of element i (accessed with **ETABLE** (SENE item) command) (see Energies)

The e_i values can be used for adaptive mesh refinement. It has been shown by Babuska and Rheinboldt([103] (p. 926)) that if e_i is equal for all elements, then the model using the given number of elements is the most efficient one. This concept is also referred to as "error equilibration".

At the bottom of all printed fluxes (with the **PRNSOL** command), which consists of the 3 magnetic fluxes, a summary printout labeled: ESTIMATED BOUNDS CONSIDERING THE EFFECT OF DISCRETIZATION ERROR gives minimum nodal values and maximum nodal values. These are:

$$B_j^{mnb} = \min(B_{j,n}^a - \Delta B_n) \tag{17.104}$$

$$B_j^{mxb} = \max(B_{j,n}^a + \Delta B_n) \tag{17.105}$$

where min and max are over the selected nodes, and

where:

B_j^{mnb} = nodal minimum of magnetic flux quantity (output as VALUE (printout))

B_j^{mxb} = nodal maximum of magnetic flux quantity (output as VALUE (printout))
j = subscript to refer to either a particular magnetic flux component or a particular combined magnetic flux

$$B_{j,n}^a = \begin{cases} B_{j,n}^{avg} & \text{if nodal quantities (}\textbf{PLNSOL}\text{ or }\textbf{PRNSOL}\text{ command) are used} \\ B_{j,n}^{max} & \text{if element quantities (}\textbf{PLESOL}\text{ command) are used} \end{cases}$$

$B_{j,n}^{avg}$ = average of magnetic flux quantity j at node n of element attached to node n

$B_{j,n}^{max}$ = maximum of magnetic flux quantity j at node n of element attached to node n
ΔB_n = maximum of all ΔB_i from elements connecting to node n

ΔB_i = maximum absolute value of any component of $\{\Delta B_n^i\}$ for all nodes connecting to element (accessed with **ETABLE** (BDSG item) command)

17.8. POST1 - Crack Analysis

The stress intensity factors at a crack for a linear elastic fracture mechanics analysis may be computed (using the **KCALC** command). The analysis uses a fit of the nodal displacements in the vicinity of the

crack. The actual displacements at and near a crack for linear elastic materials are (Paris and Sih([106] (p. 926))):

$$u = \frac{K_I}{4G}\sqrt{\frac{r}{2\pi}}\left((2\kappa-1)\cos\frac{\theta}{2}-\cos\frac{3\theta}{2}\right)-\frac{K_{II}}{4G}\sqrt{\frac{r}{2\pi}}\left((2\kappa+3)\sin\frac{\theta}{2}+\sin\frac{3\theta}{2}\right)+0(r) \quad (17.106)$$

$$v = \frac{K_I}{4G}\sqrt{\frac{r}{2\pi}}\left((2\kappa-1)\sin\frac{\theta}{2}-\sin\frac{3\theta}{2}\right)-\frac{K_{II}}{4G}\sqrt{\frac{r}{2\pi}}\left((2\kappa+3)\cos\frac{\theta}{2}+\cos\frac{3\theta}{2}\right)+0(r) \quad (17.107)$$

$$w = \frac{2K_{III}}{G}\sqrt{\frac{r}{2\pi}}\sin\frac{\theta}{2}+0(r) \quad (17.108)$$

where:

u, v, w = displacements in a local Cartesian coordinate system as shown in Figure 17.11: Local Coordinates Measured From a 3-D Crack Front (p. 877).
r, θ = coordinates in a local cylindrical coordinate system also shown in Figure 17.11: Local Coordinates Measured From a 3-D Crack Front (p. 877).
G = shear modulus
K_I, K_{II}, K_{III} = stress intensity factors relating to deformation shapes shown in Figure 17.12: The Three Basic Modes of Fracture (p. 877)

$$\kappa = \begin{cases} 3-4\nu & \text{if plane strain or axisymmetric} \\ \dfrac{3-\nu}{1+\nu} & \text{if plane stress} \end{cases}$$

ν = Poisson's ratio
0(r) = terms of order r or higher

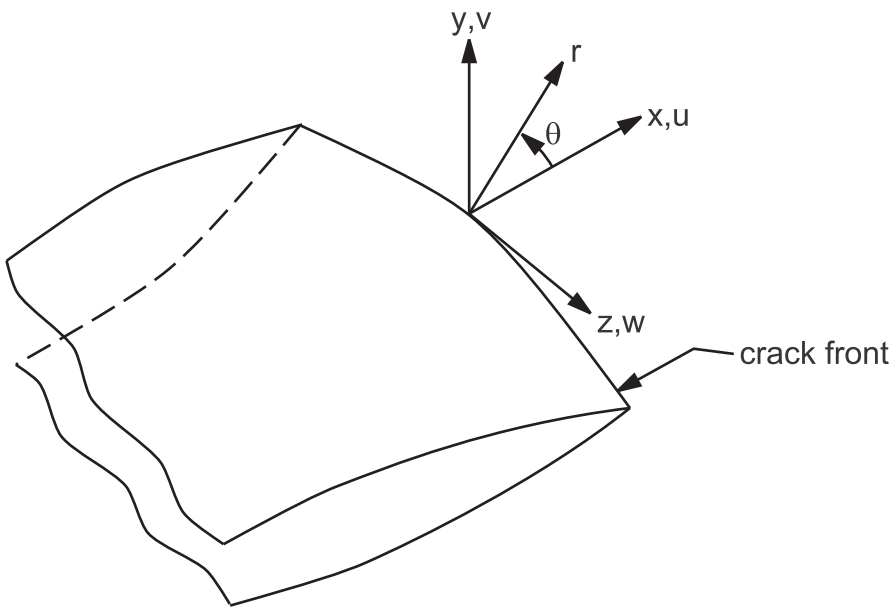
Evaluating Equation 17.106 (p. 876) through Equation 17.108 (p. 876) at $\theta = \pm 180.0°$ and dropping the higher order terms yields:

$$u = +\frac{K_{II}}{2G}\sqrt{\frac{r}{2\pi}}(1+\kappa) \quad (17.109)$$

$$v = +\frac{K_I}{2G}\sqrt{\frac{r}{2\pi}}(1+\kappa) \quad (17.110)$$

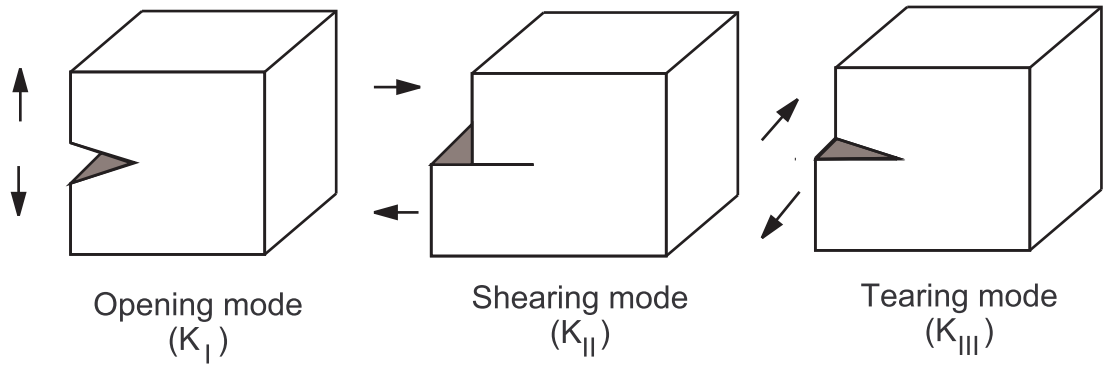
$$w = +\frac{2K_{III}}{G}\sqrt{\frac{r}{2\pi}} \quad (17.111)$$

Figure 17.11: Local Coordinates Measured From a 3-D Crack Front



The crack width is shown greatly enlarged, for clarity.

Figure 17.12: The Three Basic Modes of Fracture



For models symmetric about the crack plane (half-crack model, Figure 17.13: Nodes Used for the Approximate Crack-Tip Displacements (p. 878)(a)), Equation 17.109 (p. 876) to Equation 17.111 (p. 876) can be reorganized to give:

$$K_I = \sqrt{2\pi}\frac{2G}{1+\kappa}\frac{|v|}{\sqrt{r}} \tag{17.112}$$

$$K_{II} = \sqrt{2\pi}\frac{2G}{1+\kappa}\frac{|u|}{\sqrt{r}} \tag{17.113}$$

$$K_{III} = \sqrt{2\pi}2G\frac{|w|}{\sqrt{r}} \tag{17.114}$$

and for the case of no symmetry (full-crack model, Figure 17.13: Nodes Used for the Approximate Crack-Tip Displacements (p. 878)(b)),

$$K_I = \sqrt{2\pi}\frac{G}{1+\kappa}\frac{|\Delta v|}{\sqrt{r}} \tag{17.115}$$

$$K_{II} = \sqrt{2\pi}\frac{G}{1+\kappa}\frac{|\Delta u|}{\sqrt{r}} \tag{17.116}$$

$$K_{III} = \sqrt{2\pi}\frac{G}{1+\kappa}\frac{|\Delta w|}{\sqrt{r}} \tag{17.117}$$

where Δv, Δu, and Δw are the motions of one crack face with respect to the other.

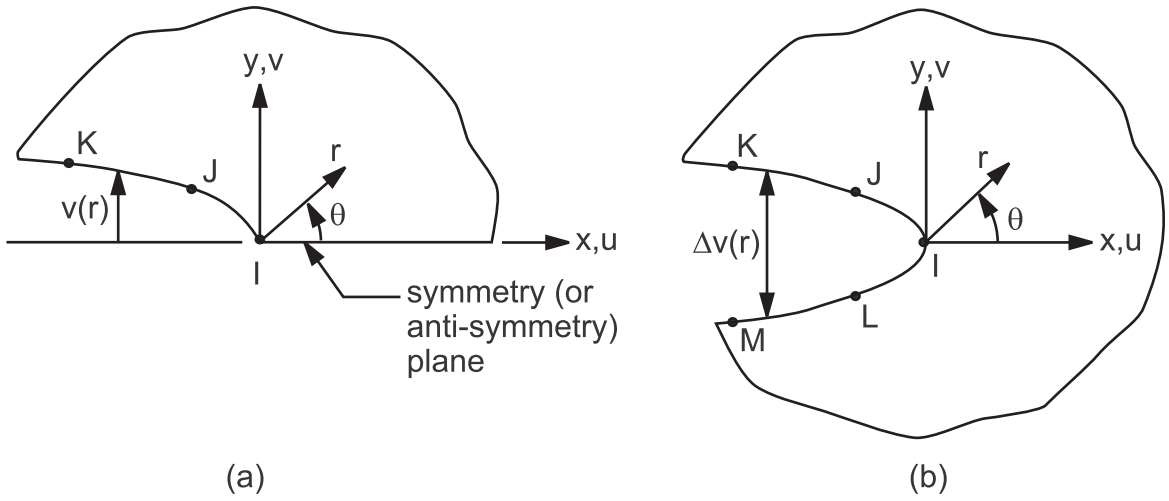
As the above six equations are similar, consider only the first one further. The final factor is $\frac{|v|}{\sqrt{r}}$, which needs to be evaluated based on the nodal displacements and locations. As shown in Figure 17.13: Nodes Used for the Approximate Crack-Tip Displacements (p. 878)(a), three points are available. v is normalized so that v at node I is zero. Then A and B are determined so that

$$\frac{|v|}{\sqrt{r}} = A + Br \tag{17.118}$$

at points J and K. Next, let r approach 0.0:

$$\lim_{r \to 0}\frac{|v|}{\sqrt{r}} = A \tag{17.119}$$

Figure 17.13: Nodes Used for the Approximate Crack-Tip Displacements



(a) Half Model, (b) Full Model

Thus, Equation 17.112 (p. 877) becomes:

$$K_I = \sqrt{2\pi}\frac{2GA}{1+\kappa} \tag{17.120}$$

Equation 17.113 (p. 877) through Equation 17.117 (p. 878) are also fit in the same manner.

17.9. POST1 - Harmonic Solid and Shell Element Postprocessing

As discussed in Axisymmetric Elements with Nonaxisymmetric Loads of the *Element Reference*, results from load cases with different values of mode number (input as MODE on **MODE** command) but at the same angular location (input as ANGLE on the **SET** command) can be combined in POST1 (with the **LCOPER** command). The below assumes values of the mode number and angle and shows how the results are extracted.

17.9.1. Thermal Solid Elements (PLANE75, PLANE78)

Data processed in a harmonic fashion includes nodal temperatures, element data stored on a per node basis (thermal gradient and thermal flux) and nodal heat flow. Nodal temperature is calculated at harmonic angle θ for each node j.

$$T_{j\theta} = FKT_j \tag{17.121}$$

where:

$T_{j\theta}$ = temperature at node j at angle q
F = scaling factor (input as *FACT*, **SET** command)

$$K = \begin{cases} \cos n\theta & \text{if mode is symmetric (input as ISYM=1 on } \textbf{MODE} \text{ command)} \\ \sin n\theta & \text{if mode is antisymmetric (input as ISYM=-1 on } \textbf{MODE} \text{ command)} \end{cases}$$

n = mode number (input as *MODE* on **MODE** command)
θ = angle at which harmonic calculation is being made (input as *ANGLE*, **SET** command)
T_j = temperature at node j from nodal solution

Thermal gradient are calculated at harmonic angle θ for each node j of element i:

$$G^t_{xij\theta} = FKG^t_{xij} \tag{17.122}$$

$$G^t_{yij\theta} = FKG^t_{yij} \tag{17.123}$$

$$G^t_{zij\theta} = FKG^t_{zij} \tag{17.124}$$

where:

$G^t_{xij\theta}$ = thermal gradient in x (radial) direction at node j of element i at angle θ

$$L = \begin{cases} \sin n\theta & \text{if mode is symmetric (input as ISYM=1 on } \textbf{MODE} \text{ command)} \\ \cos n\theta & \text{if mode is antisymmetric (input as ISYM=-1 on } \textbf{MODE} \text{ command)} \end{cases}$$

G^T_{xij} = thermal gradient in x (radial) direction at node j of element i

Nodal heat flow is processed in the same way as temperature. Thermal flux is processed in the same way as thermal gradient.

17.9.2. Structural Solid Elements (PLANE25, PLANE83)

Data processed in a harmonic fashion include nodal displacements, nodal forces, and element data stored on a per node basis (stress and elastic strain).

Nodal displacement is calculated at harmonic angle θ for each node j:

$$u_{xj\theta} = FKu_{xj} \quad (17.125)$$

$$u_{yj\theta} = FKu_{yj} \quad (17.126)$$

$$u_{zj\theta} = FLu_{zj} \quad (17.127)$$

where:

$u_{xj\theta}$ = x (radial) displacement at node j at angle θ
u_{xj} = maximum x (radial) displacement at node j (from nodal solution)

Stress is calculated at harmonic angle θ for each node j of element i:

$$\sigma_{xij\theta} = FK\sigma_{xij} \quad (17.128)$$

$$\sigma_{yij\theta} = FK\sigma_{yij} \quad (17.129)$$

$$\sigma_{zij\theta} = FK\sigma_{zij} \quad (17.130)$$

$$\sigma_{xyij\theta} = FK\sigma_{xyij} \quad (17.131)$$

$$\sigma_{yzij\theta} = FL\sigma_{yzij} \quad (17.132)$$

$$\sigma_{xzij\theta} = FL\sigma_{xzij} \quad (17.133)$$

where:

$\sigma_{xij\theta}$ = x (radial) stress at node j of element i at angle θ
σ_{xij} = maximum x (radial) stress at node j of element i

Nodal forces are processed in the same way as nodal displacements. Strains are processed in the same way as stresses.

17.9.3. Structural Shell Element (SHELL61)

Data processed in a harmonic fashion include displacements, nodal forces, member forces, member moments, in-plane element forces, out-of-plane element moments, stress, and elastic strain.

Nodal displacement is calculated at harmonic angle θ for each node j:

$$u_{xj\theta} = FKu_{xj} \quad (17.134)$$

$$u_{yj\theta} = FKu_{yj} \quad (17.135)$$

$$u_{zj\theta} = FLu_{zj} \quad (17.136)$$

$$\phi_{zj\theta} = FK\phi_{zj} \quad (17.137)$$

where:

$\phi_{zj\theta}$ = rotation about z (hoop) direction at node j at angle θ
ϕ_{zj} = maximum rotation about z (hoop) direction at node j (from nodal solution)

Stress is calculated at harmonic angle θ for each node/interior point j of element i:

$$\sigma_{mij\theta} = FK\sigma_{mij} \quad (17.138)$$

$$\sigma_{hij\theta} = FK\sigma_{hij} \quad (17.139)$$

$$\sigma_{tij\theta} = FK\sigma_{tij} \quad (17.140)$$

$$\sigma_{mhij\theta} = FL\sigma_{mhij} \quad (17.141)$$

where:

$\sigma_{mij\theta}$ = meridional stress at point j of element i at angle θ
σ_{mij} = meridional stress j of element i

In-plane element forces at harmonic angle θ for each node/interior point j of element i:

$$T_{xij\theta} = FKT_{xij} \quad (17.142)$$

$$T_{zij\theta} = FKT_{zij} \quad (17.143)$$

$$T_{xzij\theta} = FLT_{xzij} \quad (17.144)$$

where:

$T_{xij\theta}$ = in-plane element force in x (meridional) direction at point j of element i at angle θ
T_{xij} = maximum in-plane element force in x (meridional) direction at point j of element i

Nodal forces, member forces, and member moments are processed in the same way as nodal displacements. Strains are processed in the same way as stresses. Finally, out-of-plane element moments are processed in the same way as in-plane element forces.

17.10. POST26 - Data Operations

Table 17.1: POST26 Operations (p. 881) shows the operations that can be performed on the time-history data stored by POST26. (Input quantities *FACTA*, *FACTB*, *FACTC*, and table *IC* are omitted from Table 17.1: POST26 Operations (p. 881) for clarity of the fundamental operations.) All operations are performed in complex variables. The operations create new tables which are also complex numbers.

Table 17.1: POST26 Operations

Description	POST26 Command	Real Operation and Result	Complex Operation	Complex Result
Addition	ADD	a + c	(a + ib) + (c + id)	(a + c) + i(b + d)
Multiplication	PROD	a x c	(a + ib) x (c + id)	(ac - bd) + i(ad + bc)
Division	QUOT	a/c	$\frac{(a+ib)}{(c+id)}$	$\frac{(ac+d)+i(-ad+bc)}{(c^2+d^2)}$

Description	POST26 Command	Real Operation and Result	Complex Operation	Complex Result
Absolute Value	ABS	|a|	|a + ib|	$\sqrt{a^2 + b^2}$
Arc Tangent	ATAN	0	atan (a + ib)	atan (b / a)
Square Root	SQRT	$\sqrt{a}$	$\sqrt{a + ib}$	$(a^2 + b^2)^{\frac{1}{2}}\left(\cos\frac{\theta}{2} + i\sin\frac{\theta}{2}\right)$
Largest Variable	LARGE	Maximum of a and c		
Smallest Variable	SMALL	Minimum of a and c		
Derivative	DERIV	da/dc	d(a + ib)/dc	da/dc + i db/dc
Integration	INT1	$\int adc$	$\int (a + ib)dc$	$\int adc + i\int bdc$
Common Logarithm	CLOG	$\log_{10}a$	$\log_{10}(a + ib)$	$\log_{10} e\,(\ell n\sqrt{a^2 + b^2} + i\theta)$
Natural Logarithm	NLOG	ℓ n a	ℓ n (a + ib)	$\ell n\sqrt{a^2 + b^2} + i\theta$
Exponential	EXP	e^a	$e^{(a + ib)}$	$e^a(\cos b + i \sin b)$
Complex Conjugate	CONJUG	a	conj (a + ib)	a - ib
Real Part	REALVAR	a	real (a + ib)	a
Imaginary Part	IMAGIN	0	imag (a + ib)	b
Read Data into Table	DATA	-	-	-
Fill Table with Data	FILL	-	-	-
See Response Spectrum Generator Description (POST26 - Response Spectrum Generator (RESP) (p. 883))				

NOTES:

1. a + ib (from Table *IA or IX*) and c + id (from Table *IB or IY*) are complex numbers, where *IA* and *IB* are input quantities on above commands.
2. $\theta = \tan^{-1}\left(\frac{b}{a}\right)$
3. For derivative and integration, see Integration and Differentiation Procedures (p. 831)

17.11. POST26 - Response Spectrum Generator (RESP)

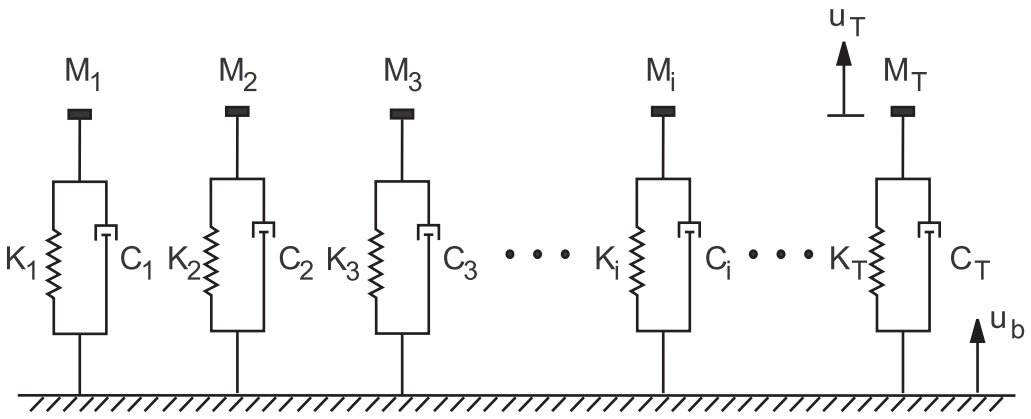
Given a motion as output from a transient dynamic analysis, POST26 generates a response spectrum in terms of displacement, velocity, or acceleration.

A response spectrum is generated by imposing the motion of the point of interest on a series of single-mass oscillators over a period of time and calculating the maximum displacement, velocity, or acceleration. This is illustrated in Figure 17.14: Single Mass Oscillators (p. 883).

In Figure 17.14: Single Mass Oscillators (p. 883), the following definitions are used:

M_i = mass of oscillator i
C_i = damping of oscillator i
K_i = stiffness of oscillator i
u_i = motion of oscillator i
u_b = motion of point of interest

Figure 17.14: Single Mass Oscillators



In the absence of damping, the natural frequency of an oscillator i is:

$$\omega_i = \sqrt{\frac{K_i}{M_i}} \tag{17.145}$$

The basic equation of motion of the oscillator can be given as a one degree of freedom (DOF) version of Equation 15.5 (p. 764):

$$M_i \ddot{u}_i + C_i \dot{u}_i^r + K_i u_i^r = 0 \tag{17.146}$$

where:

a dot ($\dot{}$) over a variable = derivative with respect to time

u_i^r, the relative motion of oscillator i, is defined by:

$$u_i^r = u_i - u_b \tag{17.147}$$

The damping is given by:

$$\xi_i = \frac{C_i}{C_{cr,i}} \tag{17.148}$$

where:

$C_{cr,i} = 2\sqrt{K_i M_i}$ = critical damping coefficient

Equation 17.145 (p. 883) through Equation 17.148 (p. 884) are combined to give:

$$\ddot{u}_i^r + 2\xi_i\omega_i\dot{u}_i^r + \omega_i^2 u_i^r = -\ddot{u}_b \tag{17.149}$$

Equation 17.149 (p. 884) is solved when the input time-history is the acceleration $\ddot{u}_b$ (*inputType* = 1).

