

Modélisation et Simulation

Cours 1 : Organisation et Eléments Introductifs

Gaëtan Hello

Université d'Evry Val d'Essonne
UFR Sciences et Technologies
gaetan.hello@ufrst.univ-evry.fr

2015-2016

1 Organisation

2 Introduction à la Modélisation

1 Organisation

- Contenu de l'enseignement
- Programmation avec Python - Anaconda : exemple

2 Introduction à la Modélisation

1 Organisation

- Contenu de l'enseignement
- Programmation avec Python - Anaconda : exemple

2 Introduction à la Modélisation

Contenu de l'enseignement

Compétences visées :

- ▶ connaissances de base en modélisation numérique pour des applications potentielles dans différents champs disciplinaires (GE, GI, GM, GSI...),
- ▶ capacité à utiliser et rédiger des programmes informatiques simples pour la résolution de problèmes en science physique et de l'ingénieur.

Pré-requis :

- ▶ bases en analyse (dérivées partielles, intégration, ...),
- ▶ bases en algèbre (opérations sur vecteurs et matrices, ...),
- ▶ bases en algorithmie (variables, boucles, structures conditionnelles).

Contenu de l'enseignement

Plan des cours (6 séances de 2H) :

- ▶ Cours 1 : Organisation et introduction à la notion de modélisation
- ▶ Cours 2 : Modèles pour la description de la physique
 - analyse dimensionnelle
 - équations différentielles
 - résolution analytique et numérique (schéma explicite)
- ▶ Cours 3 : Optimisation continue
 - établissement de fonctions-objectifs à minimiser
 - condition nécessaire d'optimalité
 - résolution analytique et numérique (Newton-Raphson, dichotomie)
- ▶ Cours 4 : Optimisation combinatoire
 - concept de complexités algorithmiques
 - problèmes classiques (voyageur de commerce, ...)
- ▶ Cours 5 : Processus aléatoires
 - Méthodes de Monte-Carlo
 - Chaînes de Markov

Contenu de l'enseignement

Travaux pratiques (3 séances de 4H) :

- ▶ utilisation et rédaction de programmes informatiques simples en langage *python* pour la résolution numérique de problèmes de modélisation décrits dans le cours,
- ▶ le langage de programmation *python* s'avère particulièrement adapté au développement d'applications scientifiques/techniques simples car il s'agit d'un langage :
 - de haut-niveau,
 - interprété,
 - multiparadigme,
 - à la syntaxe simple,
 - doté d'une librairie standard riche,
 - aisé à étendre par des modules complémentaires,
 - free,
 - opensource ...
- ▶ utilisation de la distribution *Anaconda* proposée par la compagnie *Continuum Analytics* (free, multiplateforme, contenu riche, IDE conviviale...)

1 Organisation

- Contenu de l'enseignement
- Programmation avec Python - Anaconda : exemple

2 Introduction à la Modélisation

Programmation avec Python - Anaconda : exemple

The image shows the Spyder Python IDE interface. On the left, a Python script is open, defining parameters for a projectile and plotting its trajectory. The script includes comments in French and uses NumPy and Matplotlib. A central window titled 'Figure 1' displays a plot of the trajectory, showing a parabolic path of a projectile. The x-axis is labeled 'x (m)' and ranges from 0 to 900. The y-axis is labeled 'y (m)' and ranges from 0 to 140. The plot title is 'Trajectoire du boulet de canon soumis à son poids'. On the right, the 'Explorateur de variables' (Variable Explorer) shows a list of variables and their values. Below it, the 'Console' window shows the execution output, including file paths and warnings.

```
m=10. #masse
g=-9.81 #accélération pesanteur
v0=100. #vitesse initiale
alpha=30. #angle initial en degrés
deltaT=0.1 #pas de temps
x0=0. #position initiale
y0=0. #position initiale
16
17=== paramètres numériques =====
18nInstants=101 #nb d'instants ou est d
19
20=== initialisations =====
21alpha=alpha/180*math.pi; #passage de
22tab_t=numpy.zeros(nInstants)
23tab_x=numpy.zeros(nInstants)
24tab_y=numpy.zeros(nInstants)
25
26=== on remplit les tableaux =====
27for i in range(nInstants):
28    tab_t[i]=i*deltaT;
29    tab_x[i]=v0*math.cos(alpha)*tab_t
30    tab_y[i]=1./2.*g*tab_t[i]**2+v0*math.sin(alpha)*tab_t[i]+y0;
31
32=== affichage de la courbe x(t) =====
33matplotlib.pyplot.plot(tab_x,tab_y,label='y(x)',linewidth=3,color=[0,0,1])
34matplotlib.pyplot.xlabel('x (m)')
35matplotlib.pyplot.ylabel('y (m)')
36matplotlib.pyplot.title('Trajectoire du boulet de canon soumis à son poids')
37matplotlib.pyplot.grid(b=None, which='major', axis='both');
38matplotlib.pyplot.legend()
39matplotlib.pyplot.savefig("graphe.pdf")
40
```