For a displacement input u_b (*inputType* = 0), the equation is rewritten as:

$$\ddot{u}_i \quad 2\xi_i\omega_i\dot{u}_i \quad \omega_i^2 u_i \quad 2\xi_i\omega_i\dot{u}_b \quad \omega_i^2 u_b \tag{17.150}$$

where:

$\ddot{u}_b$ is the velocity of the point of interest. It is the derivative of the displacement input.

Both equations are solved using Newmark integration scheme. See Description of Structural and Other Second Order Systems (p. 764) for more details.

Depending on the spectrum type (*specType*), the output spectrum values are the following maximum quantities:

- the relative displacement: $(u_i^r)_{max}$
- the relative velocity: $(\dot{u}_i^r)_{max}$
- the absolute acceleration: $(\ddot{u}_i)_{max}$
- the pseudo-velocity: $\omega(u_i^r)_{max}$
- the pseudo-acceleration: $\omega^2(u_i^r)_{max}$

17.11.1. Time Step Size

The time step size (Δt) is selected in the following way. If data is from a full transient analysis (**ANTYPE**,TRANS with **TRNOPT**,FULL):

Δt = input time step size (input as *DTIME* on **RESP** command)

or if no input is provided:

$$\Delta t = \frac{1}{20 f_{max}} \tag{17.151}$$

where:

f_{max} = highest value of frequency table (table input using *LFTAB* on the **RESP** command

The transient data from full transient analysis (**ANTYPE**,TRANS with **TRNOPT**, FULL analysis) is taken from the next available time step used in the analysis. This can cause a decrease in accuracy at higher frequencies if Δt is less than the time step size of the input transient.

17.12. POST1 and POST26 - Interpretation of Equivalent Strains

The equivalent strains for the elastic, plastic, creep and thermal strains are computed in postprocessing using the von Mises equation:

$$\varepsilon_{eq} = \frac{1}{\sqrt{2}(1+\nu')}\left[(\varepsilon_x-\varepsilon_y)^2+(\varepsilon_y-\varepsilon_z)^2+(\varepsilon_z-\varepsilon_x)^2+\frac{3}{2}(\gamma_{xy}^2+\gamma_{yz}^2+\gamma_{xz}^2)\right]^{\frac{1}{2}} \quad (17.152)$$

where:

ε_x, ε_y, etc. = appropriate component strain values
ν' = effective Poisson's ratio

The default effective Poisson's ratio for both POST1 and POST26 are:

$$= \begin{cases} \text{material Poisson's ratio for elastic and thermal strains} \\ \text{0.5 for plastic, creep, and hyperelastic strains} \\ \text{0.0 for line elements, cyclic symmetry analyses, and load case operations} \end{cases}$$

The **AVPRIN**,,EFFNU command may be issued to override the above defaults (but it is intended to be used only for line elements, etc.).

The equivalent strain is output with the EQV or PRIN component label in POST1 (using the **PRNSOL**, **PLNSOL**, **PDEF**, or **ETABLE** commands) and in POST26 (using the **ESOL** command).

17.12.1. Physical Interpretation of Equivalent Strain

The von Mises equation is a measure of the "shear" strain in the material and does not account for the hydrostatic straining component. For example, strain values of $\varepsilon_x = \varepsilon_y = \varepsilon_z = 0.001$ yield an equivalent strain $\varepsilon_{eq} = 0.0$.

17.12.2. Elastic Strain

The equivalent elastic strain is related to the equivalent stress when $\nu' = \nu$ (input as PRXY or NUXY on **MP** command) by:

$$\sigma_{eq} = E\varepsilon_{eq}^{el} \quad (17.153)$$

where:

σ_{eq} = equivalent stress (output using SEQV)

ε_{eq}^{el} = equivalent elastic strain (output using EPEL, EQV)

E = Young's modulus

Note that when ν' = 0 then the equivalent elastic strain is related via

$$\sigma_{eq} = 2G\varepsilon_{eq}^{el} \tag{17.154}$$

where:

G = shear modulus

17.12.3. Plastic Strain

For plasticity, the accumulated effective plastic strain is defined by (see Equation 4.25 (p. 73) and Equation 4.42 (p. 77)):

$$\varepsilon_{eqa}^{pl} = \sum \Delta\varepsilon_{eq}^{pl} \tag{17.155}$$

where:

ε_{eqa}^{pl} = accumulated effective plastic strain (output using NL, EPEQ)

$$\Delta\varepsilon_{eq}^{pl} = \frac{\sqrt{2}}{3}\left[(\Delta\varepsilon_x^{pl} - \Delta\varepsilon_y^{pl})^2 + (\Delta\varepsilon_y^{pl} - \Delta\varepsilon_z^{pl})^2 + (\Delta\varepsilon_z^{pl} - \Delta\varepsilon_x^{pl})^2 + \frac{3}{2}\left(\Delta\gamma_{xy}^{pl^2} + \Delta\gamma_{yz}^{pl^2} + \Delta\gamma_{xz}^{pl^2}\right)\right]^{\frac{1}{2}}$$

This can be related to ε_{eq}^{pl} (output using EPPL, EQV) only under proportional loading situations during the initial loading phase and only when ν' is set to 0.5.

17.12.4. Creep Strain

As with the plastic strains, to compute the equivalent creep strain ε_{eq}^{pl} (EPCR, EQV), use ν' = 0.5.

17.12.5. Total Strain

The equivalent total strains in an analysis with plasticity, creep and thermal strain are:

$$\varepsilon_{eq}^{tot} = \varepsilon_{eq}^{el} + \varepsilon_{eq}^{th} + \varepsilon_{eq}^{pl} + \varepsilon_{eq}^{cr} \tag{17.156}$$

$$\varepsilon_{eq}^{tm} = \varepsilon_{eq}^{el} + \varepsilon_{eq}^{pl} + \varepsilon_{eq}^{cr} \tag{17.157}$$

where:

ε_{eq}^{tot} = equivalent total strain (output using EPTT, EQV)

ε_{eq}^{tm} = equivalent total mechanical strain (output using EPTO, EQV)

ε_{eq}^{th} = equivalent thermal strain

For line elements, use an appropriate value of ν'. If $\varepsilon_{eq}^{pl} >> \varepsilon_{eq}^{el}$, use $\nu' = 0.5$. For other values, use an effective Poisson's ratio between n and 0.5. One method of estimating this is through:

$$\nu' = \frac{1}{2} - \left(\frac{1}{2} - \nu\right)\frac{\varepsilon_{eq}^{el}}{\varepsilon_{eq}^{tot}} \tag{17.158}$$

This computation of equivalent total strain is only valid for proportional loading, and is approximately valid for monotonic loading.

In Mechanical APDL, with the AVPRIN,0,0 command in effect:

17.13. POST26 - Response Power Spectral Density

The cross response PSD between two items is computed using the equation:

$$\begin{aligned} S_{pq}(\omega) = &\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{(\phi_{pj}\phi_{qk} + \phi_{qj}\phi_{pk})}{2}R_{jk}(\omega) \\ &+\sum_{l=1}^{r_2}\sum_{m=1}^{r_2}\frac{(A_{pl}A_{qm} + A_{ql}A_{pm})}{2}\bar{R}_{lm}(\omega) \\ &+\sum_{j=1}^{n}\sum_{l=1}^{r_2}\frac{(\phi_{pj}A_{ql} + \phi_{qj}A_{pl})}{2}\hat{R}_{jl}(\omega) \end{aligned} \tag{17.159}$$

where:

p = reference number of first item (input as IA on **RPSD** command)
q = reference number of second item (input as IB on **RPSD** command)
p and q can be displacements, stresses, or reaction forces.

All other variables in Equation 17.159 (p. 887) are defined in Spectrum Analysis (p. 799). When p = q, the above cross response PSD becomes the auto response PSD.

17.14. POST26 - Computation of Covariance

The covariance between two items p and q is computed using the equation:

$$\begin{aligned} \sigma_{pq}^2 = &\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{(\phi_{pj}\phi_{qk} + \phi_{qj}\phi_{pk})}{2}Q_{jk}(\omega) \\ &+\sum_{l=1}^{r_2}\sum_{m=1}^{r_2}\frac{(A_{pl}A_{qm} + A_{ql}A_{pm})}{2}\bar{Q}_{lm}(\omega) \\ &+\sum_{j=1}^{n}\sum_{l=1}^{r_2}(\phi_{pj}A_{ql} + \phi_{qj}A_{pl})\hat{Q}_{jl}(\omega) \end{aligned} \tag{17.160}$$

where:

p = reference number of first item (input as IA on **CVAR** command)
q= reference number of second item (input as IB on **CVAR** command)
p and q can be displacements, stresses, or reaction forces.

All other variables in Equation 17.160 (p. 887) are defined in Spectrum Analysis (p. 799). When p = q, the above covariance becomes the variance.

17.15. POST1 and POST26 – Complex Results Postprocessing

The modal solution obtained using the complex eigensolvers (UNSYM, DAMP, QRDAMP) and the solution from a harmonic analysis are complex. It can be written as

$$R = R_R + iR_I \tag{17.161}$$

where:

R = the complex degree of freedom solution (a nodal displacement U_x, a reaction force F_y, etc.).
R_R = the real part of the solution R.
R_I = the imaginary part of the solution R.

The same complex solution may also be expressed as:

$$R = R_{max}.e^{i\phi} \tag{17.162}$$

where:

R_{max} = the degree of freedom amplitude.
ϕ = the degree of freedom phase shift.

The phase shift of the solution is different at each degree of freedom so that the total amplitude at a node is not the square root of the sum of squares of the degrees of freedom amplitudes (R_{max}). More generally, total amplitudes (SUM), phases and other derived results (principal strains/stresses, equivalent strain/stress,… for example) at one node do not vary harmonically as degree of freedom solutions do.

The relationship between R_R, R_I, R_{max} and ϕ is defined as follows:

$$R_{max} = \sqrt{R_R^{\,2} + R_I^{\,2}} \tag{17.163}$$

$$\phi = \tan^{-1}\frac{R_I}{R_R} \tag{17.164}$$

$R_R = R_{max}\cos\phi$

$R_I = R_{max}\sin\phi$

In POST1, use KIMG in the **SET** command to specify which results are to be stored: the real parts, the imaginary parts, the amplitudes or the phases.

In POST26, use **PRCPLX** and **PLCPLX** to define the output form of the complex variables.

The complete complex solution is harmonic. It is defined as:

$$R(t) = Re^{i\Omega t} \tag{17.165}$$

where:

Ω = the excitation frequency in a harmonic analysis, or the natural damped frequency in a complex modal analysis.

In the equations of motion for harmonic and complex modal analyses, the complex notations are used for ease of use but the time dependant solution at one degree of freedom is real.

The solution of the harmonic analysis is defined as:

$$R(t)_{real} = \mathrm{R}_R \cos \Omega t - R_I \sin \Omega t \tag{17.166}$$

The **ANHARM** command issued after the harmonic analysis is based on Equation 17.166 (p. 889). The **HRCPLX**and **LCOPER** (with *Oper2* = *CPXMAX*) commands are based on Equation 17.166 (p. 889) for both modal and harmonic analyses.

The complete solution of the complex modal analysis is defined as:

$$R\left(t\right)_{real} = e^{\sigma t}\left(R_R \cos \Omega t - R_I \sin \Omega t\right) \tag{17.167}$$

Where σ is the real part of the complex frequency for the damped eigensolvers (DAMP and QRDAMP) and its imaginary part for the unsymmetric eigensolver (UNSYM). See Complex Eigensolutions (p. 736) for more details about complex eigensolutions.

The **ANHARM** command issued after a complex modal analysis is based on equation (Equation 17.167 (p. 889)) except if NPERIOD is set to -1. In this case, the exponential term (decay or growth of the oscillation) is ignored and the equation reduces to (Equation 17.166 (p. 889)).

In the **LCOPER** command with Oper2=*CPXMAX*, a loop on the phase (Ωt), is performed to calculate the maximum (that may occur at any phase at each location) of the degree of freedom results, displacements and stresses, the principal stresses (S1, S2, and S3), the stress intensity (*SINT*), the equivalent stress (*SEQV*), and the equivalent strain.

17.16. POST1 - Modal Assurance Criterion (MAC)

The modal assurance criterion (MAC) can compare two real solutions or two complex solutions.

The MAC between two real solutions is computed using the equation:

$$mac(\phi_i^{(1)}, \phi_j^{(2)}) = \frac{(\phi_i^{(1)^t} \,"\, m^{(k)} \,"\, \phi_j^{(2)})^2}{(\phi_i^{(1)^t} \,"\, m^{(k)} \,"\, \phi_i^{(1)})(\phi_j^{(2)^t} \,"\, m^{(k)} \,"\, \phi_j^{(2)})} \tag{17.168}$$

where:

$\phi_i^{(1)}$ = the *ith* displacement vector of solution 1. (solution 1 is read on *File1* and index *i* corresponds to *Sbstep1* on the **RSTMAC** command).

$\phi_j^{(2)}$ = the *jth* displacement vector of solution 2. (solution 2 is read on *File2* and index *j* corresponds to *Sbstep2* on the **RSTMAC** command).

$m^{(k)}$ = diagonal of the mass matrix used in obtaining solution k. It is included in the calculation only if *KeyMass* is ON (default). k = 1 if nodes are matched (*TolerN* > 0 on the **RSTMAC** command). k = 2 if nodes are mapped and solution 1 is interpolated (*TolerN* = -1 on the **RSTMAC** command).

The MAC between two complex solutions is computed using the equation:

$$mac(\phi_i^{(1)}, \phi_j^{(2)}) = \frac{(\bar{\phi}_i^{(1)^t} \,"\, m^{(k)} \,"\, \phi_j^{(2)})(\phi_i^{(1)^t} \,"\, m^{(k)} \,"\, \bar{\phi}_j^{(2)})}{(\bar{\phi}_i^{(1)^t} \,"\, m^{(k)} \,"\, \phi_i^{(1)})(\bar{\phi}_j^{(2)^t} \,"\, m^{(k)} \,"\, \phi_j^{(2)})} \tag{17.169}$$

where:

$\bar{\phi}$ = the complex conjugate of a complex vector ϕ.

If the diagonal of the mass matrix is not available (for example on a Universal Format file), the modal assurance criterion is not weighted with the mass, i.e. the mass is assumed to be equal at all degrees of freedom.

The dot product of the displacement vectors is calculated at matched nodes if *TolerN*>0, and at mapped nodes if *TolerN*=-1.

Chapter 18: Probabilistic Design

In general, a finite element analysis program starts with a set of input data such as geometric parameters, material parameter, loads and boundary conditions. The program then generates some output data for the analyzed component such as temperatures, displacements, stresses, strains, voltages and/or velocities. Almost all input parameters are subjected to scatter due to either natural variability or inaccuracies during manufacturing or operation. In a probabilistic approach, the uncertainties on the input side are described by statistical distribution functions, allowing you to obtain answers to common questions about your analysis.

The following probabilistic design topics are available:

18.1. Uses for Probabilistic Design

A probabilistic analysis can be used to answer the following most common questions.

1. If some of the input parameters are subjected to scatter and are therefore identified as random input variable, how large is the resulting scatter or uncertainty induced on the side of the output parameters?

2. If the output parameters are uncertain or random as well, what is the probability that a certain design criterion formulated in terms of these output parameters is no longer fulfilled?

3. Which random input variables are contributing the most to the scatter of the random output parameters and the probability that a certain design criteria is no longer fulfilled?

Probabilistic Modeling and Preprocessing explains the mathematical background for describing random input variables in terms of statistical distribution functions.

Probabilistic Methods provides the theoretical background of the methods that are used to provide the probabilistic results that enable the user to answer the questions above. In this section the Monte Carlo Simulation Method and the Response Surface Method are explained in detail.

Regression Analysis for Building Response Surface Models is dedicated to a technique called regression analysis, which is an option for some probabilistic methods and a necessity for others in order to generate probabilistic results.

Probabilistic Postprocessing is focused on the mathematical background of the statistical procedures that are used to postprocess and interpret the probabilistic results. The interpretation of the probabilistic results then provides the answers to the questions listed above.

A simpler and manually driven form of performing Monte Carlo simulations is explained in Statistical Procedures (p. 834) of this manual.

Glossary of Symbols

$\langle \ldots)$ = one-sided lower confidence limit

$(\ldots \rangle$ = one-sided upper confidence limit

$\langle \ldots \rangle$ = two-sided confidence interval

{...} = vector in column format

$\lfloor \ldots \rfloor$ = vector in row format

[...] = matrix

Notations

A symbol given as an upper case character always refers to a random variable, whereas a symbol specified with the corresponding lower case character indicates a particular, but arbitrary value of that random variable. Example: X is a random variable and x is a particular, but arbitrary value of X. This rule does not apply to functions of variables, such as distribution functions or other mathematical functions.

A function of one or more independent variables can have one or more parameters, which further specify the shape of the function. Here, we follow the notation that such a function is denoted with f (x_1, x_2, x_3, ... | a, b, c ...), where x_1, x_2, x_3, etc. are the independent variables of the function and a, b, c, etc. are the parameters that influence it.

For the exponential function the notation (...) is used.

18.2. Probabilistic Modeling and Preprocessing

In the following, we will use the expression random input variable for the inaccuracies and uncertainties influencing the outcome of an analysis. In probabilistic design, statistical distribution functions are used to describe and quantify random input variables. In the following section, various statistical distribution types are explained in detail. The following information is typically used characterize a statistical distribution:

$f_X(x)$ = Probability density function. The probability density function of a random input variable X is a measure for the relative frequency at which values of random input variables are expected to occur.

$F_X(x)$ = Cumulative distribution function. The cumulative distribution function of a random input variable X is the probability that values for the random input variable remain below a certain limit x.

$F_X^{-1}(x)$ = Inverse cumulative distribution function

μ = Mean value. The mean value of a random input variable X is identical to the arithmetic average. It is a measure for the location of the distribution of a random input variable.

σ = Standard deviation. The standard deviation is a measure for the width of the distribution of a random input variable.

18.2.1. Statistical Distributions for Random Input Variables

18.2.1.1. Gaussian (Normal) Distribution

A Gaussian or normal distribution of a random variable X has two distribution parameters, namely a mean value μ and a standard deviation σ. The probability density function of a Gaussian distribution is:

$$f_X(x \mid \mu,\sigma) = \frac{1}{\sigma}\ \varphi\left(\frac{x-\mu}{\sigma}\right) \tag{18.1}$$

where:

$f_X(x \mid \mu,\sigma)$ = probability density function of the Gaussian distribution. According to the notation mentioned in Notations (p. 892), x is the independent variable and μ and σ are the parameters of the probability density function.
$\phi(...)$ = probability density function of the standard normal distribution. The standard normal distribution is a normal distribution with a mean value of 0.0 and a standard deviation of 1.0.

$$\varphi(z) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}z^2\right) \tag{18.2}$$

The cumulative distribution function of the Gaussian distribution is:

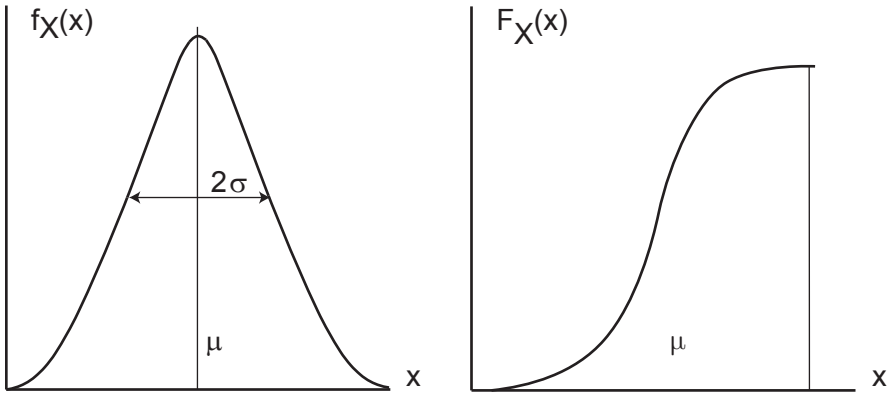
$$F_X(x \mid \mu,\sigma) = \Phi\left(\frac{x-\mu}{\sigma}\right) \tag{18.3}$$

where:

$\Phi(...)$ = cumulative distribution function of the standard normal distribution

There is no closed-form solution available for Equation 18.3 (p. 893). See Abramowitz and Stegun([303] (p. 938)) for more details. The probability density function and the cumulative distribution function of a Gaussian distribution are shown in Figure 18.1: Gaussian Distribution Functions (p. 893).

Figure 18.1: Gaussian Distribution Functions



Probability Density Function (left) and Cumulative Distribution Function (right)

The inverse cumulative distribution function of the Gaussian distribution is:

$$x = F_X^{-1}(p \mid \mu, \sigma) \tag{18.4}$$

where:

p = a given probability

The random variable value x, for which Equation 18.4 (p. 894) is satisfied, can only be found iteratively using the solution of Equation 18.3 (p. 893).

Obviously, the mean value and the standard deviation of a random variable X with a Gaussian distribution are the same as the two distribution parameters μ and σ respectively.

18.2.1.2. Truncated Gaussian Distribution

A truncated Gaussian distribution of a random variable X has four distribution parameters, namely a mean value μ_G and a standard deviation σ_G of the non-truncated Gaussian distribution, and the lower limit x_{min} and the upper limit x_{max}.

The probability density function of a truncated Gaussian distribution is:

For $x < x_{min}$ or $x > x_{max}$:

$$f_X(x \mid \mu_G, \sigma_G, x_{min}, x_{max}) = 0 \tag{18.5}$$

For $x_{min} \leq x \leq x_{max}$:

$$f_X(x \mid \mu_G, \sigma_G, x_{min}, x_{max}) = \frac{1}{\left(\Phi\left(\frac{x_{max} - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{x_{min} - \mu_G}{\sigma_G}\right)\right)\sigma_G} \varphi\left(\frac{x - \mu_G}{\sigma_G}\right) \tag{18.6}$$

where:

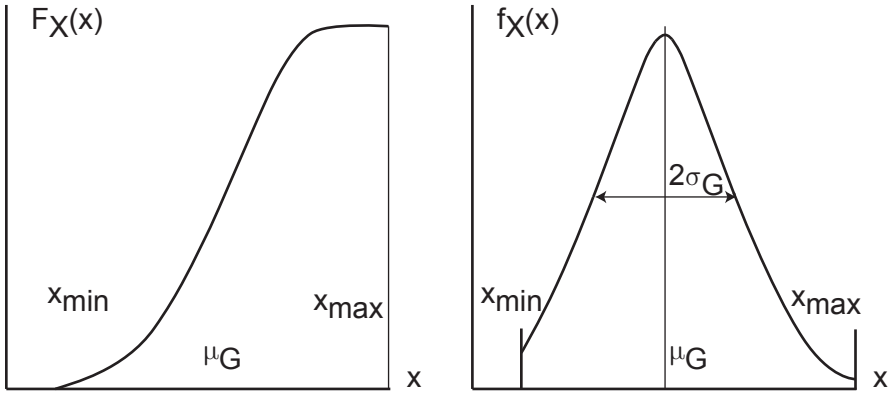
$\Phi(...)$ = cumulative distribution function of the standard normal distribution
$\phi(...)$ = probability density function of the standard normal distribution (see Equation 18.2 (p. 893))

The cumulative distribution function of the truncated Gaussian distribution is:

$$F_X(x \mid \mu_G, \sigma_G, x_{min}, x_{max}) = \frac{\Phi\left(\frac{x - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{x_{min} - \mu_G}{\sigma_G}\right)}{\Phi\left(\frac{x_{max} - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{x_{min} - \mu_G}{\sigma_G}\right)} \tag{18.7}$$

There is no closed-form solution available for Equation 18.7 (p. 894). See Abramowitz and Stegun([303] (p. 938)) for more details. The probability density function and the cumulative distribution function of a truncated Gaussian distribution are shown in Figure 18.2: Truncated Gaussian Distribution (p. 895).

Figure 18.2: Truncated Gaussian Distribution



Probability Density Function (left) and Cumulative Distribution Function (right)

Same as for Equation 18.4 (p. 894) also the inverse cumulative distribution function of the truncated Gaussian distribution must be found iteratively using the solution of Equation 18.7 (p. 894).

The mean value of a random variable X with a truncated Gaussian distribution is:

$$\mu = \mu_G - \sigma_G^2 \frac{\varphi\left(\frac{x_{max} - \mu_G}{\sigma_G}\right) - \varphi\left(\frac{x_{min} - \mu_G}{\sigma_G}\right)}{\Phi\left(\frac{x_{max} - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{x_{min} - \mu_G}{\sigma_G}\right)} \tag{18.8}$$

and the standard deviation is:

$$\sigma = \sigma_G \sqrt{(\mu_G - \sigma_G^2 (f_X(x_{max}) - f_X(x_{min})))(f_X(x_{max}) - f_X(x_{min})) + 1 - x_{max} f_X(x_{max}) + x_{min} f_X(x_{min})} \tag{18.9}$$

where:

$f_X(x_{min}) = f_x(x_{min} \mid \mu_G, \sigma_G, x_{min}, x_{max})$ is the value of the probability density function of the truncated Gaussian distribution according to Equation 18.6 (p. 894) at $x = x_{min}$. This expression has been abbreviated to shorten the equation above.
$f_X(x_{max})$ = defined analogously.

18.2.1.3. Lognormal Distribution

A random variable X is said to follow a lognormal distribution if ln(X) follows a Gaussian (or normal) distribution. A lognormal distribution of a random input variable X has two distribution parameters, namely a logarithmic mean value ξ and the logarithmic deviation δ. The distribution parameter ξ is the mean value of ln(X) and the logarithmic deviation δ is the standard deviation of ln(X).

The probability density function of a truncated Gaussian distribution is:

$$f_X(x \mid \xi, \delta) = \frac{1}{x\delta} \varphi\left(\frac{\ln x - \xi}{\delta}\right) \tag{18.10}$$

where:

$\phi(...)$ = probability density function of the standard normal distribution (see Equation 18.2 (p. 893))

Usually, a lognormal distribution is specified as one of two cases:

Case 1: Using the mean value m and the standard deviation σ of the random input variable X. In this case, the parameters ξ and δ can be derived from the mean value μ and the standard deviation σ using:

$$\xi = \ln\mu - 0.5\delta \tag{18.11}$$

$$\delta = \sqrt{\ln\left[\left(\frac{\sigma}{\mu}\right)^2 + 1\right]} \tag{18.12}$$

Case 2: Using the logarithmic mean ξ and the logarithmic deviation δ as mentioned above.

The cumulative distribution function of the lognormal distribution is:

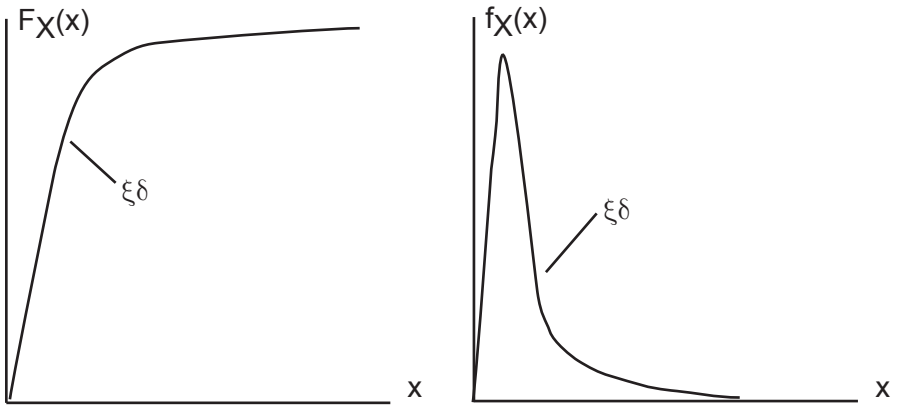
$$F_X(x \mid \mu, \sigma) = \Phi\left(\frac{\ln x - \xi}{\delta}\right) \tag{18.13}$$

where:

$\Phi(...)$ = cumulative distribution function of the standard normal distribution

There is no closed-form solution available for Equation 18.13 (p. 896). See Abramowitz and Stegun([303] (p. 938)) for more details. The probability density function and the cumulative distribution function of a lognormal distribution are shown in Figure 18.3: Lognormal Distribution (p. 896).

Figure 18.3: Lognormal Distribution



Probability Density Function (left) and Cumulative Distribution Function (right)

As with Equation 18.4 (p. 894), the inverse cumulative distribution function of the lognormal distribution must be found iteratively using the solution of Equation 18.13 (p. 896).

For case 1, the specified parameters μ and σ directly represent the mean value and the standard deviation of a random variable X respectively.

For case 2, the mean value of the random variable X is:

$$\mu = \exp(\xi + 0.5\delta^2) \tag{18.14}$$

and the standard deviation is:

$$\sigma = \sqrt{\exp(2\xi + \delta^2)(\exp(\delta^2) - 1)} \tag{18.15}$$

18.2.1.4. Triangular Distribution

A triangular distribution of a random variable X is characterized by three distribution parameters, namely the lower limit x_{min}, the maximum likely value x_{mlv} and the upper limit x_{max}.

The probability density function of a triangular distribution is:

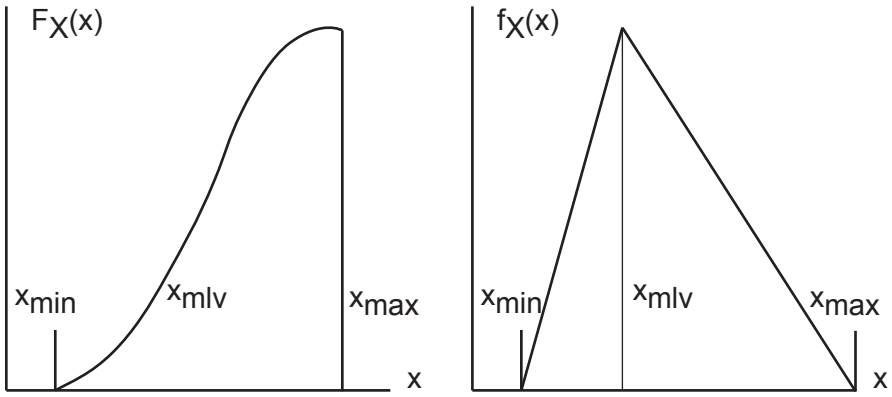
$$\begin{aligned} f_X(x \mid x_{min}, x_{mlv}, x_{max}) &= \frac{2(x - x_{min})}{(x_{mlv} - x_{min})(x_{max} - x_{min})} \quad \text{for } x \le x_{mlv} \\ f_X(x \mid x_{min}, x_{mlv}, x_{max}) &= \frac{2(x_{max} - x)}{(x_{max} - x_{mlv})(x_{max} - x_{min})} \quad \text{for } x > x_{mlv} \end{aligned} \tag{18.16}$$

The cumulative distribution function of a triangular distribution is:

$$\begin{aligned} F_X(x \mid x_{min}, x_{mlv}, x_{max}) &= \frac{(x - x_{min})^2}{(x_{mlv} - x_{min})(x_{max} - x_{min})} \quad \text{for } x \le x_{mlv} \\ F_X(x \mid x_{min}, x_{mlv}, x_{max}) &= 1 - \frac{(x_{max} - x)^2}{(x_{max} - x_{mlv})(x_{max} - x_{min})} \quad \text{for } x > x_{mlv} \end{aligned} \tag{18.17}$$

The probability density function and the cumulative distribution function of a triangular distribution are shown in Figure 18.4: Triangular Distribution (p. 897).

Figure 18.4: Triangular Distribution



Probability Density Function (left) and Cumulative Distribution Function (right)

The inverse cumulative distribution function of a triangular distribution is:

$$\begin{aligned} x &= x_{min} + \sqrt{p(x_{mlv} - x_{min})(x_{max} - x_{min})} \quad \text{for } p \le \frac{(x_{mlv} - x_{min})}{(x_{max} - x_{min})} \\ x &= x_{max} - \sqrt{(1-p)(x_{max} - x_{mlv})(x_{max} - x_{min})} \quad \text{for } p > \frac{(x_{mlv} - x_{min})}{(x_{max} - x_{min})} \end{aligned} \tag{18.18}$$

where:

p = a given probability

The mean value of a random variable X with a triangular distribution is:

$$\mu = \frac{(x_{min} + x_{mlv} + x_{max})}{3} \tag{18.19}$$

and the standard deviation is:

$$\sigma = \sqrt{\frac{x_{min}^2 + x_{mlv}^2 + x_{max}^2 - x_{min}x_{mlv} - x_{mlv}x_{max} - x_{min}x_{max}}{18}} \tag{18.20}$$

18.2.1.5. Uniform Distribution

A uniform distribution of a random variable X is characterized by two distribution parameters, namely the lower limit x_{min} and the upper limit x_{max}.

The probability density function of a uniform distribution is:

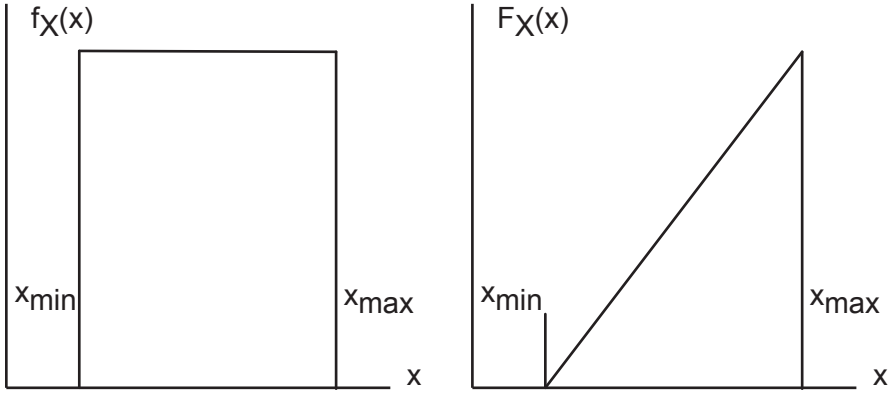
$$f_X(x \mid x_{min}, x_{max}) = \frac{1}{x_{max} - x_{min}} \tag{18.21}$$

The cumulative distribution function of a uniform distribution is:

$$F_X(x \mid x_{min}, x_{max}) = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{18.22}$$

The probability density function and the cumulative distribution function of a uniform distribution are shown in Figure 18.5: Uniform Distribution (p. 898).

Figure 18.5: Uniform Distribution



Probability Density Function (left) and Cumulative Distribution Function (right)

The inverse cumulative distribution function of a uniform distribution is given by:

$$x = x_{min} + p(x_{max} - x_{min}) \tag{18.23}$$

where:

p = a given probability

The mean value of a random variable X with a uniform distribution is:

$$\mu = 0.5(x_{min} + x_{max}) \tag{18.24}$$

and the standard deviation is:

$$\sigma = \frac{x_{min} - x_{max}}{\sqrt{12}} \tag{18.25}$$

18.2.1.6. Exponential Distribution

An exponential distribution of a random variable X has two distribution parameters, namely the decay parameter λ and the shift parameter (or lower limit) x_{min}.

The probability density function of an exponential distribution is:

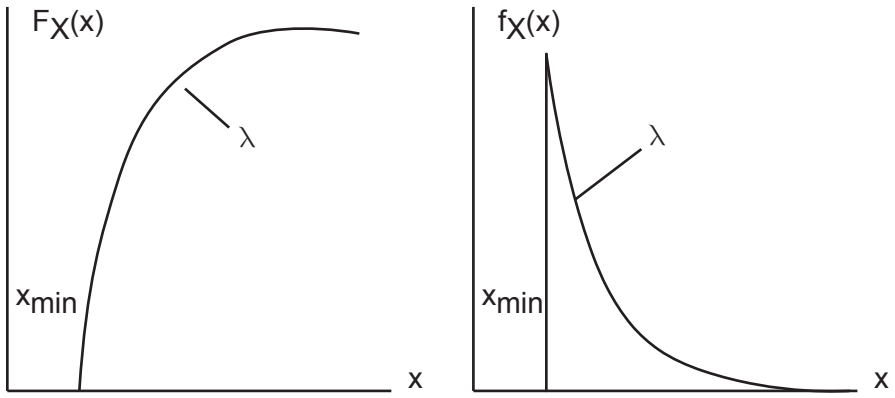
$$f_X(x \mid \lambda, x_{min}) = \lambda \exp(-\lambda(x - x_{min})) \tag{18.26}$$

The cumulative distribution function of the exponential distribution is:

$$F_X(x \mid \lambda, x_{min}) = 1 - \exp(-\lambda(x - x_{min})) \tag{18.27}$$

The probability density function and the cumulative distribution function of an exponential distribution are shown in Figure 18.6: Exponential Distribution (p. 899).

Figure 18.6: Exponential Distribution



Probability Density Function (left) and Cumulative Distribution Function (right)

The inverse cumulative distribution function of the exponential distribution is:

$$x = x_{min} - \frac{\ln(1-p)}{\lambda} \tag{18.28}$$

where:

p = a given probability

The mean value of a random variable X with an exponential distribution is:

$$\mu = x_{min} + \frac{1}{\lambda} \tag{18.29}$$

and the standard deviation is:

$$\sigma = \frac{1}{\lambda} \tag{18.30}$$

18.2.1.7. Beta Distribution

A Beta distribution of a random variable X has four distribution parameters, namely the shape parameters r and t, the lower limit x_{min} and the upper limit x_{max}. The probability density function of a Beta distribution is:

$$f_X(x \mid r,t,x_{min},x_{max}) = \frac{\left(\frac{x - x_{min}}{x_{max} - x_{min}}\right)^{r-1}\left(1 - \frac{x - x_{min}}{x_{max} - x_{min}}\right)^{t-1}}{B(r,t)(x_{max} - x_{min})} \tag{18.31}$$

where:

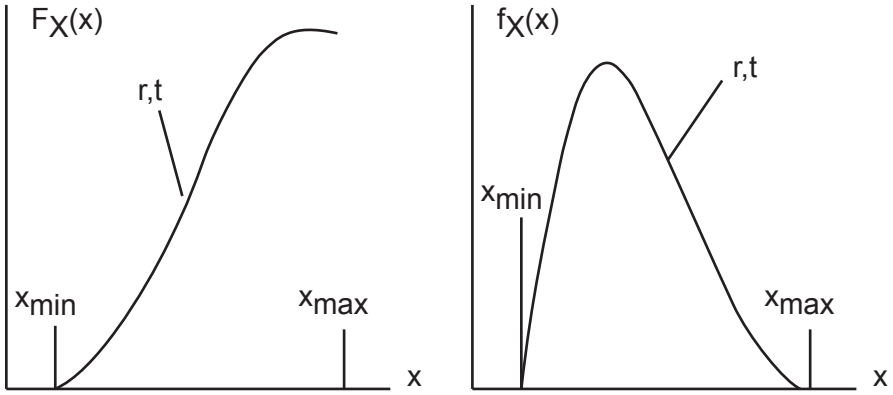
B(...) = complete Beta function

$$F_X(x \mid r,t,x_{min},x_{max}) = \int_{x_{min}}^{x} f_X(\xi \mid r,t,x_{min},x_{max})d\xi \tag{18.32}$$

There is no closed-form solution available for Equation 18.32 (p. 900). See Abramowitz and Stegun([303] (p. 938)) for more details.

The probability density function and the cumulative distribution function of a Beta distribution are shown in Figure 18.7: Beta Distribution (p. 900).

Figure 18.7: Beta Distribution



Probability Density Function (left) and Cumulative Distribution Function (right)

As with Equation 18.4 (p. 894) also the inverse cumulative distribution function of the Beta distribution must be found iteratively using the solution of Equation 18.32 (p. 900).

The mean value of a random variable X with a Beta distribution is:

$$\mu = x_{min} + (x_{max} - x_{min})\frac{r}{r+t} \tag{18.33}$$

and the standard deviation is:

$$\sigma = \frac{x_{max} - x_{min}}{r+t}\sqrt{\frac{r\,t}{r+t+1}} \tag{18.34}$$

18.2.1.8. Gamma Distribution

A Gamma distribution of a random variable X has two distribution parameters, namely an exponential parameter k and the decay parameter λ.

The probability density function of a Gamma distribution is:

$$f_X(x \mid k,\lambda) = \frac{\lambda^k x^{k-1}}{\Gamma(k)} \exp(-\lambda x) \tag{18.35}$$

where:

$\Gamma(...)$ = Gamma function

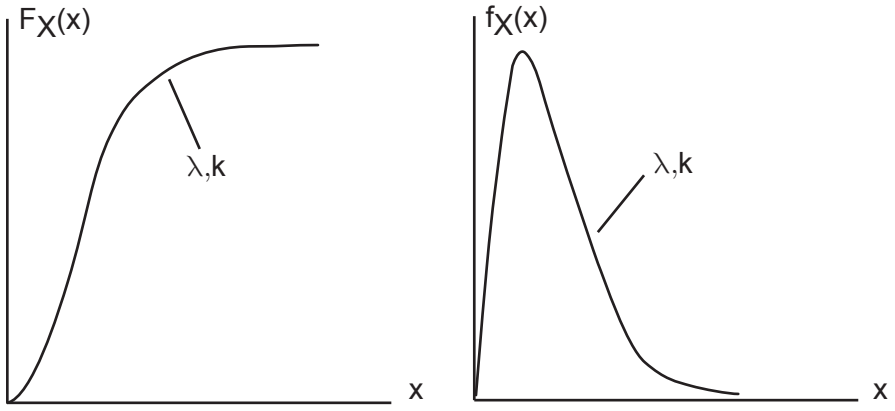
The cumulative distribution function of the Gamma distribution is:

$$F_X(x \mid k,\lambda) = \int_0^x f_X(\xi \mid k,\lambda) d\xi \tag{18.36}$$

There is no closed-form solution available for Equation 18.36 (p. 901). See Abramowitz and Stegun([303] (p. 938)) for more details.

The probability density function and the cumulative distribution function of a Gamma distribution are shown in Figure 18.8: Gamma Distribution (p. 901).

Figure 18.8: Gamma Distribution



Probability Density Function (left) and Cumulative Distribution Function (right)

As with Equation 18.4 (p. 894) also the inverse cumulative distribution function of the Gamma distribution must be found iteratively using the solution of Equation 18.36 (p. 901).

The mean value of a random variable X with a Gamma distribution is:

$$\mu = \frac{k}{\lambda} \tag{18.37}$$

and the standard deviation is:

$$\sigma = \frac{\sqrt{k}}{\lambda} \tag{18.38}$$

18.2.1.9. Weibull Distribution

A Weibull distribution is also called a "Type III smallest" distribution. A Weibull distribution of a random variable X is characterized by three distribution parameters, namely the Weibull exponent m, the Weibull characteristic value x_{chr} and the lower limit x_{min}. A two parameter Weibull distribution may be used, in which case $x_{min} = 0.0$.

The probability density function of a Weibull distribution is:

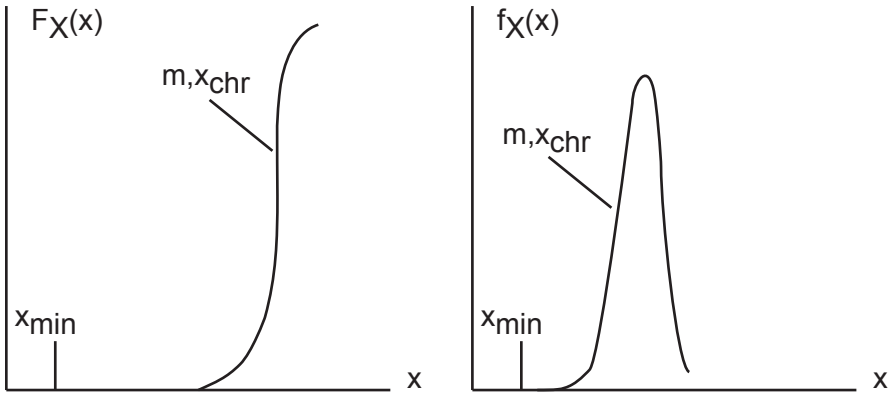
$$f_X(x \mid x_{chr}, m, x_{min}) = \frac{m(x - x_{min})^{m-1}}{(x_{chr} - x_{min})^m} \exp\left(-\left(\frac{x - x_{min}}{x_{chr} - x_{min}}\right)^m\right) \tag{18.39}$$

The cumulative distribution function of a Weibull distribution is:

$$F_X(x \mid x_{chr}, m, x_{min}) = 1 - \exp\left(-\left(\frac{x - x_{min}}{x_{chr} - x_{min}}\right)^m\right) \tag{18.40}$$

The probability density function and the cumulative distribution function of a Weibull distribution are shown in Figure 18.9: Weibull Distribution (p. 902).

Figure 18.9: Weibull Distribution



Probability Density Function (left) and Cumulative Distribution Function (right)

The inverse cumulative distribution function of a Weibull distribution is:

$$x = x_{min} + (\ln(1-p))^{\frac{1}{m}} \tag{18.41}$$

where:

p = a given probability

The mean value of a random variable X with a Weibull distribution is:

$$\mu = x_{min} + \Gamma\left(1 + \frac{1}{m}\right) \tag{18.42}$$

and the standard deviation is:

$$\sigma = (x_{chr} - x_{min})\sqrt{\Gamma\left(1+\frac{2}{m}\right) - \Gamma^2\left(1+\frac{1}{m}\right)} \quad (18.43)$$

18.3. Probabilistic Methods

18.3.1. Introduction

All probabilistic methods execute the deterministic problem several times, each time with a different set of values for the random input variables. The various probabilistic methods differ in the way in which they vary the values of the random input variables from one execution run to the next.

One execution run with a given set of values for the random input variables $\{x\} = \lfloor x_1\ x_2 \ldots x_m \rfloor^T$ with m is the number of random input variables is called a sampling point, because the set of values for the random input variables marks a certain point in the space of the random input variables.

18.3.2. Common Features for all Probabilistic Methods

18.3.2.1. Random Numbers with Standard Uniform Distribution

A fundamental feature of probabilistic methods is the generation of random numbers with standard uniform distribution. The standard uniform distribution is a uniform distribution with a lower limit x_{min} = 0.0 and an upper limit x_{max} = 1.0. Methods for generating standard uniformly distributed random numbers are generally based on recursive calculations of the residues of modulus m from a linear transformation. Such a recursive relation is given by the equation:

$$s_i = a\ s_{i-1} + c - k_{i-1}\ m \quad (18.44)$$

where:

a, c, m = nonnegative integers
s_{i-1} = previous seed value of the recursion
k_{i-1} = integer part of the ratio (a s_{i-1} + c) / m

A set of random numbers with standard uniform distribution is obtained by normalizing the value calculated by Equation 18.44 (p. 903) with the modulus m:

$$p_i = \frac{s_i}{m} \quad (18.45)$$

It is obvious from Equation 18.44 (p. 903) that an identical set of random numbers will be obtained if the same start value for the seed s_{i-1} is used. Therefore, the random numbers generated like that are also called "pseudo random" numbers. See Hammersley and Handscomb([308] (p. 938)) for more details about the generation of random numbers with standard uniform distribution.

18.3.2.2. Non-correlated Random Numbers with an Arbitrary Distribution

For probabilistic analyses, random numbers with arbitrary distributions such as the ones described in Statistical Distributions for Random Input Variables (p. 893) are needed. The most effective method to generate random number with any arbitrary distribution is the inverse transformation method. A set of random numbers for the random variable X having a cumulative distribution function F_x (x) can be

generated by using a set of standard uniformly distributed random numbers according to Equation 18.45 (p. 903) and transforming them with the equation:

$$x_i = F_X^{-1}(p_i) \tag{18.46}$$

Depending on the distribution type of the random variable X, the inverse cumulative distribution function can be calculated as described in Statistical Distributions for Random Input Variables (p. 893).

18.3.2.3. Correlated Random Numbers with an Arbitrary Distribution

Correlated random input variables must be dealt with by all probabilistic methods, if there are random input variables, the user has identified as being correlated with each other. In order to handle correlated random input variables it is necessary to transform the random variable values using the Nataf model. The Nataf model is explained in detail in Liu and Der Kiureghian([311] (p. 938))).

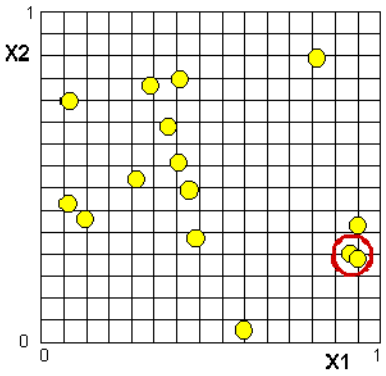
18.3.3. Monte Carlo Simulation Method

A fundamental characteristic of the Monte Carlo Simulation method is the fact that the sampling points are located at random locations in the space of the random input variables. There are various techniques available in literature that can be used to evaluate the random locations of the sampling points (see Hammersley and Handscomb([308] (p. 938)), Iman and Conover([309] (p. 938))).

18.3.3.1. Direct Monte Carlo Simulation

The direct Monte Carlo Simulation method is also called the crude Monte Carlo Simulation method. It is based on randomly sampling the values of the random input variables for each execution run. For the direct Monte Carlo Simulation method the random sampling has no memory, i.e., it may happen that one sampling point is relative closely located to one or more other ones. An illustration of a sample set with a sample size of 15 generated with direct Monte Carlo Simulation method for two random variables X_1 and X_2 both with a standard uniform distribution is shown in Figure 18.10: Sample Set Generated with Direct Monte Carlo Simulation Method (p. 904).

Figure 18.10: Sample Set Generated with Direct Monte Carlo Simulation Method



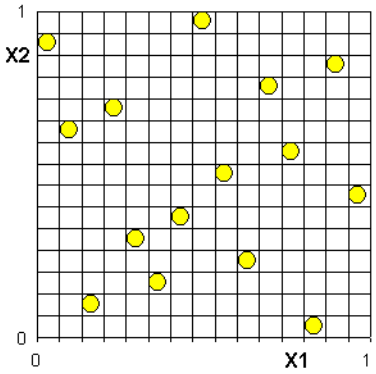
As indicated with the circle, there may be sample points that are located relatively close to each other.

18.3.3.2. Latin Hypercube Sampling

For the Latin Hypercube Sampling technique the range of all random input variables is divided into n intervals with equal probability, where n is the number of sampling points. For each random variable each interval is "hit" only once with a sampling point. The process of generating sampling points with Latin Hypercube has a "memory" in the meaning that the sampling points cannot cluster together, be-

cause they are restricted within the respective interval. An illustration of a sample with a sample size of 15 generated with Latin Hypercube Sampling method for two random variables X_1 and X_2 both with a standard uniform distribution is shown in Figure 18.11: Sample Set Generated with Latin Hypercube Sampling Method (p. 905).

Figure 18.11: Sample Set Generated with Latin Hypercube Sampling Method



There are several ways to determine the location of a sampling point within a particular interval.

1. *Random location*: Within the interval the sampling point is positioned at a random location that agrees with the distribution function of the random variable within the interval.

2. *Median location*: Within the interval the sampling point is positioned at the 50% position as determined by the distribution function of the random variable within the interval.

3. *Mean value*: Within the interval the sampling point is positioned at the mean value position as determined by the distribution function of the random variable within the interval.

See Iman and Conover([309] (p. 938)) for further details.

18.3.4. The Response Surface Method

For response surface methods the sampling points are located at very specific, predetermined positions. For each random input variable the sampling points are located at given levels only.

Response surface methods consist of two key elements:

1. *Design of Experiments*: Design of Experiments is a technique to determine the location of the sampling points. There are several versions for design of experiments available in literature (see Montgomery([312] (p. 938)), Myers([313] (p. 938))). These techniques have in common that they are trying to locate the sampling points such that the space of random input variables is explored in a most efficient way, meaning obtaining the required information with a minimum number of sampling points. An efficient location of the sampling points will not only reduce the required number of sampling points, but also increase the accuracy of the response surface that is derived from the results of those sampling points. Two specific forms of design of experiments are outlined in the remainder of this section.

2. *Regression Analysis*: Regression analysis is a technique to determine the response surface based on the results obtained at the sampling points (see Neter et al.([314] (p. 938))). Regression Analysis for Building Response Surface Models (p. 909) has been dedicated to discuss regression analysis, because regression analysis is not only used in the context of response surface methods.

18.3.4.1. Central Composite Design

Location of Sampling Points Expressed in Probabilities

For central composite design the sampling points are located at five different levels for each random input variable. In order to make the specification of these levels independent from the distribution type of the individual random input variables, it is useful to define these levels in terms of probabilities. The five different levels of a central composite design shall be denoted with p_i, with i = 1, ... , 5.

A central composite design is composed of three different parts, namely:

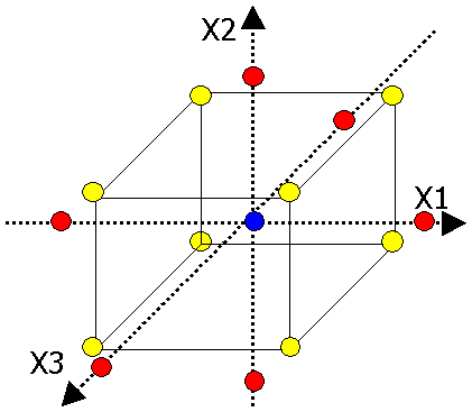
1. *Center point*: At the center point the values of all random input variables have a cumulative distribution function that equals p_3.

2. *Axis points*: There are two points for each random variable located at the axis position, i.e., if there are m random input variables then there are 2m axis points. For the axis points all random input variables except one have a value corresponding to the center location and one random variable has a value corresponding to p_1 for the low level point and corresponding to p_5 for the high level point.

3. *Factorial points*: In a central composite design there are 2^{m-f} factorial points. Here, f is the fraction of the factorial part. The fraction of the factorial part is explained in more detail in the next subsection. For the factorial points all random input variables have values corresponding to permutations of p_2 for the lower factorial level and p_4 for the upper factorial level.

A sample set based on a central composite design for three random variables X_1, X_2 and X_3 is shown in Figure 18.12: Sample Set Based on a Central Composite Design (p. 906).

Figure 18.12: Sample Set Based on a Central Composite Design



For this example with three random input variables the matrix describing the location of the sampling points in terms of probabilities is shown in Table 18.1: Probability Matrix for Samples of Central Composite Design (p. 906).

Table 18.1: Probability Matrix for Samples of Central Composite Design

Sample	X_1	X_2	X_3	Part
1	p_3	p_3	p_3	Center
2	p_1	p_3	p_3	Axis Points
3	p_5	p_3	p_3	

Sample	X_1	X_2	X_3	Part
4	p_3	p_1	p_3	
5	p_3	p_5	p_3	
6	p_3	p_3	p_1	
7	p_3	p_3	p_5	
8	p_2	p_2	p_2	Factorial Points
9	p_2	p_2	p_4	
10	p_2	p_4	p_2	
11	p_2	p_4	p_4	
12	p_4	p_2	p_2	
13	p_4	p_2	p_4	
14	p_4	p_4	p_2	
15	p_4	p_4	p_4	

Resolution of the Fractional Factorial Part

For problems with a large number of random input variables m, the number of sampling points is getting extensively large, if a full factorial design matrix would be used. This is due to the fact that the number of sampling points of the factorial part goes up according to 2^m in this case. Therefore, with increasing number of random variables it is common practice to use a fractional factorial design instead of a full factorial design. For a fractional factorial design, the number of the sampling points of the factorial part grows only with 2^{m-f}. Here f is the fraction of the factorial design so that f = 1 represents a half-factorial design, f = 2 represents a quarter-factorial design, etc. Consequently, choosing a larger fraction f will lead to a lower number of sampling points.

In a fractional factorial design the m random input variables are separated into two groups. The first group contains m - f random input variables and for them a full factorial design is used to determine their values at the sampling points. For the second group containing the remaining f random input variables defining equations are used to derive their values at the sampling points from the settings of the variables in the first group.

As mentioned above, we want to use the value of the random output parameters obtained at the individual sampling points for fitting a response surface. This response surface is an approximation function that is determined by a certain number of terms and coefficients associated with these terms. Hence, the fraction f of a fractional factorial design cannot become too large, because otherwise there would not be enough data points in order to safely and accurately determine the coefficients of the response surface. In most cases a quadratic polynomial with cross-terms will be used as a response surface model. Therefore, the maximum value for the fraction f must be chosen such that a resolution V design is obtained (here V stands for the Roman numeral 5). A design with a resolution V is a design where the regression coefficients are not confounded with each other. A resolution V design is given if the defining equation mentioned above includes at least 5 random variables as a total on both sides of the equation sign. Please see Montgomery([312] (p. 938)) for details about fractional factorial designs and the use of defining equations.

For example with 5 random input variables X_1 to X_5 leads to a resolution V design if the fraction is f = 1. Consequently, a full factorial design is used to determine the probability levels of the random input variables X_1 to X_4. A defining equation is used to determine the probability levels at which the sampling

points are located for the random input variable X_5. See Montgomery([312] (p. 938)) for details about this example.

Location of Sampling Points Expressed in Random Variable Values

In order to obtain the values for the random input variables at each sampling point, the probabilities evaluated in the previous section must be transformed. To achieve this, the inverse transformation outlined under Common Features for all Probabilistic Methods (p. 903) can be used for non-correlated random variables. The procedure dealing with correlated random variables also mentioned under Common Features for all Probabilistic Methods (p. 903) can be used for correlated random variables.

18.3.4.2. Box-Behnken Matrix Design

Location of Sampling Points Expressed in Probabilities

For a Box-Behnken Matrix design, the sampling points are located at three different levels for each random input variable. In order to make the specification of these levels independent from the distribution type of the individual random input variables, it is useful to define these levels in terms of probabilities. The three different levels of a Box-Behnken Matrix design shall be denoted with p_1, with i = 1, ... , 3.

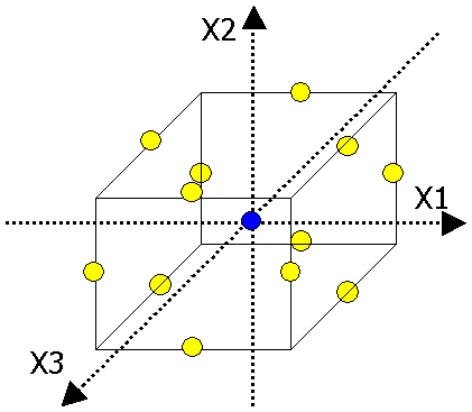
A Box-Behnken Matrix design is composed of two different parts, namely:

1. *Center point*: At the center point the values of all random input variables have a cumulative distribution function that equals p_2.

2. *Midside points*: For the midside points all random input variables except two are located at the p_2 probability level. The two other random input variables are located at probability levels with permutations of p_1 for the lower level and p_3 for the upper level.

See Box and Cox([307] (p. 938)) for further details. A sample set based on a central composite design for three random variables X_1, X_2 and X_3 is shown in Figure 18.13: Sample Set Based on Box-Behnken Matrix Design (p. 908).

Figure 18.13: Sample Set Based on Box-Behnken Matrix Design



For this example with three random input variables the matrix describing the location of the sampling points in terms of probabilities is shown in Table 18.2: Probability Matrix for Samples of Box-Behnken Matrix Design (p. 909).

Table 18.2: Probability Matrix for Samples of Box-Behnken Matrix Design

Sample	X_1	X_2	X_3	Part
1	p_2	p_2	p_2	Center
2	p_1	p_1	p_2	Midside Points
3	p_1	p_3	p_2	
4	p_3	p_1	p_2	
5	p_3	p_3	p_2	
6	p_1	p_2	p_1	
7	p_1	p_2	p_3	
8	p_3	p_2	p_1	
9	p_3	p_2	p_3	
10	p_2	p_1	p_1	
11	p_2	p_1	p_3	
12	p_2	p_3	p_1	
13	p_2	p_3	p_3	

Location of Sampling Points Expressed in Random Variable Values

In order to obtain the values for the random input variables at each sampling point, the same procedure is applied as mentioned above for the Central Composite Design.

18.4. Regression Analysis for Building Response Surface Models

Regression analysis is a statistical methodology that utilizes the relation between two or more quantitative variables so that one dependent variable can be estimated from the other or others.

In the following $\{X\} = \lfloor X_1\ X_2 \ldots X_m \rfloor^T$ denotes the vector of input variables, where m is the number of input variables. An arbitrary location in the space of input variables is denoted with $\{x\} = \lfloor x_1\ x_2 \ldots x_m \rfloor^T$ and $\{x\}_i = \lfloor x_1\ x_2 \ldots x_m \rfloor_i^T$ indicates the ith sampling point in the space of the input variables. Y is the name an output parameter, whereas y denotes a specific value of that output parameter and y_i is the value of the output parameter corresponding to the ith sampling point.

A regression analysis assumes that there are a total of n sampling points and for each sampling point $\{x\}_i$ with i = 1, ... , n the corresponding values of the output parameters y_i are known. Then the regression analysis determines the relationship between the input variables {X} and the output parameter Y based on these sample points. This relationship also depends on the chosen regression model. Typically for the regression model, either a first or a second order polynomial is preferred. In general, this regression model is an approximation of the true input-to-output relationship and only in special cases does it yield a true and exact relationship. Once this relationship is determined, the resulting approximation of the output parameter Y as a function of the input variables {X} is called the response surface.

Without loss of generality, it is assumed in the following that there is only one output parameter Y, but the procedure can be applied in the same way to process multiple output parameters.

In general, there are two types of regression analyses:

1. *Linear regression analysis*. A linear regression analysis assumes that the regression model is a linear function with respect to the parameters of the regression model, i.e., the regression parameters are the coefficients of the regression terms.

2. *Nonlinear regression analysis*. For a nonlinear regression analysis, the regression model is a nonlinear function with respect to the parameters of the regression model.

Here, we focus on linear regression only. In Transformation of Random Output Parameter Values for Regression Fitting (p. 913) we introduce the concept of nonlinear transformation functions that are applied on the values of the output parameters y_i. In principle, using nonlinear transformation function changes the nature of the regression analysis from a linear to a nonlinear regression analysis. However, in this special case we can treat the problem as a linear regression analysis because it is linear with respect to the transformed values of the output parameters.

18.4.1. General Definitions

The error sum of squares SSE is:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = (\{y\} - \{\hat{y}\})^T (\{y\} - \{\hat{y}\}) \tag{18.47}$$

where:

y_i = value of the output parameter at the ith sampling point

$\hat{y}_i$ = value of the regression model at the ith sampling point

The regression sum of squares SSR is:

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \tag{18.48}$$

where:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

The total sum of squares SST is:

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{18.49}$$

For linear regression analysis the relationship between these sums of squares is:

$$SST = SSR + SSE \tag{18.50}$$

For nonlinear regression analysis, Equation 18.50 (p. 910) does not hold.

18.4.2. Linear Regression Analysis

For a linear regression analysis the regression model at any sampled location $\{x\}_i$, with i = 1, ... , n in the m-dimensional space of the input variables can be written as:

$$y_i = \lfloor t \rfloor_i \{c\} + \varepsilon \tag{18.51}$$

where:

$\lfloor t \rfloor_i$ = row vector of regression terms of the response surface model at the ith sampled location

$\{c\} = \lfloor c_1\ c_2 \ldots c_p \rfloor^T$ = vector of the regression parameters of the regression model

p = total number of regression parameters. For linear regression analysis, the number of regression parameters is identical to the number of regression terms.

For a fully quadratic regression model, the vector of regression terms at the ith sampled location is:

$$\lfloor t \rfloor_i = \left\lfloor 1\ \ x_{1,i}\ \ x_{2,i} \ldots x_{m,i}\ \ x_{1,i}^2\ \ x_{1,i}x_{2,i} \ldots x_{1,i}x_{m,i}\ \ x_{2,i}^2 \ldots x_{2,i}x_{m,i} \ldots x_{m,i}^2 \right\rfloor \tag{18.52}$$

The total number of regression terms of a fully quadratic regression model is:

$$p = 1 + m + \frac{1}{2}(m+1)m \tag{18.53}$$

Equation 18.51 (p. 911) is called the normal error regression model, because the error term ε is assumed to have a normal distribution with zero mean value and a constant variance. The expression "constant variance" means that the variance of the error term is identical for all sampled locations $\{x\}_i$. For all sampling points Equation 18.51 (p. 911) can be written in matrix form as:

$$\{y\} = \{\hat{y}\} + \{\varepsilon\} = [d]\{c\} + \{\varepsilon\} \tag{18.54}$$

where:

$\hat{y}$ = vector of the values of the approximation of the response parameter based on the response surface model at all sampled locations

$$[d] = \begin{bmatrix} \lfloor t \rfloor_1 \\ \vdots \\ \lfloor t \rfloor_n \end{bmatrix} = \text{design matrix}$$

$\{\varepsilon\} = \{\varepsilon, ..., \varepsilon\}^T$ = vector of error terms at all sampled locations

The parameters of the regression model are determined using the method of least squares, which is based on minimizing the sum of the squared errors:

$$SSE = \sum_{j=1}^{n} \varepsilon_j^2 = (\{y\} - [d]\{c\})^T (\{y\} - [d]\{c\}) \rightarrow \min \tag{18.55}$$

From this it follows that the regression coefficients can be calculated from:

$$\{c\} = ([d]^T [d])^{-1} [d]^T \{y\} \tag{18.56}$$

Once the regression coefficients {c} are determined using Equation 18.56 (p. 912), the response surface (as being the approximation of the output parameter y as a function of the input variables {x}) is:

$$\hat{y} = \lfloor t\{x\} \rfloor \{c\} \tag{18.57}$$

18.4.3. F-Test for the Forward-Stepwise-Regression

In the forward-stepwise-regression, the individual regression terms are iteratively added to the regression model if they are found to cause a significant improvement of the regression results. Here, a partial F-test is used to determine the significance of the individual regression terms. Assume that the regression model already includes p terms, namely, $T_1, T_2, \ldots, T_p$, where p is the number of the terms in the regression model and p is smaller than the maximum number of terms in the regression model, i.e., we have only selected a subset of all possible regression terms. To determine if an additional term T_{p+1} would be a significant improvement of the regression model, we need to calculate the following characteristic value:

$$F^*_{p+1} = \frac{\frac{SSE_p - SSE_{p+1}}{\nu_p - \nu_{p+1}}}{\frac{SSE_{p+1}}{\nu_{p+1}}} \tag{18.58}$$

where:

F^*_{p+1} = partial Fisher F-test statistic
SSE_p = error sum of squares of the regression model with the p terms
SSE_{p+1} = error sum of squares of the regression model with the p+1 terms
ν_p = n - p = degrees of freedom of the regression model with the p terms
ν_{p+1} = n - (p + 1) = degrees of freedom of the regression model with the p+1 terms

An additional term T_{p+1} is considered to be a significant improvement for the regression model only if the following condition is satisfied:

$$F^*_{p+1} > F(1-\alpha \mid 1, n-(p+1)) \tag{18.59}$$

where:

$F(... \mid \nu_1, \nu_2)$ = inverse cumulative distribution function of the Fisher F-distribution with ν_1 numerator degrees of freedom and ν_2 denominator degrees of freedom
α = significance level

Usually there is a choice of several terms that are considered for inclusion in the regression model. In other words, if we currently only have a subset of all possible terms selected then there is more then one term that is not yet selected. In this case we choose that term which delivers the maximum F^*_{p+1}-value according to Equation 18.58 (p. 912) and satisfies the condition in Equation 18.59 (p. 912).

The forward-stepwise-regression also involves a significance test of all p terms that are already included in the regression model to see if they are still significant after an additional term T_{p+1} has been included. This significance test is also based on Equation 18.58 (p. 912) and any of the previously included p terms will be taken away from the regression model for which the condition in Equation 18.59 (p. 912) is no longer satisfied. See Neter et al.([314] (p. 938)) for details about the forward-stepwise-regression.

18.4.4. Transformation of Random Output Parameter Values for Regression Fitting

Only in special cases can random output parameters of a finite element analysis such as displacements or stresses be exactly described by a second order polynomial as a function of the random input parameters. Usually a second order polynomial provides only an approximation. The quality of the approximation can be significantly improved by applying a transformation to the random output parameter values y_i, i = 1, ..., n, before fitting a response surface. The transformed values of the random output parameters shall be denoted with y_i^*. The following transformations are available:

1. Exponential: $y_i^* = \exp(y_i)$
2. Logarithm with a user-defined base a: $y_i^* = \log_a(y_i)$
3. Natural logarithm: $y_i^* = \ln(y_i)$
4. Logarithm with a base 10: $y_i^* = \log_{10}(y_i)$
5. Square Root: $y_i^* = \sqrt{y_i}$
6. Power Transformation with a user-defined exponent a: $y_i^* = y_i^a$
7. Box-Cox Transformation (see Box and Cox([307] (p. 938))):

$$y_i^* = \begin{cases} \dfrac{y_i^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \ln(y_i) & \lambda = 0 \end{cases}$$

Fitting of a second order polynomial response surface takes place after this transformation, i.e., the transformed values of the random output parameter y_i^* are used for the regression analysis. After the regression coefficients have been determined the evaluation of the value of the response surface approximation $\hat{y}$ requires a back-transformation using the inverse function of the transformation listed above.

It should be noted that the transformations mentioned above are nonlinear functions. Therefore, the regression analysis is a linear regression in terms of the transformed values of the random output

parameter y_i^* , but it is a nonlinear regression with respect to the original values of the random output parameter y_i.

18.4.5. Goodness-of-Fit Measures

Goodness-of-fit measures express how well or how accurately a response surface represents the sample points the response surface is based on. It should be noted that the goodness-of-fit measures always indicate a very accurate fit if there are not enough sample points. For example, the response surface will always exactly fit through the underlying sample points if the number of sample points n is identical to the number of coefficients p in the regression model. However, this does not mean that the response surface is an exact representation of the true input-output relationship. Example: If we only have two sample points, we can always fit a straight line exactly through these two sample points. That, however, does not necessarily mean that this straight line correctly represents the true input-output relationship.

18.4.5.1. Error Sum of Squares SSE

The error sum of squares as a measure for the goodness-of-fit of a response surface is calculated using Equation 18.47 (p. 910). A good fit is achieved if the error sum of squares SSE is as close as possible to zero.

18.4.5.2. Coefficient of Determination R^2

The coefficient of determination is often called the R-squared measure. It is calculated with the equation:

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{18.60}$$

A good fit is achieved if the coefficient of determination is as close as possible to 1.0. A value of 1.0 indicates that the response surface model explains all of the variability of the output parameter Y. It should be noted that for a nonlinear regression analysis, the coefficient of determination is not a suitable measure for the goodness-of-fit. This is because the error sum of squares SSE and the regression sum of squares SSR do not add up to the total sum of squares SST. For this case the coefficient of determination may become larger than 1.0. If this happens the value is truncated to 1.0. See Neter et al.([314] (p. 938)) for details about the coefficient of determination.

18.4.5.3. Maximum Absolute Residual

The maximum absolute residual as a measure for the goodness-of-fit is given by the equation:

$$y_{res,max} = \max(|y_1|, |y_2|, \ldots, |y_n|) \tag{18.61}$$

A good fit is achieved if the maximum absolute residual is as close to 0.0 as possible.

18.5. Probabilistic Postprocessing

Regardless which probabilistic method has been used to generate probabilistic result data, the postprocessing of the data is always based on a statistical evaluation of sampled data. Let X be a random variable

with a certain but arbitrary cumulative distribution function F_X. Each sample of size n will be a set of $\lfloor x_1\ x_2 \ldots x_n \rfloor^T$, which will be used for the probabilistic postprocessing. The statistical analysis of sample data is based on some assumptions. One key assumption is the independence within the samples or, in other words, the observations $\lfloor x_1\ x_2 \ldots x_n \rfloor^T$ are independent. This means that the results of one sample do not depend in any way on the results of another sample. This assumption is typically valid for numerical experiments. Another assumption is the Central Limit Theorem. It states that for a set of independent random variables $\lfloor X_1\ X_2 \ldots X_n \rfloor^T$ with identical distribution the sum of these random variables as well as the arithmetic mean will have approximately a Gaussian distribution, if the sample size n is sufficiently large. Furthermore, it is assumed that the true cumulative distribution function F_X is unknown, but can be approximated by the empirical cumulative distribution function derived from the set of observations $\lfloor x_1\ x_2 \ldots x_n \rfloor^T$.

In some cases, probabilistic postprocessing requires the comparison of the sampled data from two random variables. In this case we use X as the first random variable with $\lfloor x_1\ x_2 \ldots x_n \rfloor^T$ as the set of sampled observations and Y as the second random variable with $\lfloor y_1\ y_2 \ldots y_n \rfloor^T$ as the set of sampled observations. The same assumptions explained above for the random variable X apply in a similar manner for the random variable Y.

The statistical characteristics of sampled data are always random variables themselves, as long as the sample size n is finite. Therefore, it is necessary to estimate the accuracy of the statistical characteristics using confidence intervals or limits. In this discussion, a two-sided confidence interval is referred to as a confidence interval, and a one-sided confidence interval is referred to as a confidence limit. The width of confidence intervals is characterized by the probability of falling inside or outside the confidence interval. The probability of the statistical characteristic of the sampled data falling outside the confidence interval is usually denoted with the symbol α. Consequently, the probability of the statistical characteristic of the sampled data falling inside the confidence interval is 1-α.

18.5.1. Statistical Procedures

18.5.1.1. Mean Value

An estimate for the mean value of a random variable X derived from a sample of size n is:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{18.62}$$

The estimate of the mean value is a random variable itself and it converges to the true mean value m of the random variable X if the sample size n tends to infinity. By virtue of the central limit theorem, the distribution of the estimate of the mean value can be assumed as a Gaussian distribution. Hence, the 1 - α confidence interval is

$$\langle \mu \rangle_{1-\alpha} = \left[\bar{x} - t(1-0.5\alpha \mid n-1)\frac{s}{\sqrt{n}};\ \bar{x} + t(1-0.5\alpha \mid n-1)\frac{s}{\sqrt{n}} \right] \tag{18.63}$$

where:

t (... | n - 1) = inverse cumulative distribution function of the Student's t- distribution with n - 1 degrees of freedom
s = the estimate for the standard deviation of the sample data as given by Equation 18.64 (p. 916)

The confidence interval should be interpreted as follows: "There is a 1 - α confidence that the estimated interval contains the unknown, true mean value m" (Ang and Tang([304] (p. 938))).

18.5.1.2. Standard Deviation

An estimate for the standard deviation of a random variable X derived from a sample of size n is:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{18.64}$$

The estimate of the standard deviation is a random variable itself and it converges to the true standard deviation σ of the random variable X if the sample size n tends to infinity. The 1 - α confidence interval is:

$$\langle\sigma\rangle_{1-\alpha} = \left[s\sqrt{\frac{n-1}{\chi^{2^{-1}}}(1-0.5\alpha \mid n-1)};\ s\sqrt{\frac{n-1}{\chi^{2^{-1}}}(0.5\alpha \mid n-1)} \right] \tag{18.65}$$

where:

$\chi^{2^{-1}}$ (...|n - 1) = inverse of the cumulative distribution function of a chi-square distribution with n - 1 degrees of freedom

The confidence interval should be interpreted as follows: "There is a 1 - α confidence that the estimated interval contains the unknown, true standard deviation σ" (Ang and Tang([304] (p. 938))).

18.5.1.3. Minimum and Maximum Values

The minimum and the maximum values of the set of observations are:

$$x_{min} = \min(x_1, x_2, \ldots, x_n) \tag{18.66}$$

$$x_{max} = \max(x_1, x_2, \ldots, x_n) \tag{18.67}$$

Since every observed value is unpredictable prior to the actual observation, it can be assumed that each observation is a realization of the set of the sample random variables $\lfloor X_1\ X_2 \ldots X_n \rfloor^T$. The minimum and the maximum of the sample random variables are:

$$X_{min} = \min(X_1, X_2, \ldots, X_n) \tag{18.68}$$

$$X_{max} = \max(X_1, X_2, \ldots, X_n) \tag{18.69}$$

This means that the minimum and the maximum of a sample of size n taken from a population X are also random variables. For the minimum value, only an upper confidence limit can be given and for the maximum value only a lower confidence limit can be derived. Since the $X_1, X_2, \ldots, X_n$ are statistically independent and identically distributed to X, the upper confidence limit of the minimum value and the lower confidence limit of the maximum value are:

$$(x_{min}\rangle_{1-\alpha} = F_X^{-1}(1-\alpha^{1/n}) \tag{18.70}$$

$$\langle x_{max} \rangle_{1-\alpha} = F_X^{-1}(\alpha^{1/n}) \tag{18.71}$$

Obviously, the evaluation of the confidence limits requires the computation of the inverse cumulative distribution function of the random variable X based on sampled data. This is explained in Inverse Cumulative Distribution Function (p. 919).

The upper confidence limit of the minimum value should be interpreted as follows: "There is a 1 - α confidence that the unknown, true minimum value is below the estimated upper limit" (Ang and Tang([305] (p. 938))). An analogous interpretation should be applied for the lower confidence limit of the maximum value.

18.5.2. Correlation Coefficient Between Sampled Data

18.5.2.1. Pearson Linear Correlation Coefficient

The Pearson linear correlation coefficient (Sheskin([315] (p. 938))) is:

$$r_P = \frac{\sum_i^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i^n (x_i - \bar{x})^2} \sqrt{\sum_i^n (y_i - \bar{y})^2}} \tag{18.72}$$

Since the sample size n is finite, the correlation coefficient r_p is a random variable itself. Hence, the correlation coefficient between two random variables X and Y usually yields a small, but nonzero value, even if X and Y are not correlated at all in reality. In this case, the correlation coefficient would be insignificant. Therefore, we need to find out if a correlation coefficient is significant or not. To determine the significance of the correlation coefficient, we assume the hypothesis that the correlation between X and Y is not significant at all, i.e., they are not correlated and $r_p = 0$ (null hypothesis). In this case the variable:

$$t = r_P \sqrt{\frac{n-2}{1-r_P^2}} \tag{18.73}$$

is approximately distributed like the Student's t-distribution with $\nu = n - 2$ degrees of freedom. The cumulative distribution function Student's t-distribution is:

$$A(t \mid \nu) = \frac{1}{\sqrt{\nu}\, B\left(\frac{1}{2}, \frac{\nu}{2}\right)} \int_{-t}^{t} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} dx \tag{18.74}$$

where:

B(...) = complete Beta function

There is no closed-form solution available for Equation 18.74 (p. 917). See Abramowitz and Stegun([303] (p. 938)) for more details.

The larger the correlation coefficient r_p, the less likely it is that the null hypothesis is true. Also the larger the correlation coefficient r_p, the larger is the value of t from Equation 18.73 (p. 917) and consequently also the probability $A(t|\nu)$ is increased. Therefore, the probability that the null hypothesis is true is given

by 1-A(t|ν). If 1-A(t|ν) exceeds a certain significance level, for example 1%, then we can assume that the null hypothesis is true. However, if 1-A(t|ν) is below the significance level then it can be assumed that the null hypotheses is not true and that consequently the correlation coefficient r_p is significant.

18.5.2.2. Spearman Rank-Order Correlation Coefficient

The Spearman rank-order correlation coefficient (Sheskin([315] (p. 938))) is:

$$r_s = \frac{\sum_i^n (R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_i^n (R_i - \bar{R})^2}\sqrt{\sum_i^n (S_i - \bar{S})^2}} \tag{18.75}$$

where:

R_i = rank of x_i within the set of observations $\lfloor x_1\ x_2 \ldots x_n \rfloor^T$

S_i = rank of y_i within the set of observations $\lfloor y_1\ y_2 \ldots y_n \rfloor^T$

$\bar{R}, \bar{S}$ = average ranks of a R_i and S_i respectively

The significance of the Spearman rank-order correlation coefficient r_s is determined in the same way as outlined for the Pearson linear correlation coefficient above.

18.5.3. Cumulative Distribution Function

The cumulative distribution function of sampled data is also called the empirical distribution function. To determine the cumulative distribution function of sampled data, it is necessary to order the sample values in ascending order. Let x_i be the sampled value of the random variable X having a rank of i, i.e., being the ith smallest out of all n sampled values. The cumulative distribution function F_i that corresponds to x_i is the probability that the random variable X has values below or equal to x_i. Since we have only a limited amount of samples, the estimate for this probability is itself a random variable. According to Kececioglu([310] (p. 938)), the cumulative distribution function F_i associated with x_i is:

$$\sum_{k=i}^{n} \frac{n!}{(n-k)!k!} F_i^k (1-F_i)^{n-k} = 50\% \tag{18.76}$$

Equation 18.76 (p. 918) must be solved numerically. The lower and upper confidence limits of a 1 - α confidence interval are directly obtained in a similar way. The lower confidence limit can be determined from:

$$\sum_{k=i}^{n} \frac{n!}{(n-k)!k!} \langle F_i \rangle_{\alpha/2}^k (1-\langle F_i \rangle_{\alpha/2})^{n-k} = \frac{\alpha}{2} \tag{18.77}$$

$$\sum_{k=i}^{n} \frac{n!}{(n-k)!k!} \langle F_i \rangle_{1-\alpha/2}^k (1-\langle F_i \rangle_{1-\alpha/2})^{n-k} = 1-\frac{\alpha}{2} \tag{18.78}$$

18.5.4. Evaluation of Probabilities From the Cumulative Distribution Function

The cumulative distribution function of sampled data can only be given at the individual sampled values $x_1, x_2, ..., x_i, x_{i+1}, ..., x_n$ using Equation 18.76 (p. 918). Hence, the evaluation of the probability that the random variable is less or equal an arbitrary value x requires an interpolation between the available data points.

If x is for example between x_i and x_{i+1} then the probability that the random variable X is less or equal to x is:

$$P(X \le x) = F_i + (F_{i+1} - F_i)\frac{x - x_i}{x_{i+1} - x_i} \tag{18.79}$$

The confidence interval for the probability $P(X \le x)$ can be evaluated by interpolating on the confidence interval curves using the same approach.

18.5.5. Inverse Cumulative Distribution Function

The cumulative distribution function of sampled data can only be given at the individual sampled values $x_1, x_2, ..., x_i, x_{i+1}, ..., x_n$ using Equation 18.76 (p. 918). Hence, the evaluation of the inverse cumulative distribution function for any arbitrary probability value requires an interpolation between the available data points.

The evaluation of the inverse of the empirical distribution function is most important in the tails of the distribution. In the tails of the distribution, the slope of the empirical distribution function is very flat. In this case a direct interpolation between the points of the empirical distribution function similar to Equation 18.79 (p. 919) can lead to very inaccurate results. Therefore, the inverse standard normal distribution function Φ^{-1} is applied for all probabilities involved in the interpolation. If p is the requested probability for which we are looking for the inverse cumulative distribution function value and p is between F_i and F_{i+1}, then the inverse cumulative distribution function value can be calculated using:

$$x = x_i + (x_{i+1} - x_i)\frac{\Phi^{-1}(p) - \Phi^{-1}(F_i)}{\Phi^{-1}(F_i + 1) - \Phi^{-1}(F_i)} \tag{18.80}$$

The confidence interval for x can be evaluated by interpolating on the confidence interval curves using the same approach.

Bibliography

[1] Ahmad, S., Irons, B. M. and Zienkiewicz, O. C.. *"Analysis of Thick and Thin Shell Structures by Curved Finite Elements"*. *International Journal for Numerical Methods in Engineering*. Vol. 2, No. 3. 419-451. 1970.

[2] K. J. Bathe. *Finite Element Procedures*. Prentice-Hall. Englewood Cliffs. 1996.

[3] M. A. Biot. *Mechanics of Incremental Deformation*. John Wiley and Sons. New York, 1965.

[4] L. H. Chen. *"Piping Flexibility Analysis by Stiffness Matrix"*. *ASME, Journal of Applied Mechanics*. December 1959.

[5] R. D. Cook. *Concepts and Applications of Finite Element Analysis, Second Edition*. John Wiley and Sons. New York. 1981.

[6] R. D. Cook. *"Two Hybrid Elements for Analysis of Thick, Thin and Sandwich Plates"*. *International Journal for Numerical Methods in Engineering*. Vol. 5, No. 2. 277-288. 1972.

[7] D. F. Cuniff and G. J. O'Hara. *"Normal Mode Theory for Three-Directional Motion"*. *NRL Report*. 6170. U. S. Naval Research Laboratory. Washington D. C.. 1965.

[8] M. M. Denn. *Optimization by Variational Methods*. McGraw-Hill. New York. 1969.

[9] K. D. Henshell. *"Automatic Masters for Eigenvalue Economization"*. *Earthquake Engineering and Structural Dynamics*. Vol. 3. 375-383. 1975.

[10] M. C. Imgrund. *ANSYS® Verification Manual*. Swanson Analysis Systems, Inc.. 1992.

[11] W. Flugge. *Stresses in Shells*. Springer Verla. Berlin. 1967.

[12] R. J. Fritz. *"The Effect of Liquids on the Dynamic Motions of Immersed Solids"*. *ASME Journal of Engineering for Industry*. February, 1972.

[13] T. V. Galambos. *Structural Members and Frames*. Prentice-Hall. Englewood Cliffs. 1968.

[14] R. J. Guyan. *"Reduction of Stiffness and Mass Matrices"*. *AIAA Journal*. Vol. 3, No. 2. February, 1965.

[15] A. S. Hall and R. W. Woodhead. *Frame Analysis*. John Wiley and Sons. New York. 1961.

[16] C. Rajakumar and C. R. Rogers. *"The Lanczos Algorithm Applied to Unsymmetric Generalized Eigenvalue Problem"*. *International Journal for Numerical Methods in Engineering*. Vol. 32. 1009-1026. 1992.

[17] B. M. Irons. *"A Frontal Solution Program for Finite Element Analysis"*. *International Journal for Numerical Methods in Engineering*. Vol. 2, No. 1. January, 1970, 5-23 Discussion May, 1970, p. 149.

[18] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press. Oxford. 515-569. 1988.

[19] P. C. Kohnke and J. A. Swanson. *"Thermo-Electric Finite Elements"*. *Proceedings, International Conference on Numerical Methods in Electrical and Magnetic Field Problems*. Santa Margherita Liqure Italy. June 1-4, 1976.

[20] P. C. Kohnke. *"Large Deflection Analysis of Frame Structures by Fictitious Forces"*. *International Journal of Numerical Methods in Engineering*. Vol. 12, No. 8. 1278-1294. 1978.

[21] C. F. Kollbrunner and K. Basler. *Torsion in Structures*. Springer-Verlag. Berlin. 1969.

[22] E. J. Konopinski. *Classical Descriptions of Motion*. Freeman and Company. San Francisco. 1969.

[23] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley and Sons, Inc.. New York. 1962.

[24] S. G. Lekhnitskii. *Theory of Elasticity of an Anisotropic Elastic Body*. Holden-Day. San Francisco. 1963.

[25] R. J. Melosh and R. M. Bamford. *"Efficient Solution of Load-Deflection Equations"*. *ASCE Journal of the Structural Division*. Vol. 95, No. ST4, Proc. Paper 6510, Apr., 1969. 661-676 Discussions Dec., 1969, Jan., Feb., May, 1970, Closure, Feb., 1971.

[26] Kanok-Nukulchai. *"A Simple and Efficient Finite Element for General Shell Analysis"*. *International Journal for Numerical Methods in Engineering*. Vol. 14. 179-200 . 1979.

[27] J. T. Oden. *Mechanics of Elastic Structures*. McGraw-Hill. New York . 1968.

[28] J. S. Przemieniecki. *Theory of Matrix Structural Analysis*. McGraw-Hill. New York. 1968.

[29] W. C. Schnobrich and M. Suidan. *"Finite Element Analysis of Reinforced Concrete"*. *ASCE Journal of the Structural Division*. ST10. 2109-2122 . October, 1973.

[30] P. Seide. *"Large Deflection of Rectangular Membranes Under Uniform Pressure"*. *International Journal of Non-Linear Mechanics*. Vol. 12. 397-406.

[31] L. Skjelbreia and J. A. Hendrickson. *"Fifth Order Gravity Wave Theory"*. *Proceedings, Seventh Conference on Coastal Engineering*. Ch. 10, 184-196. 1961.

[32] S. Timoshenko and S. Woinowskey-Kreiger. *Theory of Plates and Shells*. McGraw-Hill. New York . 1959.

[33] D. M. Tracey. *"Finite Elements for Three Dimensional Elastic Crack Analysis"*. *Nuclear Engineering and Design*. Vol. 26. 1973.

[34] E. H. Vanmarcke. *"Structural Response to Earthquakes"*. *Seismic Risk and Engineering Decisions*. Elsvier Scientific Publishing Co.. Amsterdam-Oxford, New York. edited by C. Lomnitz and E. Rosemblueth. 287-337. 1976.

[35] J. D. Wheeler. *"Method of Calculating Forces Produced by Irregular Waves"*. *Journal of Petroleum Technology*. Vol. 22. 359-367. 1970.

[36] K. J. Willam. University of Colorado, Boulder. , *Private Communication*. 1982.

[37] K. J. Willam and E. D. Warnke. *"Constitutive Model for the Triaxial Behavior of Concrete"*. *Proceedings, International Association for Bridge and Structural Engineering*. Vol. 19. ISMES. Bergamo, Italy. p. 174. 1975.

[38] E. L. Wilson, R. L. Taylor, W. P., Doherty, and J. Ghaboussi. *"Incompatible Displacement Models"*. *Numerical and Computer Methods in Structural Mechanics*. edited by S. J. Fenves, et al. Academic Press, Inc.. N. Y. and London. 43-57. 1973.

[39] O. C. Zienkiewicz. *The Finite Element Method*. McGraw-Hill Company. London. 1977.

[40] *ASME Boiler and Pressure Vessel Code, Section III, Division 1, Subsection NC, Class 2 Components*. 1974.

[41] *"Regulatory Guide"*. Published by the U. S. Nuclear Regulatory Commission, Regulatory Guide 1.92, Revision 1. February 1976.

[42] Kumar K. Tamma and Raju R. Namburu. *"Recent Advances, Trends and New Perspectives Via Enthalpy-Based Finite Element Formulations for Applications to Solidification Problems"*. *International Journal for Numerical Methods in Engineering*. Vol. 30. 803-820. 1990.

[43] *Shore Protection Manual, Published by the U. S. Army Coastal Engineering Research Center*. Vol. I, Third Edition. 1977.

[44] F. P. Beer and R. E. Johnston. *Vector Mechanics for Engineers, Statics and Dynamics*. McGraw-Hill. New York. 1962.

[45] E. Hinton, A. Rock, and O. Zienkiewicz. *"A Note on Mass Lumping and Related Processes in the Finite Element Method"*. *International Journal of Earthquake Engineering and Structural Dynamics*. Vol. 4. 245-249 . 1976.

[46] R. D. Krieg and D. B. Krieg. *"Accuracies of Numerical Solution Methods for the Elastic-Perfectly Plastic Model"*. *Journal of Pressure Vessel Technology*. *Transactions of the ASME*. Vol. 99 No. 4, Series J. 510-515. November, 1977.

[47] William T. Thomson. *Theory of Vibrations with Applications*. Prentice Hall. 343-352. 1971.

[48] R. J. Roark and W. C. Young. *Formulas for Stress and Strain*. McGraw-Hill. New York. 1975.

[49] R. L. Taylor, P. J. Beresford, and E. L. Wilson. *"A Non-Conforming Element for Stress Analysis"*. *International Journal for Numerical Methods in Engineering*. Vol. 10. 1211-1219. 1976.

[50] R. Hill. *The Mathematical Theory of Plasticity*. Oxford University Press. New York. 1983.

[51] C. F. Shih. D. Lee. *"Further Developments in Anisotropic Plasticity"*. *Journal of Engineering Materials and Technology*. Vol. 100. 294-302. July 1978.

[52] S. Valliappan. *"Nonlinear Analysis for Anistropic Materials"*. *International Journal for Numerical Methods in Engineering*. Vol. 10. 597-606. 1976.

[53] J. F. Besseling. *"A Theory of Elastic, Plastic, and Creep Deformations of an Initially Isotropic Material Showing Aisotropic Strain-Hardening Creep Recovery and Secondary Creep"*. *Journal of Applied Mechanics*. 529-536. December 1958.

[54] R. J. Owen, A. Prakash, and O. C. Zienkiewicz. *"Finite Element Analysis of Non-Linear Composite Materials by Use of Overlay Sytems"*. *Computers and Structures, Pergamon Press*. Vol. 4. 1251-1267.

[55] J. P. Holman. *Heat Transfer*. Fourth Edition. McGraw-Hill. New York. 1976.

[56] J. L. Batoz, K. J. Bathe, and L. W. Ho. *"A Study of Three-Node Triangular Plate Bending Elements"*. *International Journal of Numerical Methods in Engineering*. Vol. 15. 1771-1812. 1980.

[57] A. Razzaque. *"On the Four Noded Discrete Kirchhoff Shell Elements"*. *Accuracy Reliability Training in FEM Technology*. edited by Robinson, J.. 473-483. 1984.

[58] P. M. Gresho and R. L. Lee. *"Don't Suppress the Wiggles - They're Telling You Something"*. *Finite Element Methods for Convection Dominated Flows*. Vol. 34. ASME Publication AMD. 37-61. 1979.

[59] R. G. Dean. *Evaluation and Development of Water Wave Theories for Engineering Application*. prepared for U. S. Army Corp of Engineers, Coastal Engineering Research Center. November 1974.

[60] *ASME Boiler and Pressure Vessel Code*. Section III, Division 1-1974, Subsection NB, Class 1 Components.

[61] American National Standard Code for Pressure Piping, Power Piping, ANSI B31.1-1977, Published by the American Society of Mechanical Engineers.

[62] R. M. Orris and M. Petyt. *"Finite Element Study of Harmonic Wave Propagation in Periodic Structures"*. *Journal of Sound and Vibration*. 223-236. 1974.

[63] J. L. Gordon. *"OUTCUR: An Automated Evaluation of Two-Dimensional Finite Element Stresses" according to ASME*. Paper No. 76-WA/PVP-16. ASME Winter Annual Meeting. December 1976.

[64] M. J. D. Powell. *"An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives"*. *Computer Journal*. Vol. 7. 155-162. 1964.

[65] E. L. Wilson, A. Der Kiereghian, and E. Bayo. *"A Replacement for the SRSS Method in Seismic Analysis"*. *Earthquake and Structural Dynamics*. Vol. 9, No. 2. University of California, Berkeley. 187. March 1981.

[66] C. C. Rankin. F. A. Brogan. *"An Element Independent Corotational Procedure for the Treatment of Large Rotations"*. *Journal of Pressure Vessel Technology*. Vol. 108. 165-174. May 1986.

[67] J. Argyris. *"An Excursion into Large Rotations"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 32. 85-155. 1982.

[68] S. Tse, I. E. Morse, and R. T. Hinkle. *Mechanical Vibrations*. Allyn and Bacon. Boston. 1963.

[69] M. V. K. Chari. *"Finite Element Solution of the Eddy Current Problem in Magnetic Structures"*. *IEEE Transactions on Power Apparatus and Systems*. Vol. PAS-93. 62-72 . 1974.

[70] J. R. Brauer. *"Finite Element Analysis of Electromagnetic Induction in Transformers"*. paper A77-122-5, IEEE Winter Power Meeting. New York City. 1977.

[71] S. C. Tandon. M. V. K. Chari. *"Transient Solution of the Diffusion Equation by the Finite Element Method"*. *Journal of Applied Physics*. March 1981.

[72] P. P. Silvester, H. S. Cabayan, and B. T. Browne. *"Efficient Techniques for Finite Element Analysis of Electric Machines"*. *IEEE Transactions on Power Apparatus and Systems*. Vol. PAS-92. 1274-1281. 1973.

[73] M. V. K. Chari and J. D'Angelo. *"Finite Element Analysis of Magneto-Mechanical Devices"*. *Fifth International Workshop in Rare Earth-Cobalt Permanent Magnets and Their Application*. 7-10, Paper No. V1-1.. Roanoke, VA. June 1981.

[74] O. W. Anderson. *"Transform Leakage Flux Program Based on the Finite Element Method"*. *IEEE Transactions on Power Apparatus and Systems*. Vol. PAS-92, No. 2. 1973.

[75] O. C. Zienkiewicz, J. Lyness, and D. R Owen. *"Three-Dimensional Magnetic Field Determination Using a Scalar Potential - A Finite Element Solution"*. *IEEE Transactions on Magnetics*. Vol. MAG-13, No. 5. 1649-1656. 1977.

[76] J. L. Coulomb and G. Meunier. *"Finite Element Implementation of Virtual Work Principle for Magnetic for Electric Force and Torque Calculation"*. *IEEE Transactions on Magnetics*. Vol. Mag-2D, No. 5. 1894-1896. 1984.

[77] F. C. Moon. *Magneto-Solid Mechanics*. John Wiley and Sons. New York. 1984.

[78] A. J. Baker. *Finite Element Computational Fluid Mechanics*. McGraw-Hill Book Company. New York. 266-284. 1983.

[79] S. NW. Yuan. *Foundations of Fluid Mechanics*. Prentice-Hall International, Inc.. London. 71-102. 1976.

[80] Ray W. Clough and Joseph Penzien. *Dynamics of Structures*. McGraw-Hill. New York. p. 559. 1975.

[81] H. Allik and J. R. Hughes. *"Finite Element for Piezoelectric Vibration". International Journal Numerical Methods of Engineering*. No. 2. 151-157. 1970.

[82] N. P. Eer Nisse. *"Variational Method for Electroelastic Vibration Analysis". IEEE Transactions on Sonics and Ultrasonics*. Vol. SU-14, No. 4. 1967.

[83] J. Sato, M. Kawabuchi, and A. Fukumoto. *"Dependence of the Electromechanical Coupling Coefficient on the Width-to-Thickness Ratio of Plant-Shaped Piezoelectric Transducers Used for Electronically Scanned Ultrasound Diagnostic Systems". Journal of Acoustics Society of America*. No. 66 6. 1609-1611 . 1979.

[84] E. L. Kinsler. et. al.. *Fundamentals of Acoustics*. John Wiley and Sons. New York. 98-123. 1982.

[85] A. Craggs. *"A Finite Element Model for Acoustically Lined Small Rooms". Journal of Sound and Vibration*. Vol. 108, No. 2. 327-337.

[86] O. C. Zienkiewicz. R. E. Newton. *"Coupled Vibrations of a Structure Submerged in a Compressible Fluid". Proceedings of the Symposium on Finite Element Techniques*. University of Stuttgart. Germany. June 1969.

[87] Lawrence E. Malvern. *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall, Inc.. Englewood Cliffs, NJ. 1969.

[88] R. Siegal and J. R. Howell. *Thermal Radiation Heat Transfer*. Second Edition. Hemisphere Publishing Corporation. 1981.

[89] *ANSI/IEEE Standard on Piezoelectricity IEEE Standard*. 176. 1987.

[90] E. E. Antonova. D. C. Looman. *"Finite elements for thermoelectric device analysis in ANSYS". ICT 2005 24th International Conference on Thermoelectrics*. 215-218. 2005.

[91] E. Onate, J. Rojek, R. L. Taylor, and O. C. Zienkiewicz. *"Finite calculus formulation for incompressible solids using linear triangles and tetrahedra". International Journal for Numerical Methods in Engineering*. Vol. 59. 1473–1500. 2004.

[92] A. F. Fossum and J. T. Fredrich. *Cap plasticity model and compactive and dilatant prefailure deformation. Pacific Rocks 200: Rock Around the Rim*. A. A. Balkema. 1169-1176. 2000.

[93] Stephen W. Tsai. *Composites Design*. Third Edition, Section 11.6. Think Composites. Dayton, Ohio. 1987.

[94] J. Weiss. *"Efficient Finite Element Solution of Multipath Eddy Current Problems". IEEE Transactions on Magnetics*. Vol. MAG-18, No. 6. 1710-1712. 1982.

[95] V. K. Garg and J. Weiss. *"Finite Element Solution of Transient Eddy-Current Problems in Multiply-Excited Magnetic Systems". IEEE Transactions on Magnetics*. Vol. MAG-22, No. 5. 1257-1259 . 1986.

[96] E. N. Dvorkin. *"On Nonlinear Finite Element Analysis of Shell Structures"*. *Ph.D Thesis*. Massachusetts Institute of Technology. 1984.

[97] E. N. Dvorkin and K. J. Bathe. *"A Continuum Mechanics Based Four-Node Shell Element for General Nonlinear Analysis"*. *Engineering Computations*. Vol. 1. 77-88. 1984.

[98] K. J. Bathe and E. N. Dvorkin. *"A Formulation of General Shell Elements - The Use of Mixed Interpolation of Tensorial Components"*. *International Journal for Numerical Methods in Engineering*. Vol. 22. 697-722. 1986.

[99] M. Hoit and E. L. Wilson. *"An Equation Numbering Algorithm Based on a Minimum Front Criteria"*. *Computers and Structures*. Vol. 16. 225-239. 1983.

[100] E. Cuthill and J. McKee. *"Reducing the Band Width of Sparse Symmetric Matrices"*. *Proceedings of the ACM National Conference*. New York. 1969.

[101] A. Georges and D. McIntyre. *"On the Application of the Minimum Degree Algorithm to Finite Element Systems"*. *SIAM Journal of Numerical Analysis*. Vol. 15. 1978.

[102] O. C. Zienkiewicz and J. Z. Zhu. *"A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis"*. *International Journal for Numerical Methods in Engineering*. Vol. 24. 337-357. 1987.

[103] I. Babuska and W. C. Rheinboldt. *"Analysis of Optimal Finite Element Meshes in R"*. *Mathematics of Computation*. Vol. 33. 431-463. 1979.

[104] W. Carnegie. *"Vibrations of Rotating Cantilever Blading"*. *Journal of Mechanical Engineering Science*. Vol. 1. No. 3. 1959.

[105] P. G. Bergan and E. Mollestad. *"An Automatic Time-Stepping Algorithm for Dynamic Problems"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 49. 1985.

[106] P. C. Paris and G. C. Sih. *"Stress Analysis of Cracks"*. *Fracture Toughness and Testing and its Applications*. American Society for Testing and Materials. Philadelphia, STP 381. 30-83. 1965.

[107] G. J. O'Hara and R. O. Belsheim. *"Interim Design Values for Shock Design of Shipboard Equipment"*. *NRL Memorandum Report 1396*. U.S. Naval Research Laboratory. Washington D.C.. 1963.

[108] A. Markovsky, T. F. Soules, and M. R. Vukcevich. *"Mathematical and Computational Aspects of a General Viscoelastic Theory"*. *G. E. Lighting and Research and Technical Services Operation*. Report No. 86-LRL-2021. February 1986.

[109] G. W. Scherer and S. M. Rekhson. *"Viscoelastic-Elastic Composites: I, General Theory"*. *Journal of the American Ceramic Society*. Vol. 65, No. 7. 1982.

[110] O. S. Narayanaswamy. *"A Model of Structural Relaxation in Glass"*. *Journal of the American Ceramic Society*. Vol. 54, No. 10. 491-498. 1971.

[111] O. C. Zienkiewicz, M. Watson, and I. P. King. *"A Numerical Method of Visco-Elastic Stress Analysis"*. *International Journal of Mechanical Science*. Vol. 10. 807-827. 1968.

[112] R. L. Taylor, K. S. Pister, and G. L. Goudreas. *"Thermochemical Analysis of Viscoelastic Solids"*. *International Journal for Numerical Methods in Engineering*. Vol. 2. 45-59. 1970.

[113] D. J. Allman. *"A Compatible Triangular Element Including Vertex Rotations for Plane Elasticity Analysis"*. *Computers and Structures*. Vol. 19. 1-8. 1984.

[114] R. D. Cook. *"On the Allman Triangle and a Related Quadrilateral Element"*. *Computers and Structures*. Vol. 22. 1065-1067. 1986.

[115] R. H. MacNeal and R. L. Harder. *"A Refined For-Noded Membrane Element with Rotational Degrees of Freedom"*. *Computers and Structures*. Vol. 28, No. 1. 75-84.

[116] S. J. Garvey. *"The Quadrilateral Shear Panel"*. *Aircraft Engineering*. p. 134. May 1951.

[117] Shah M. Yunus, Timothy P. Pawlak, and R. D. Cook. *"Solid Elements with Rotational Degrees of Freedom Part 1 and Part 2"*. *International Journal for Numerical Methods in Engineering*. Vol. 31. 573-610. 1991.

[118] O. A. Mohammed. *"Magnetic Vector Potential Based Formulation and Computation of Nonlinear Magnetostatic Fields and Forces in Electrical Devices by Finite Elements"*. *Ph.D. Dissertation*. Virginia Polytechnic Institute and State University. Blacksburg, VA . May 1983.

[119] I. D. Mayergoyz. *"A New Scalar Potential Formulation for Three-Dimensional Magnetostatic Problems"*. *IEEE Transactions on Magnetics*. Vol. MAG-23, No. 6. 3889-3894. 1987.

[120] Oszkar Biro and Kurt Preis. *"On the Use of the Magnetic Vector Potential in the Finite Element Analysis of Three-Dimensional Eddy Currents"*. *IEEE Transactions on Magnetics*. Vol. 25, No. 4. 3145-3159 . July 1989.

[121] J. Robinson. *Basic and Shape Sensivity Tests for Membrane and Plate Bending Finite Elements*. Robinson and Associates. January 1985.

[122] Y. Kagawa, T. Yamabuchi, and S. Kitagami. *"Infinite Boundary Element and its Application to a Combined Finite-Boundary Element Technique for Unbounded Field Problems"*. *Boundary Elements VIII*. ed. C. A. Brebbia. Springer-Verlag,. New York, NY. 1986.

[123] J. T. Oden and N. Kikuchi. *"Finite Element Methods for Constrained Problems in Elasticity"*. *International Journal for Numerical Methods in Engineering*. Vol. 18, No. 5. 701-725. 1982.

[124] T. Sussman and K. J. Bathe. *"A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis"*. *Computers and Structures*. Vol. 26, No. 1/2. 357-409. 1987.

[125] O. C. Zienkiewicz, Y. C. Liu, and G. C. Huang. *"Error Estimates and Convergence Rates for Various Incompressible Elements"*. *International Journal for Numerical Methods in Engineering*. Vol. 28, No. 9. 2191-2202. 1989.

[126] H. C. Huang and R. W. Lewis. *"Adaptive Analysis for Heat Flow Problems Using Error Estimation Techniques"*. Paper presented at the 6th International Conference on Numerical Methods in Thermal Problems. Also University of Wales, University College of Swansea Internal Report CR/635/89 April 1989.

[127] G. G. Weber, A. M. Lush, A. Zavaliangos, and L. Anand. *"An Objective Time-Integration Procedure for Isotropic Rate-Independent Elastic-Plastic Constitutive Equations"*. *International Journal of Plasticity*. Vol. 6. 701-749. 1990.

[128] G. M. Eggert and P. R. Dawson. *"A Viscoplastic Formulation with Plasticity for Transient Metal Forming"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 70. 165-190. 1988.

[129] R. Narayanaswami and H. M. Adelman. *"Inclusion of Transverse Shear Deformation in Finite Element Displacement Formulations"*. *American Institute of Aeronautics and Astronautics Journal*. Vol. 12, No. 11. 1613-1614. 1974.

[130] I. Kaljevic, S. Saigal, and A. Ali. *"An Infinite Boundary Element Formulation for Three-Dimensional Potential Problems"*. *International Journal for Numerical Methods in Engineering*. Vol. 35, No. 10. 2079-2100. 1992.

[131] Simo .. *"Finite Deformation Post-Buckling Analysis Involving Inelasticity and Contact Constraints"*. *International Journal for Numerical Methods in Engineering*. Vol. 23. 779-800. 1986.

[132] H. Parisch. *"A Consistent Tangent Stiffness Matrix for Three-Dimensional Non-Linear Contact Analysis"*. *International Journal for Numerical Methods in Engineering*. Vol.28. 1803-1812. 1989.

[133] Bayo. *"A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 71. 183-195. 1988.

[134] Jiang and Rodgers. *"Combined Lagrangian Multiplier and Penalty Function Finite Element Technique for Elastic Impact Analysis"*. *Computers and Structures*. Vol. 30. 1219-1229. 1988.

[135] Giannakopoulos. *"The Return Mapping Method for the Integration of Friction Constitutive Relations"*. *Computers and Structures*. Vol. 32. 157-167. 1989.

[136] Ridic and Owen. *"A Plasticity Theory of Friction and Joint Elements"*. *Computational Plasticity: Models, Software, and Applications*. Part II. Proceedings of the Second International Conference, Barcelona, Spain, Pineridge Press, Swanse. 1043-1062. Editors Owen, Hinton, Ornate. 1989.

[137] Wriggers, VuVan, and Stein. *"Finite Element Formulation of Large Deformation Impact-Contact Problems with Friction"*. *Computers and Structures*. Vol. 37. 319-331. 1990.

[138] Stein, Wriggers and VuVan. *"Models of Friction, Finite-Element Implementation and Application to Large Deformation Impact-Contact Problems"*. *Computational Plasticity: Models, Software, and Applications*. Part II. Proceedings of the Second International Conference, Barcelona, Spain, Pineridge Press, Swansea. 1015-1041. Editors Owen, Hinton, Ornate. 1989.

[139] S. M. Yunus, P. C. Kohnke, and S. Saigal. *"An Efficient Through-Thickness Integration Scheme in an Unlimited Layer Doubly Curved Isoparametric Composite Shell Element"*. *International Journal for Numerical Methods in Engineering*. Vol. 28. 2777-2793. 1989.

[140] E. R. Geddes. *"An Analysis of the Low Frequency Sound Field in Non-Rectangular Enclosures Using the Finite Element Method"*. Ph.D Thesis, Pennsylvania State University. 1982.

[141] M. Gyimesi, D. Lavers, T. Pawlak, and D. Ostergaard. *"Application of the General Potential Formulation in the ANSYS®Program"*. *IEEE Transactions on Magnetics*. Vol. 29. 1345-1347. 1993.

[142] C. Rajakumar and A. Ali. *"A Solution Method for Acoustic Boundary Element Eigenproblem With Sound Absorption Using Lanczos Algorithm"*. Proceedings of 2nd International Congress on Recent Developments in Air- and Structure-Borne Sound and Vibration. Auburn University, AL. 1001-1010 . March 4-6, 1992.

[143] H. Nishimura, M. Isobe, and K. Horikawa. *"Higher Order Solutions of the Stokes and the Cnoidal Waves"*. *Journal of the Faculty of Engineering*. Vol. XXXIV, No. 2. The University of Tokyo. Footnote on page 268. 1977.

[144] G. Mahinthakumar and S.R.H. Hoole. *"A Parallelized Element by Element Jacobi Conjugate Gradients Algorithm for Field Problems and a Comparison with Other Schemes"*. *Applied Electromagnetics in Materials*. Vol. 1. 15-28. 1990.

[145] T.J.R. Hughes. *"Analysis of Transient Algorithms with Particular Reference to Stability Behavior"*. *Computation Methods for Transient Analysis*. Vol. 1. edited by T. Belytschko and K. J. Bathe. North-Holland, Amsterdam. 67-155. 1983.

[146] L. Anand. *"Constitutive Equations for the Rate-Dependent Deformation of Metals at Elevated Temperatures"*. *Journal of Engineering Materials and Technology*. Vol. 104. 12-17. 1982.

[147] S. B. Brown, K. H. Kim, and L. Anand. *"An Internal Variable Constitutive Model for Hot Working of Metals"*. *International Journal of Plasticity*. Vol. 5. 95-130. 1989.

[148] John M. Dickens. *"Numerical Methods for Dynamic Substructure Analysis"*. PH.D. Thesis from University of California, Berkeley. 1980.

[149] M. Gyimesi and J. D. Lavers. *"Generalized Potential Formulation for 3-D Magnetostatic Problems"*. *IEEE Transactions on Magnetics*. Vol. 28, No. 4. 1992.

[150] W. R. Smythe. *Static and Dynamic Electricity*. McGraw-Hill Book Co.. New York, NY. 1950.

[151] N. A. Demerdash, T. W. Nehl, F. A. Fouad, and O. A. Mohammed. *"Three Dimensional Finite Element Vector Potential Formulation of Magnetic Fields in Electrical Apparatus"*. *IEEE Transactions on Power Apparatus and Systems*. Vol. PAS-100, No. 8. 4104-4111. 1981.

[152] G. M. Eggert, P. R. Dawson, and K. K. Mathur. *"An Adaptive Descent Method for Nonlinear Viscoplasticity*. *International Journal for Numerical Methods in Engineering*. Vol. 31. 1031-1054. 1991.

[153] K. H. Schweizerhof and P. Wriggers. *"Consistent Linearization for Path Following Methods in Nonlinear FE Analysis"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 59. 261-279. 1986.

[154] O. C. Zienkiewicz and I. C. Cormeau. *"Visco-plasticity - Plasticity and Creep in Elastic Solids - A Unified Numerical Solution Approach"*. *International Journal for Numerical Methods in Engineering*. Vol. 8. 821-845. 1974.

[155] J. C. Simo and R. L. Taylor. *"Consistent Tangent Operators for Rate-Independent Elastoplasticity"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 48. 101-118. 1985.

[156] T. J. R. Hughes. *"Numerical Implementation of Constitutive Models: Rate-Independent Deviatoric Plasticity"*. *Theoretical Foundation for Large-Scale Computations for Nonlinear Material Behavior*. edited by S. Nemat-Nasser, R. J. Asaro, and G. A. Hegemier. Martinus Nijhoff Publishers. Dordrecht, The Netherlands . 1984.

[157] T. J. R. Hughes and E. Carnoy. *"Nonlinear Finite Element Shell Formulation Accounting for Large Membrane Strains"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 39. 69-82 . 1983.

[158] J. Nedelec. *"Mixed finite elements in R3"*. Numer. Math., Vol.35. 315-341. 1980.

[159] L. Anand. *"Constitutive Equations for Hot-Working of Metals"*. *International Journal of Plasticity*. Vol. 1. 213-231. 1985.

[160] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards Applied Mathematics Series 55. p. 966. 1972.

[161] C. G. Swain and M. S. Swain. *"A Uniform Random Number Generator That is Reproducible, Hardware-Independent, and Fast"*. *Journal of Chemical Information and Computer Sciences*. 56-58. 1980.

[162] Edwin Kreyszig. *Advanced Engineering Mathematics*. 3rd Edition. John Wiley & Sons, Inc.. 1972.

[163] Paul G. Hoel. *Introduction to Mathematical Statistics*. 3rd Edition. Johnn Wiley & Sons, Inc.. p. 196. 1962.

[164] Neter, et al. *Applied Statistics*. Allyn and Bacon, Inc.. Boston, MA. 1978.

[165] T. J. R. Hughes. *The Finite Element Method Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Inc.. Englewood Cliffs, NJ . 1987.

[166] E. L. Wilson and Tetsuji Itoh. *"An Eigensolution Strategy for Large Systems"*. *Computers and Structures*. Vol. 16, No. 1-4. 259-265. 1983.

[167] T. Yokoyama. *"Vibrations of a Hanging Timoshenko Beam Under Gravity"*. *Journal of Sound and Vibration*. Vol. 141, No. 2. 245-258. 1990.

[168] J. L. Coulomb. *"A Methodology for the Determination of Global Electromechanical Quantities from a Finite Element Analysis and its Application to the Evaluation of Magnetic Forces, Torques and Stiffness"*. *IEEE Transactions on Magnetics*. Vol. MAG-19, No. 6. 1983. 2514-2519.

[169] O. C. Zienkiewicz, C. Emson, and P. Bettess. *"A Novel Boundary Infinite Element"*. *International Journal for Numerical Methods in Engineering*. Vol. 19. 393-404. 1983.

[170] F. Damjanic and D. R. J. Owen. *"Mapped Infinite Elements in Transient Thermal Analysis"*. *Computers and Structures*. Vol. 19, No. 4. 673-687. 1984.

[171] J. M. M. C. Marques and D. R. J. Owen. *"Infinite Elements in Quasi-Static Materially Nonlinear Problems"*. *Computers and Structures*. Vol. 18, No. 4. 739-751. 1984.

[172] Hui Li, Sunil Saigal, Ashraf Ali, and Timothy P. Pawlak. *"Mapped Infinite Elements for 3-D Vector Potential Magnetic Problems"*. *International Journal for Numerical Methods in Engineering*. Vol. 37. 343-356. 1994.

[173] M. Gyimesi, J. Lavers, T. Pawlak, and D. Ostergaard. *"Biot-Savart Integration for Bars and Arcs"*. *IEEE Transactions on Magnetics*. Vol. 29, No. 6. 2389-2391. 1993.

[174] M.A. Crisfield. *"A fast and incremental/iterative solution procedure that handles snap-through"*. *Computers & Structures*. Vol. 13. 55–62. 1981.

[175] B. Nour-Omid and C. C. Rankin. *"Finite Rotation Analysis and Consistent Linearization Using Projectors"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 93. 353-384 . 1991.

[176] C.R.I. Emson and J. Simkin. *"An Optimal Method for 3-D Eddy Currents"*. *IEEE Transactions on Magnetics*. Vol. MAG-19, No. 6. 2450-2452. 1983.

[177] P.L. Viollet. *"The Modelling of Turbulent Recirculating Flows for the Purpose of Reactor Thermal-Hydraulic Analysis"*. *Nuclear Engineering and Design*. 99. 365-377. 1987.

[178] B.E. Launder and D.B. Spalding. *"The Numerical Computation of Turbulent Flows"*. *Computer Methods In Applied Mechanics and Engineering*. Vol. 3. pp 269-289. 1974.

[179] J.G. Rice and R.J. Schnipke. *"A Monotone Streamline Upwind Finite Element Method for Convection-Dominated Flows"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 48. pp.313-327. 1985.

[180] F.H. Harlow and A.A. Amsden. *"A Numerical Fluid Dynamics Calculation Method for All Flow Speeds"*. *Journal of Computational Physics*. Vol 8. 1971.

[181] F.M. White. *Viscous Fluid Flow*. Second Edition. McGraw-Hill. New York. 1991.

[182] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere, New York. 1980.

[183] Magnus R. Hestenes and Eduard Stiefel. *"Methods of Conjugate Gradients for Solving Linear System"*. *Journal of Research of the National Bureau of Standards*. Vol. 49, No.6. 1952.

[184] J.K. Reid. *"On the Method of Conjugate Gradients for the Solution of Large Sparse Sets of linear Euqations"*. *Proceedings of the Conference on Large Sparse Sets of Linear Equations*. edited by J.K. Reid. Academic Press. 231-254. 1971.

[185] H.C. Elman. *"Preconditioned Conjugate-Gradient Methods for Nonsymmetric Systems of Linear Equations"*. *Advances In Computer Methods For Partial Differential Equations IV*. edited by Vichnevetsky, R., Stepleman. IMACS. 409-413. 1981.

[186] J.J. More and S.J. Wright. *Optimization Software Guide*. SIAM. Philadelphia. p. 13. 1993.

[187] R.W. Bilger. *"A Note on Favre Averaging in Variable Density Flows"*. *Combustion Science and Technology*. Vol. 11. 215-217. 1975.

[188] M. C. McCalla. *Fundamentals of Computer-Aided Circuit Simulation*. Kluwer Academic. 1988.

[189] P.A. Vermeer and A. Verrujit. *"An Accuracy Condition for Consolidation by Finite Elements"*. *International Journal for Numerical and Analytical Methods in Geomechanics*. Vol. 5. 1-14. 1981.

[190] Stephen W. Tsai and H. Thomas Hahn. *Introduction to Composite Materials*. Section 7.2. Technomic Publishing Company. 1980.

[191] G.E.P. Box, W.G. Hunter, and J.S. Hunter. *Statistics for Experimenters*. Chapter 10. John Wiley & Sons. 1978.

[192] Barna Szabo and Ivo Babuska. *Finite Element Analysis*. John Wiley & Sons. 1991.

[193] M.T. Chen. A. Ali. *"An Efficient and Robust Integration Technique for Applied Random Vibration Analysis"*. *Computers and Structures*. Vol. 66 No. 6. 785–798. 1998.

[194] R.S. Harichandran. *Random Vibration Under Propagating Excitation: Closed-Form Solutions*. *Journal of Engineering Mechanics ASCE*. Vol. 118, No. 3. 575-586. 1992.

[195] R.G. Grimes, J.G. Lewis, and H.D. Simon. *"A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems"*. *SIAM Journal Matrix Analysis Applications*. Vol. 15 No. 1. 228-272 . 1996.

[196] C. Rajakumar and C.R. Rogers. *"The Lanczos Algorithm Applied to Unsymmetric Generalized Eigenvalue Problems"*. *International Journal for Numercial Method in Engineering*. Vol. 32. 1009-1026. 1991.

[197] D.K. Gartling. *"Finite Element Methods for Non-Newtonian Flows"*. *report SAND92-0886, CFD Dept.,*. Sandia National Laboratories. Albuquerque, NM. 1992.

[198] M.J. Crochet, A.R. Davies, and K. Walters. *Numerical Simulation of Non-Newtonian Flow*. Elsevier Science Publishers B.V.. 1984.

[199] John O. Hallquist. *LS-DYNA Theoretical Manual*. Livermore Software Technology Corporation. 1998.

[200] O. Biro, K. Preis, C. Magele, W. Renhart, K.R. Richter, and G. Vrist. *"Numerical Analysis of 3D Magnetostatic Fields"*. *IEEE Transaction on Magnetics*. Vol. 27, No. 5. 3798-3803. 1991.

[201] M. Gyimesi and D Ostergaard. *"Non-Conforming Hexahedral Edge Elements for Magnetic Analysis"*. ANSYS, Inc. internal development, submitted to COMPUMAG. Rio. 1997.

[202] Gyimesi, M. and Lavers, D., "Application of General Potential Formulation to Finite Elements", Second Japan Hungarian Joint Seminar on Electromagnetics, Sapporo, Japan 1992. Applied Electromagnetics in Materials and Computational Technology, ed. T. Honma, I. Sebestyen, T. Shibata. Hokkaido University Press. 1992.

[203] K. Preis, I. Bardi, O. Biro, C. Magele, Vrisk G., and K. R. Richter. *"Different Finite Element Formulations of 3-D Magnetostatic Fields"*. *IEEE Transactions on Magnetics*. Vol. 28, No. 2. 1056-1059. 1992.

[204] J.C. Nedelec. *"Mixed Finite Elements in R3"*. *Numerical Methods*. Vol. 35. 315-341. 1980.

[205] J.S. Van Welij. *"Calculation of Eddy Currents in Terms of H on Hexahedra"*. *IEEE Transactions on Magnetics*. Vol. 18. 431-435. 1982.

[206] A. Kameari. *"Calculation of Transient 3D Eddy Current Using Edge Elements"*. *IEEE Transactions on Magnetics*. Vol. 26. 466-469. 1990.

[207] J. Jin. *The Finite Element Method in Electromagnetics*. John Wiley and Sons, Inc.. New York. 1993.

[208] H. Whitney. *Geometric Integration Theory*. Princeton U. P.. Princeton. 1957.

[209] J.A. Stratton. *Electromagnetic Theory*. Section 1.14. McGraw-Hill. New York. 1941.

[210] K.M. Mitzner. *"An Integral Equation Approach to Scattering From a Body of Finite Conductivity"*. *Radio Science*. Vol. 2. 1459-1470. 1967.

[211] R. Mittra and O. Ramahi. *"Absorbing Boundary Conditions for the Direct Solution of Partial Differential Equations Arising in Electromagnetic Scattering Problems"*. *Finite Element Finite Difference Methods in Electromagnetic Scattering*. Vol. II. 133-173. 1989.

[212] D. Peric and D.R.J. Owen. *"Computational Model for 3-D Contact Problems with Friction Based on the Penalty Method"*. *International Journal for Numercial Method in Engineering*. Vol. 35. 1289-1309. 1992.

[213] S. Cescotto and R. Charilier. *"Frictional Contact Finite Elements Based on Mixed Variational Principles"*. *International Journal for Numercial Method in Engineering*. Vol. 36. 1681-1701. 1992.

[214] S. Cescotto and Y.Y. Zhu. *"Large Strain Dynamic Analysis Using Solid and Contact Finite Elements Based on a Mixed Formulation - Application to Metalforming"*. *Journal of Metals Processing Technology*. Vol. 45. 657-663. 1994.

[215] J.C. Simo and T.A. Laursen. *"An Augmented Lagrangian Treatment of Contact Problems Involving Friction"*. *Computers and Structures*. Vol. 42, No. 1. 97-116. 1992.

[216] T.A. Laursen and J.C. Simo. *"Algorithmic Symmetrization of Coulomb Frictional Problems Using Augmented Lagrangians"*. *Computers Methods in Applied Mechanics and Engineering*. Vol. 108, No. 1 & 2. 133-146. 1993.

[217] A. Barry, J. Bielak, and R.C. MacCamy. *"On absorbing boundary conditions for wave propagations"*. *Journal of Computational Physics*. Vol. 792. 449-468 . 1988.

[218] L.F. Kallivokas, J. Bielak, and R.C. MacCamy. *"Symmetric Local Absorbing Boundaries in Time and Space"*. *Journal of Engineering Mechanics*. Vol. 1179. 2027-2048. 1991.

[219] T.J.R. Hughes. *"Generalization of Selective Integration Procedures to Anisotropic and Nonlinear Media"*. *International Journal for Numerical Methods in Engineering*. Vol. 15, No. 9. 1413-1418. 1980.

[220] J.C. Nagtegaal, D.M. Parks, and J.R. Rice. *"On Numerically Accurate Finite Element Solutions in the Fully Plastic Range"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 4. 153-178. 1974.

[221] Miklos Gyimesi and Dale Ostergaard. *"Mixed Shape Non-Conforming Edge Elements"*. *CEFC '98*. Tucson, AZ. 1998.

[222] Dale Ostergaard and Miklos Gyimesi. *"Analysis of Benchmark Problem TEAM20 with Various Formulations"*. Proceedings of the TEAM Workshop, COMPUMAG Rio. 18-20 . 1997.

[223] Dale Ostergaard and Miklos Gyimesi. *"Magnetic Corner: Accurate Force Computations"*. *Analysis Solutions*. Vol 1, Issue 2. 10-11. 1997-98.

[224] A.N. Brooks and T.J.R. Hughes. *"Streamline Upwind/Petro-Galkerin Formulation for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 32. 199-259 . 1982.

[225] N.A. Demerdash and A.A. Arkadan. *"Notes on FEM Modeling of Permanent Magnets in Electrical Devices"*. *FEM for Electromagnetic Applications*. Section 3. p.26-7, 17, 19. 1981.

[226] N.A. Demerdash and T.W. Nehl. *"Determination of Inductances in Ferrite Type Magnet Electric Machinery by FEM"*. *IEEE Trans. on MAG*. Vol.18. pp.1052-54. 1982.

[227] T.W. Nehl, F.A. Faud, and N.A. Demerdash. *"Determination of Saturated Values of Rotation Machinery Incremental and Apparent Inductances by an Energy Perturbation Method"*. *IEEE Trans. on PAS*. Vol.101. pp.4441-51 . 1982.

[228] Miklos Gyimesi, Vladimir Zhulin, and Dale Ostergaard. *"Particle Trajectory Tracing in ANSYS"*. Fifth International Conference on Charged Particle Optics, Delft University, Netherlands. To be Published in Nuclear Instruments and Methods in Physics Research, Section A. 1998.

[229] Miklos Gyimesi and Ostergaard. *Dale*. *"Inductance Computation by Incremental Finite Element Analysis"*. CEFC 98. Tucson, Arizona. 1998.

[230] *"Computer-Aided Generation of Nonlinear Reduced-Order Dynamic Macromodels - I: Non-Stress-Stiffened Case"*. *Journal of Microelectromechanical Systems*. S. 262–269. June 2000.

[231] N.A. Demerdash and D.H. Gillott. *"A New Approach for Determination of Eddy Currents and Flux Penetration in Nonlinear Ferromagnetic Materials"*. *IEEE Trans. on Magnetics*. Vol. 10. 682-685. 1974.

[232] D.P. Flanagan and T. Belytschko. *"A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control"*. *International Journal for Numerical Methods in Engineering*. Vol. 17. 679-706. 1981.

[233] F. Vogel. *"Topological Optimization of Linear-Elastic Structures with ANSYS 5.4."*. NAFEMS Conference on Topological Optimization. 1997.

[234] H.P. Mlejnek and R. Schirrmacher. *"An Engineer's Approach to Optimal Material Distribution and Shape Finding"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 106. 1-26. 1993.

[235] M.P. Bendsoe and N. Kikucki. *"Generating Optimal Topologies in Structural Design Using a Homogenization Method"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 71. 197-224. 1988.

[236] Javier Bonet and Richard D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press. 1997..

[237] J.C. Simo and L. Vu-Quoc. *"A Three Dimensional Finite Strain Rod Model. Part II: Computational Aspects"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 58. 79-116. 1986.

[238] Adnan Ibrahimbegovic. *"On Finite Element Implementation of Geometrically Nonlinear Reissner's Beam Theory: Three-dimensional Curved Beam Elements"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 122. 11-26. 1995.

[239] Istvan Vago and Miklos Gyimesi. *Electromagnetic Fields*. Published by Akademiiai Kiado. Budapest, Hungary. 1998.

[240] S. Flugge. *"Electric Fields and Waves"*. *Encyclopedia of Physics*. Vol. 16. Springer, Berlin . 1958.

[241] M. Lagally. *Vorlesungen uber Vektorrechnung*. Geest u. Portig, Peipzip. 1964.

[242] D.P. Flanagan and T. Belytschko. *"A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control"*. *International Journal for Numerical Methods in Engineering*. Vol. 17. 679-706. 1981.

[243] H.B. Calllen. *Thermodynamics and Introduction to Thermostatistics*. 2nd Edition. p. 84. Wiley & Sons. New York, NY. 1985.

[244] J.L. Chaboche. *"Constitutive Equations for Cyclic Plasticity and Cyclic Viscoplasticity"*. *International Journal of Plasticity*. Vol. 5. 247-302. 1989.

[245] J.L. Chaboche. *"On Some Modifications of Kinematic Hardening to Improve the Description of Ratcheting Effects"*. *International Journal of Plasticity*. Vol. 7. 661-678. 1991.

[246] Timoshenko. *Theory of Elastic Stability*. McGraw Hill Book Company. 1961.

[247] M. Schulz and F. C. Fillippou. *"Generalized Warping Torsion Formulation"*. *Journal of Engineering Mechanics*. 339-347. 1998.

[248] M. Gyimesi and D. Ostergaard. *"Electro-Mechanical Capacitor Element for MEMS Analysis in ANSYS"*. *Proceedings of Modelling and Simulation of Microsystems Conference*. 270 . Puerto Rico. 1999.

[249] M. Schulz and F. C. Fillippou. *"Capacitance Computation with Ammeter Element"*. University of Toronto, Department of Electrical Engineering, Unpublished Report available upon request from ANSYS, Inc.. , 1993.

[250] J. Mehner and S.D. Senturia. *"Computer-Aided Generation of Nonlinear Reduced-Order Dynamic Macromodels - II: Stress-Stiffened Case"*. *Journal of Microelectromechanical Systems,*. S. 270–279. June 2000.

[251] Hieke, A., Siemens and IBM. *"ANSYS APDL for Capacitance"*. Proceedings from `Second International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators'. 172. San Juan, Puerto Rico. 1999.

[252] J.C. Simo and T.J.R. Hughes. *Computational Inelasticity*. Springer-Verlag. 1997.

[253] E. Voce. *Metallurgica*. Col. 51, 219 . 1955.

[254] W.H. Press. *Numerical Recipes in C: The Art of Scienfitic Computing*. Cambridge University Press. 1993.

[255] M. Gyimesi, D. Lavers, D Ostergaard, and T. Pawlak. *"Hybrid Finite Element - Trefftz Method for Open Boundary Anslysis"*. COMPUMAG, Berlin 1995, IEEE Transactions on Magnetics, Vol. 32, No. 3, 671-674 1996.

[256] M. Gyimesi and D. Lavers. *"Application of the Trefftz Method to Exterior Problems"*. University of Toronto, Department of Electrical Engineering, unpublished report. Available upon request from ANSYS, Inc.. 1992.

[257] M. Gyimesi and D. Lavers. *"Application of the Trefftz Method to Exterior Problems"*. University of Toronto, Department of Electrical Engineering, unpublished report. Available upon request from ANSYS, Inc.. 1993.

[258] M. Gyimesi and D. Lavers. *"Implementation to the Exterior Trefftz Element"*. University of Toronto, Department of Electrical Engineering, unpublished report. Available upon request from ANSYS, Inc.. , 1993.

[259] E. Trefftz. *"Ein Gegenstuck zum Ritz'schen Verfahren"*. Proceedings of the Second International Congress on Applied Mechanics. Zurich. 1926.

[260] E. Trefftz. *"Mechanik det elastischen Korper"*. In Vol. VI of Handbuch der Physik, Berlin 1928. Translated from Matematicheskais teoriia Uprognosti, L. GTTI 1934.

[261] I. Herrera. *"Trefftz Method"*. in progress, Boundary Element Methods, Vol. 3. Wiley, New York . 1983.

[262] O.C. Zienkiewicz. *"The Generalized Finite Element Method and Electromagnetic Problems"*. COMPUMAG Conference. 1978.

[263] A.P. Zielinski and O.C. Zienkiewicz. *"Generalized Finite Element Analysis with T-Complete Boundary Solution Function"*. *International Journal for Numerical Methods in Engineering*. Vol. 21. 509-528. 1985.

[264] O.C. Zienkiewicz, D.W. Kelly, and P Bettess. *"The Coupling of the Finite Element Method and Boundary Solution Procedures"*. *International Journal for Numerical Methods in Engineering*. Vol. 11. 355-375. 1977.

[265] O.C. Zienkiewicz, D.W. Kelly, and P. Bettess. *"Marriage a la mode - The Best of Both Worlds Finite Element and Boundary Integrals"*. *Energy Methods in Finite Element Analysis*. John Wiley. New York. 1979.

[266] J. Jirousek and L. Guex. *"The Hybrid-Trefftz Finite Element Model and its Application to Plate Bending"*. *International Journal for Numerical Methods in Engineering*. Vol. 23. 651-693. 1986.

[267] I.D. Mayergoyz, M.V.C. Chari, and A. Konrad. *"Boundary Galerkin's Method for Three-Dimensional Finite Element Electromagnetic Field Computation"*. *IEEE Transactions on Magnetics*. Vol. 19, No. 6. 2333-2336. 1983.

[268] M.V.K. Chari. *"Electromagnetic Field Computation of Open Boundary Problems by Semi-Analytic Approach"*. *IEEE Transactions on Magnetics*. Vol. 23, No. 5. 3566-3568. 1987.

[269] M.V.K. Chari and G. Bedrosian. *"Hybrid Harmonic/Finite element Method for Two-Dimensional Open Boundary Problems"*. *IEEE Transactions on Magnetics*. Vol. 23, No. 5. 3572-3574. 1987.

[270] E.M. Arruda and M.C. Boyce. *"A Three-dimensional Constitutive Model for the Large STretch Behavior of Rubber Elastic Materials"*. *Journal of the Mechanics and Physics of Solids*. Vol. 41 2. 389-412. 1993.

[271] J.S. Bergstrom. *"Constitutive Modeling of the Large Strain Time-dependent Behavior of Elastomers"*. *Journal of the Mechanics and Physics of Solids*. Vol. 45 5. 931-954. 1998.

[272] M.W. Glass. *"Chaparral - A library package for solving large enclosure radiation heat transfer problems"*. Sandia National Laboratories. Albuquerque, NM. 1995.

[273] A.R. Diaz and N. Kikucki. *"Solutions to Shape and Topology Eigenvalue Optimization Problems using a Homogenization Method"*. *International Journal for Numerical Methods in Engineering*. Vol. 35. pp 1487-1502. 1992.

[274] P. Ladeveze and D. Leguillon. *"Error estimation procedure in the finite element method and applications"*. *SIAM Journal of Numerical Analysis*. Vol. 20 3. 483-509. 1983.

[275] J.L. Synge. *The Hypercircle in Mathematical Physics*. Cambridge University Press. 1957.

[276] M.F. Cohen and D.P. Greenberg. *"The Hemi-Cube: A Radiosity Solution for Complex Environments"*. *Computer Graphics*. Vol. 19, No. 3. 31-40. 1985.

[277] M.L. Williams, R.F. Landel, and J.D. Ferry. *"The Temperature Dependence of Relaxation Mechanisms in Amorphous Polymers and Other Glass-forming Liquids"*. *Journal of the American Chemical Society*. Vol. 77. 3701-3706. 1955.

[278] A. Huerta and W.K. Liu. *"Viscous Flow with Large Free Surface Motion"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 69. 277-324. 1988.

[279] W. Weaver and P.R. Johnston. *Structural Dynamics by Finite Elements*. 413-415. Prentice-Hall. 1987.

[280] Y.Y. Zhu and S. Cescotto. *"Transient Thermal and Thermomechanical Analysis by Mixed FEM"*. *Computers and Structures*. Vol. 53. 275-304. 1994.

[281] J.U. Brackbill, D.B. Kothe, and C. Zemach. *"A Continuum Method for Modeling Surface Tension"*. *Journal of Computational Physics*. Vol. 100. 335-354 . 1992.

[282] D.B. Kothe and R.C. Mjolsness. *"RIPPLE: A New Model for Incompressible Flows with Free Surfaces"*. *AIAA Journal*. Vol. 30. 2694-2700. 1992.

[283] J.R. Richards, A.M. Lenhoff, and A.N. Beris. *"Dynamic Breakup of Liquid-Liquid Jets"*. *Physics of Fluids*. Vol. 8. 2640-2655. 1994.

[284] G.P. Sasmal and J.I. Hochstein. *"Marangoni Convection with a Curved and Deforming Free Surface in a Cavity"*. *Transaction of ASME, Journal of Fluid Engineering*. Vol. 116. 577-582. 1994.

[285] G. Wang. *"Finite Element Simulations of Gas-Liquid Flows with Surface Tension"*. Presented at the 2000 International Mechanical Engineering Congress and Exposition. Orlando, FL. 11/2000.

[286] M. Gyimesi and D. Ostergaard. *"Finite Element Based Reduced Order Modeling of Micro Electro Mechanical Sytems MEMS"*. Presented at MSM 2000. San Diego, CA . 3/2000.

[287] D. Ostergaard, M. Gyimesi, Bachar Affour, Philippe Nachtergaele, and Stevan Stirkovich. *"Efficient Reduced Order Modeling for System Simulation of Micro Electro Mechanical Systems MEMS from FEM Models"*. Symposium on Design Test Integration and Packaging of MEMS/MOEMS. Paris, France. 5/2000.

[288] M. Gyimesi, Jian-She, Wang, and D. Ostergaard. *"Capacitance Computation by Hybrid P-Element and Trefftz Method"*. Presented at CEFC 2000, Milwaukee, WI 6/2000 and published in IEEE Trans. MAG, Vol. 37, 3680–83 9/2001.

[289] M. Gyimesi and D. Ostergaard. *"Capacitance Computation by Hybrid P-Element and Trefftz Method"*. Presented at MSM 2000. San Diego, CA . 3/2000.

[290] M. Gyimesi and D. Ostergaard. *"Incremental Magnetic Inductance Computation"*. ANSYS Conference and Exhibition. Pittsburgh, PA. 1998.

[291] Andreas Hieke. *"Tiny Devices, Big Problems: Computation of Capacitance in Microelectronic Structures"*. *ANSYS Solutions*. Vol. 2, No. 3. 11-15. 2000.

[292] M.S. Gadala and J. Wang. *"Simulation of Metal Forming Processes with Finite Element Methods"*. *International Journal for Numerical Methods in Engineering*. Vol. 44. 1397-1428. 1999.

[293] R.M. McMeeking and J.R. Rice. *"Finite Element Formulations for Problems of Large Elastic-Plastic Deformation"*. *International Journal of Solids and Structures*. Vol. 121. 601-616. 1975.

[294] M.A. Crisfield. *Non-linear Finite Element Analysis of Solids and Structures*. Vol. 2, Advanced Topics. John Wiley & Sons. 1997.

[295] R. W. Ogden. *Nonlinear Elastic Deformations*. Dover Publications, Inc.. 1984.

[296] P. Perzyna. *Fundamental problems in viscoplasticity*. *Advances in Applied Mechanics*. Vol. 9. 313-377. Academic Press. New York. 1968.

[297] D. Peirce, C.F. Shih, and A. Needleman. *"A tangent modulus method for rate dependent solids"*. *Computers & Structures*. Vol. 18. 975–888. 1984.

[298] D. Peirce and D.R.J. Owen. A model for large deformations of elasto-viscoplastic solids at finite strains: computational issues, Finite Inelastic Deformations: Theory and applications, Springer-Verlag, Berlin 1992.

[299] J.L. Volakis, A. Chatterjee, and C. Kempel L.. *Finite Element Method for Electromagnetics: Antennas, Microwave Circuits and Scattering Applications*. IEEE Press. 1998.

[300] T. Itoh, G. Pelosi, and P.P. Silvester. *Finite Element Software for Microwave Engineering*. John Wiley & Sons, Inc. 1996.

[301] L. Zhao and A.C. Cangellaris. *"GT-PML: Generalized Theory of Perfectly Matched Layers and Its Application to the Reflectionless Truncation of Finite-Difference Time-Domain Grids"*. *IEEE Trans. on Microwave Theory and Techniques*. Vol. 44. 2555-2563.

[302] Alan George and Joseph W-H Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, Inc.. 1981.

[303] M. Abramowitz and I. A. Stegun. *Pocketbook of Mathematical Functions, abridged version of the Handbook of Mathematical Functions*. Harry Deutsch, 1984.

[304] A. H-S Ang and W. H. Tang. *Probability Concepts in Engineering Planning and Design*. Volume 1 - Basic Principles. John Wiley & Sons. 1975.

[305] A. H-S Ang and W. H. Tang. *Probability Concepts in Engineering Planning and Design, D.*. Volume 2 - Decision, Risk, and Reliability. John Wiley & Sons. 1990.

[306] G. E. P. Box. D. W. Behnken. *Some New Three Level Designs for the Study of Quantitative Variables*. Vol. 2, No. 4. 455-476. Technometrics. 1960.

[307] G. E. P. Box and D. R. Cox. *"An Analysis of Transformations"*. *Journal of the Royal Statistical Society*. Series B, Vol. 26. 211-252. 1964.

[308] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods,*. John Wiley & Sons. New York. 1964.

[309] R.L. Iman and W. J. Conover. *"Small Sample Sensitivity Analysis Techniques for Computer Models, with an Application to Risk Assessment"*. *Communications in Statistics, Part A - Theory and Methods*. Vol A9, No. 17. 1749-1842. 1980.

[310] D. Kececioglu. *Reliability Engineering Handbook*. Vol. 1. Prentice-Hall Inc.. Englewood Cliffs, New Jersey. 1991.

[311] P.-L. Liu and A. Der Kiureghian. *"Multivariate Distribution Models with Prescribed Marginals and Covariances"*. *Probabilistic Engineering Mechanics*. Vol. 1, No. 2. 105-112. 1986.

[312] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons. New York. 1991.

[313] R. C. Myers. *Response Surface Methodology*. Allyn and Bacon, Inc.. Boston. 1971.

[314] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman. *Applied Linear Statistical Models*. 4th edition. McGraw-Hill. 1996.

[315] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press Inc.. Florida. 1997.

[316] D.A. Hancq, A.J. Walter, and J.L. Beuth. *"Development of an Object-Oriented Fatigue Tool"*. *Engineering with Computers*. Vol. 16. 131-144. 2000.

[317] David J. Benson and John O. Hallquist. *"A Single Surface Contact Algorithm for the Post-Buckling Analysis of Shell Structures"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 78, No. 2. 1990.

[318] J.C. Simo and M.S. Rifai. *"A Class of Mixed Assumed Strain Methods and the Method of Incompatible Modes"*. *International Journal for Numerical Methods in Engineering*. Vol. 29. 1595–1638 . 1990.

[319] J.C. Simo and F. Armero. *"Gometrically Non-linear Enhanced Strain Mixed Methods and the Method of Incompatible Modes"*. *International Journal for Numerical Methods in Engineering*. Vol. 33. 1413–1449. 1992.

[320] J.C. Simo, F. Armero, and R.L. Taylor. *"Improved Versions of Assumed Enhanced Strain Tri-Linear Elements for 3D Finite Deformation Problems"*. *Computer Methods in Applied Mechanics and Engieering*. Vol. 10. 359–386. 1993.

[321] U. Andelfinger and E Ramm. *"EAS-Elements for Two-Dimensional, Three-Dimensional, Plate and Shell Structures and Their Equivalence to HR-Elements"*. *International Journal for Numerical Methods in Engineering*. Vol. 36. 1311–1337. 1993.

[322] J.C. Nagtegaal and D.D. Fox. *"Using Assumed Enhanced Strain Elements for Large Compressive Deformation"*. *International Journal for Solids and Structures*. Vol. 33. 3151–3159. 1996.

[323] Jian S. Wang and Dale F. Ostergaard. *"Finite Element-Electric Circuit Coupled Simulation Method for Piezoelectric Transducer"*. *Proceedings of the IEEE Ultrasonics Symposium*. Vol. 2. 1105–1108. 1999.

[324] A.C. Pipkin. *"Lectures in Viscoelasticity Theory"*. Springer, New York. 1986.

[325] D.A. Drozdov. *Finite elasticity and viscoelasticity: A course in the nonlinear mechanics of solids*. World Pub. Co.. Singapore. 1996.

[326] G.W. Scherer. *Relaxation in glass and composites*. John-Wiley & Sons. New York. 1986.

[327] J.C. Simo. *"On fully three-dimensional finite strain viscoelastic damage model: Formulation and computational aspects"*. *Comput. Meth. In Appl. Mech. Eng.*. Vol. 60. 153-173. 1987.

[328] G.A. Holzapfel. *"On large strain viscoelasticity: continuum formulation and finite element applications to elastomeric structures"*. *Int. J. Numer. Meth. Eng.*. Vol. 39. 3903-3926. 1996.

[329] M. Gyimesi, D. Ostergaard, and I. Avdeev. *"Triangle Transducer for Micro Electro Mechanical Systems MEMS Simulation in ANSYS Finite Element Program"*. *MSM*. Puerto Rico. 2002.

[330] M. Gyimesi and D. Ostergaard. *"A Transducer Finite Element for Dynamic Coupled Electrostatic-Structural Coupling Simulation of MEMS Devices"*. MIT Conference. Cambridge, MA. 2001.

[331] I. Avdeev, M. Gyimesi, M. Lovell, and D. Onipede. *"Beam Modeling for Simulation of Electro Mechanical Transducers Using Strong Coupling Approach"*. Sixth US. National Congress on Computational Mechanics. Dearborn, Michigan . 2001.

[332] W. F. Chen and D. J. Han. *Plasticity for Structural Engineers*. Springer-Verlag. New York. 1988.

[333] P. Guillaume. *"Derivees d'ordre superieur en conception optimale de forme"*. These de l'universite Paul Sabatier de Toulouse. 1994.

[334] H. E. Hjelm. *"Yield Surface for Gray Cast iron under Biaxial Stress"*. *Journal of Engineering Materials and Technology*. Vol. 116. 148-154. 1994.

[335] J. Mehner, F. Bennini, and W. Dotzel. *"Computational Methods for Reduced Order Modeling of Coupled Domain Simulations"*. 260–263. 11th International Conference on Solid-State Sensors and Actutors Transducers 01. Munich, Germany. 2001.

[336] J. Mehner and F. Bennini. *"A Modal Decomposition Technique for Fast Harmonic and Transient Simulations of MEMS"*. 477–484. International MEMS Workshop 2001 IMEMS. Singapore. 2001.

[337] J. J. Blech. *"On Isothermal Squeeze Films"*. *Journal of Lubrication Technology*. Vol.105. 615-620. 1983.

[338] Griffin, W. S., et al. *"A Study of Squeeze-film Damping"*. *Journal of Basic Engineering*. 451-456. 1966.

[339] W. E. Langlois. *"Isothermal Squeeze Films"*. *Quarterly Applied Mathematics*. Vol. 20, No. 2. 131-150. 1962.

[340] Mehner, J. E., et al. *"Simulation of Gas Film Damping on Microstructures with Nontrivial Geometries"*. Proc. of the MEMS Conference. Heidelberg, Germany. 1998.

[341] not used.

[342] T. Veijola. *"Equivalent Circuit Models for Micromechanical Inertial Sensors"*. *ircuit Theory Laboratory Report Series CT-39*. Helsinki University of Technology. 1999.

[343] F. Sharipov. *"Rarefied Gas Flow Through a Long Rectangular Channel"*. *Journal Vac. Sci. Technol.*. A175. 3062-3066. 1999.

[344] R. R. Craig. *"A Review of Time Domain and Frequency Domain Component Mode Synthesis Methods"*. *International Journal of Analytical and Experimental Modal Analysis*. Vol. , No. 2. 59-72. 1987.

[345] R. R. Craig and M. D. D. Bampton. *"Coupling of Substructures for Dynamic Analysis"*. *AIAA Journal*. Vol. 12. 1313-1319. 1968.

[346] M. Gyimesi, I. Avdeev, and D. Ostergaard. *"Finite Element Simulation of Micro Electro Mechanical Systems MEMS by Strongly Coupled Electro Mechanical Transducers"*. *IEEE Transactions on Magnetics*. Vol. 40, No. 2. pg. 557–560. 2004.

[347] F., Auricchio, R. L., Taylor, and J. Lubliner. *"Shape-Memory Alloys: Macromodeling and Numerical Simulations of the Superelastic Behavior"*. *Computational Methods in Applied Mechanical Engineering*. Vol. 146. 281–312. 1997.

[348] T. Belytschko, W. K. Liu, and B. Moran. *"Nonlinear Finite Elements for Continua and Structures"*. *Computational Methods in Applied Mechanical Engineering*. John Wiley and Sons. 2000.

[349] David C. Wilcox. *"Reassessment of the Scale-Determining Equation for Advanced Turbulence Models"*. *AIAA Journal*. Vol. 26. 1299–1310. 1988.

[350] F. R. Menter. *"Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications"*. *AIAA Journal*. Vol. 32. 1598–1605. 1994.

[351] J. Chung and G. M. Hulbert. *"A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalized-α Method"*. *Journal of Applied Mechanics*. Vol. 60. 371. 1993.

[352] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. *"Improved Numerical Dissipation for Time Integration Algorithm in Structural Dynamics"*. *Earthquake Engineering and Structural Dynamics*. Vol. 5. 283. 1977.

[353] W. L. Wood, M. Bossak, and O. C. Zienkiewicz. *"An Alpha Modification of Newmark Method"*. *International Journal of Numerical Method in Engineering*. Vol.15. p1562. 1981.

[354] D.J.; Segalman, C.W.G.; Fulcher, G.M.; Reese, and R.V., Jr. Field. *"An Efficient Method for Calculating RMS von Mises Stress in a Random Vibration Environment"*. 117-123. Proceedings of the 16th International Modal Analysis Conference. Santa Barbara, CA. 1998.

[355] G.M., Reese, R.V. Field Jr., and D.J. Segalman. *"A Tutorial on Design Analysis Using von Mises Stress in Random Vibration Environments"*. *The Shock and Vibration Digest*. Vol. 32, No. 6. 466-474 . 2000.

[356] Alan J. Chapman. *Heat Transfer*. 4th Edition. Macmillan Publishing Company. 1984.

[357] J.H. Wilkinson and C. Reinsch. *"Linear Algebra"*. *Handbook for Automatic Computation*. Vol. II. 418–439. Springer-Verlag. New York, NY. 1971.

[358] L.D. Landau and E.M. Lifshitz. *"Electrodynamics of Continuous Media"*. *Course of Theoretical Physics*. Vol. 8. 2nd Edition. Butterworth-Heinemann. Oxford. 1984.

[359] J. F. Nye. *Physical Properties of Crystals: Their Representation by Tensors and Matrices*. Clarendon Press. Oxford. 1957.

[360] J.D. Beley, C. Broudiscou, P. Guillaume, M. Masmoudi, and F. Thevenon. *"Application de la methode des derivees d'ordre eleve a l'optimisation des structures"*. *Revue Europeenne des Elements Finis*. Vol. 5, No. 5-6. 537-567. 1996.

[361] R. Michalowski. Z. Mroz. *"Associated and non-associated sliding rules in contact friction problems"*. *Archives of Mechanics*. Vol. 30, No. 3. 259-276. 1978.

[362] H.D. Nelson and J.M. McVaugh. *"The Dynamics of Rotor-Bearing Systems Using Finite Elements"*. *Journal of Engineering for Industry*. 1976.

[363] X-P Xu and A. Needleman. *"Numerical simulations of fast crack growth in brittle solids"*. *Journal of the Mechanics and Physics of Solids*. Vol. 42. 1397-1434. 1994.

[364] D. Guo, F.L. Chu, and Z.C. Zheng. *"The Influence of Rotation on Vibration of a Thick Cylindrical Shell"*. *Journal of Sound and Vibration*. Vol. 242 3. 492. 2001.

[365] G. Alfano and M.A. Crisfield. *"Finite Element Interface Models for the Delamination Anaylsis of Laminated Composites: Mechanical and Computational Issues"*. *International Journal for Numerical Methods in Engineering*. Vol. 50. 1701-1736. 2001.

[366] A.L. Gurson. *"Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I-Yield Criterion and Flow Rules for Porous Ductile Media"*. *Journal of Engineering Materials and Technology*. Vol. 1. 2-15. 1977.

[367] A. Needleman and V. Tvergaard. *"An Analysis of Ductile Rupture in Notched Bars"*. *Journal of Mechanical Physics Solids*. Vol. 32, No. 6. 461-490. 1984.

[368] M. Geradin and D. Rixen. *Mechanical Vibrations: Theory and Application to Structural Dynamics*. 194. John Wiley and Sons, Inc.. 1997.

[369] K.J. Bathe and C. A. Almeida. *"A Simple and Effective Pipe Elbow Element – Linear Analysis"*. *Journal of Applied Mechanics*. Vol. 47, No. 1. 93-100. 1980.

[370] A. M. Yan, R. J. Jospin, and D. H. Nguyen. *"An Enhanced Pipe Elbow Element – Application in Plastic Limit Analysis of Pipe Structures"*. *International Journal for Numerical Methods in Engineering*. Vol. 46. 409-431. 1999.

[371] J.S. Bergstrom and M.C. Boyce. *"Constitutive Modeling of the Large Strain Time-Dependent Behavior of Elastomers"*. *Journal of the Mechanics and Physics of Solids*. Vol. 46. 931-954. 1998.

[372] J.S. Bergstrom and M.C. Boyce. *"Large Strain Time-Dependent Behavior of Filled Elastomers,"*. *Mechanics of Materials*. Vol. 32. 627-644. 2000.

[373] H. Dal and M. Kaliske. *"Bergstrom-Boyce Model for Nonlinear Finite Rubber Viscoelasticity: Theoretical Aspects and Algorithmic Treatment for the FE Method"*. *Computational Mechanics*. Vol. 44. 809-823. 2009.

[374] A.K. Gupta. *"Response Spectrum Method In Seismic Analysis and Design of Structures"*. CRC Press. 1992.

[375] *"NRC Regulatory Guide"*. Published by the U.S. Nuclear Regulatory Commission, Regulatory Guide 1.92, Revision 2. July 2006.

[376] T. A. Laursen and V. Chawla. *"Design of Energy Conserving Algorithms for Frictionless Dynamic Contact Problems"*. *International Journal for Numerical Methods in Engineering*. Vol. 40. 863–886. 1997.

[377] F. Armero and E. Pet cz. *"Formulation and Analysis of Conserving Algorithms for Dynamic Contact/Impact problems"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 158. 269-300. 1998..

[378] R.W. Ogden and D. G. Roxburgh. *"A Pseudo-Elastic Model for the Mullins Effect in Filled Rubber"*. *Proceedings of the Royal Society of London, Series A (Mathematical and Physical Sciences)*. Vol. 455 No. 1988. 2861-2877. 1999.

[379] K., J. Bose and A. Hurtado, et al. *"Modeling of Stress Softening in Filled Elastomers"*. *Constitutive Models for Rubber III Proceedings of the 3rd European Conference on Constitutive Models for Rubber (ECCMR)*. 223-230. Edited by J. C. Busfield and A. H. Muhr. Taylor and Francis. London. 2003.

[380] M. Geradin and N. Kill. *"A New Approach to Finite Element Modelling of Flexible Rotors,"*. *Engineering Computing*. Vol. 1. March 1984.

[381] M. E. Gurtin. *An Introduction to Continuum Mechanics*. Academic Press. Orlando, Florida. 1981.

[382] M. A. Puso and T. A. Laursen. *"A Mortar Segment-to-Segment Contact Method for Large Deformation Solid Mechanics"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 193. 601-629. 2004.

[383] M. A. Puso and T. A. Laursen. *"A Mortar Segment-to-Segment Frictional Contact Method for Large Deformations"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 193. 4891-4913. 2004.

[384] S. Arndt, B. Svenson, and E. Klingbeil. *"Modellierung der Eigenspannungen an der Rißspitze mit einem Schädigungsmodell"*. *Technische Mechanik*. Vol. 4. 323-332. 1997.

[385] J. Besson and C. Guillemer-Neel. *"An Extension of the Green and Gurson Models to Kinematic Hardening"*. *Mechanics of Materials*. Vol. 35. 1-18. 2003.

[386] I.S. Sandler, F.L. DiMaggio, and G.Y. Baladi. *"Generalized Cap Model for Geological Materials"*. *Journal of the Geotechnical Engineering Division*. Vol. 102. 683-699. 1976.

[387] D. Pelessone. *"A Modified Formulation of the Cap Model"*. *Gulf Atomic Report GA-C19579, prepared for the Defense Nuclear Agency under Contract DNA-001086-C-0277*. January, 1989.

[388] L.E. Schwer and Y.D. Murray. *"A Three-Invariant Smooth Cap Model with Mixed Hardening"*. *International Journal for Numerical and Analytical methods in Geomechanics*. Vol. 18. 657-688. 1994.

[389] C.D. Foster, R.A. Regueiro, A.F. Fossum, and R.I. Borja. *"Implicit Numerical Integration of a Three-Invariant, Isotropic/Kinematic Hardening Cap Plasticity Model for Geomaterials"*. *Computer Methods in Applied Mechanics and Engineering*. Vol. 194. 50-52, 5109-5138. 2005.

[390] Ruijie Liu, Guoyu Lin, and Grama Bhashyam. *"Transformation of the Sandler and Rubin Nonsmooth Cap Model to the Pelessone Smooth Cap Model"*. *ASCE Journal of Engineering Mechanics*. Vol. 136. 680-685. 2010.

[391] WS Atkins PLC, . *"Shell New Wave"*. *Engineering Software Report*. No. ESR960611.

[392] L.A. O'Neill, E. Fakas, and M. Cassidy. *"A Novel Application of Constrained New Wave Theory for Floatover Deck Installations"*. *Proc. OMAE04*. Canada 2004.

[393] R. Pinna and M. Cassidy. *"Dynamic Analysis of a Monopod Platform Using Constrained New Wave"*. *Proc. OMAE04*. Canada 2004.

[394] W.J. Pierson and L. Moskowitz. *"A Proposed Spectral Form of Fully Developed Wind Seas Based on the Similarity Theory of S. A. Kitaigorodskii"*. *Journal of Geophysical Research*. Vol. 69. 5181-5190. 1964.

[395] D.E. Hasselmann, M. Dunckel, and J.A. Ewing. *"Directional Wave Spectra Observed During JONSWAP 1973"*. *Journal Physical Oceanography*. Vol. 10. 1264-1280. 1980.

[396] American Petroleum Institute, . *"Recommended Practice for Planning, Designing and Constructing Fixed Offshore Platforms – Working Stress Design"*. *API RP2A-WSD*. Ed. 20. 1993.

[397] A. Bajer, V. Belsky, and S.W. Kung. *"The Influence of Friction-Induced Damping and Nonlinear Effects on Brake Squeal Analysis"*. 22nd Annual Brake Colloquium & Exhibition. Anheim, CA. 2004. SAE International. Paper Number 2004–01–2794.

[398] A. Q. Tool. *"Relation Between Inelastic Deformability and Thermal Expansion of Glass in its Annealing Range"*. *Journal of the American Ceramic Society*. Vol. 67, No. 10. 240-253. 1946.

[399] A. Q., Markovsky, T. F. Soules, and D.C. Boyd. *"An Efficient and Stable Algorithm for Calculating Fictive Temperatures"*. *Journal of the American Ceramic Society*. Vol. 67 . c56-c57. 1984.

[400] Michael R., Hatch. *"Vibration Analysis using Matlab and ANSYS.*. Chapman & Hall. 2001.

[401] A. Puck, J. Kopp, and M. Knop. *"Guidelines for the Determination of the Parameters in Puck's Action Plane Strength Criterion"*. *Composites Science and Technology*. Vol. 62 . 371-378. 2002.

[402] C. G. Davila, N. Jaunky, and S. Goswami. *"Failure Criteria for FRP Laminates in Plane Stress"*. 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. AIAA Paper 2003-1991, . 2003.

[403] Pinho, et al. *"Failure Models and Criteria for FRP Under In-Plane or Three-Dimensional Stress States Including Shear Non-linearity"*. NASA/TM-2005-213530, . 2005.

[404] E. H., Wong, Y. C. Teo, and T. B. Lim. *"Moisture Diffusion and Vapour Pressure Modeling of IC Packaging"*. *48th Electronics Components and Technology Conference*. 1372-1378. 1998.

[405] N. M. Newmark. *"Method of Computation for Structural Dynamics"*. *ASCE Journal of Engineering Mechanics Division*. Vol. 85. 67-94. 1959.

[406] L. Zhao and A. C. Cangellaris. *"GT-PML: Generalized Theory of Perfectly Matched Layers and its application to the reflectionless truncation of finite-difference time-domain grids"*. *IEEE Trans. Microwave Theory Tech*. Vol. 44. 2555-2563. 1996.

[407] J. F. Allard and N. Atalla. *Propagation of Sound in Porous Media: Modelling Sound Absorbing Materials*. Second Edition. John Wiley & Sons. 2009.

[408] J.C. Whitaker. *The Electronics Handbook*. ISBN 0-8493-1889-0. CRC Press. 2005.

[409] R. M. Hintz. *"Analytical methods in component modal synthesis"*. *AIAA Journal*. Vol 13. 8. 1007-1016. 1975.

[410] D. R. Martinez, T. G. Carne, D. L. Gregory, and A. K. Miller. *"Combined experimental/ analytical modeling using component mode synthesis"*. *AIAA/ASME/ASCE/AHS 25th Structures, Structural Dynamics and Materials Conference, Palm Springs, AIAA Paper 8-0941*. 140-152. 1984.

[411] *"Shock Design Criteria for Surface Ships"*. *Naval Sea Systems Command*. NAVSEA 0908-LP-000-3010, Rev. 1. September 1995.

[412] D. N. Herting. *"A General Purpose, Multi-stage, Component Modal Synthesis Method"*. *Finite Elements in Analysis and Design*. Vol 1. 153-164. 1985.

[413] R. D. Graglia, D. R. Wilton, and A. F. Peterson. *"Higher Order Interpolatory Vector Bases for Computational Electromagnetics"*. *IEEE Transactions on Antennas and Propagation*. Vol 45. 329-342. 1997.

[414] R. Bossart, N. Joly, and M. Bruneau. *"Hybrid Numerical and Analytical Systems for Acoustic Boundary Problems in Thermo-Viscous Fluids"*. *Journal of Sound and Vibration*. Vol. 263. 69-84. 2003.

[415] M. Beltman. *Visco-Thermal Wave Propagating Including Acousto-Elastic Interaction*. *PhD Thesis*. University of Twente, Mechanical Engineering Department. 1998.

[416] Y. Gao and A. F. Bower. *"A Simple Technique for Avoiding Convergence Problems in Finite Element Simulations of Crack Nucleation and Growth on Cohesive Interfaces"*. *Modelling and Simulation in Materials Science and Engineering*. Vol 12. 453-463. 2004.

[417] R. A. Dalrymple and J. C. Cox. *"Symmetric Finite-Amplitude Rotational Water Waves"*. *Journal of Physical Oceanography*. Vol 6. 847-852. 1976.

[418] H. C. Meng and K. C. Ludema. *"Wear models and predictive equations: their form and content"*. *Wear*. Vol 181-183. 443-457. 1995.

[419] J. F. Archard. *"Contact and rubbing of flat surfaces"*. *Journal of Applied Physics*. Vol 24. 1. 18-28. 1953.

[420] *Piping Benchmark Problems - Dynamics Analysis Uniform Support Motion Response Spectrum*. Published by the U.S. Nuclear Regulatory Commission - NUREG/CR-1677.

[421] P. Paultre. *Dynamics of Structures*. 784. ISTE/Wiley. 2010.

[422] P. Léger, I. M. Idé, and P. Paultre. *"Multiple-support seismic analysis of large structures"*. *Computers & Structures*. Vol 36. 6. 1153-1158. 1990.

[423] J.M. Dickens, J.M. Nakagawa, and M.J. Wittbrodt. *"A critique of mode acceleration and modal truncation augmentation methods for modal response analysis"*. *Computers & Structures*. Vol 62. 6. 985-998. 1997.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Z