Nom	Type	Taille	Valeur
alpha	float	1	0.5235987755982988
deltaT	float	1	0.1
g	float	1	-9.81
i	int	1	100
indiceColonne	int	1	0
j	int	1	999
m	float	1	10.0
matriceCompteurs	float64 (3L 61L)		array([[0., 311., 55.
matriceProba	float64 (3L 3L)		array([[0.9, 0.3, 0.8],
n1	int	1	868
n2	int	1	80
n3	int	1	52
nEtudiants	int	1	1000
nInstants	int	1	101
nPeriodes	int	1	60

```
22/01/2015 22:49 <REP>
22/01/2015 22:40 <REP>
22/01/2015 22:49 1 064 trajectoire.py
1 fichier(s) 1 064 octets
2 Rép(s) 456 999 473 152 octets libres

In [26]:
runfile("D:/SCIENCE/ENS/1415/S2/UE222/COURS/1/PP/trajectoire.p
y", wdir="D:/SCIENCE/ENS/1415/S2/UE222/COURS/1/PP")
C:\Python36\Anaconda110\site-
packages\matplotlib\axes\_axes.py:475: UserWarning: No
labelled objects found. Use labels... kwarg on individual
plots.
warnings.warn("No labelled objects found. ")

In [27]:
runfile("D:/SCIENCE/ENS/1415/S2/UE222/COURS/1/PP/trajectoire.p
y", wdir="D:/SCIENCE/ENS/1415/S2/UE222/COURS/1/PP")

In [28]:
```

Programmation avec Python - Anaconda : exemple

```
import numpy          #bibliothèque pour les tableaux
import math           #bibliothèque de fct mathématiques
import matplotlib     #bibliothèque pour afficher des courbes

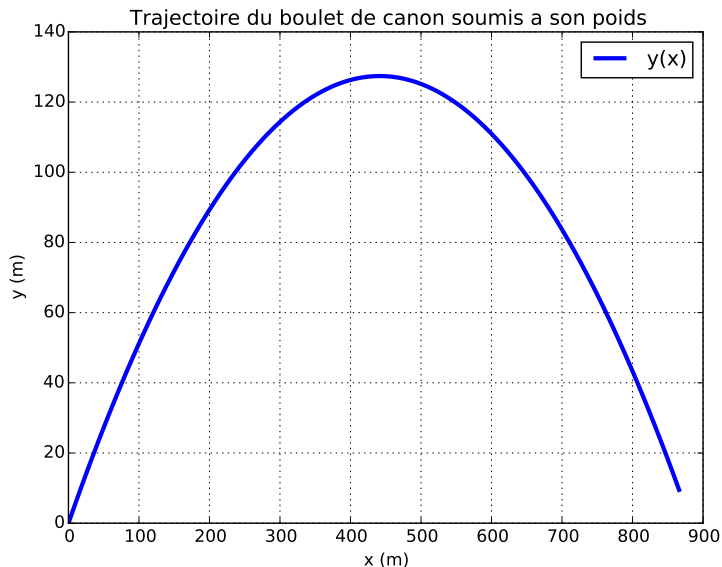
#=== paramètres physiques =====
m=10.                #masse
g=9.81               #accélération pesanteur
v0=100.              #vitesse initiale
alpha=30.            #angle initial en degrés
deltaT=0.1           #pas de temps
x0=0.                #position initiale
y0=0.                #position initiale

#=== paramètres numériques =====
nInstants=101        #nb d'instants où est calculée la trajectoire
```

Programmation avec Python - Anaconda : exemple

```
#=== initialisations =====  
tab_t=numpy.zeros(nInstants)  
tab_x=numpy.zeros(nInstants)  
tab_y=numpy.zeros(nInstants)  
alpha=alpha/180.*math.pi; #passage de l'angle en radian  
  
#=== on remplit les tableaux =====  
for i in range(nInstants):  
    tab_t[i]=i*deltaT;  
    tab_x[i]=v0*math.cos(alpha)*tab_t[i]+x0;  
    tab_y[i]=-1./2.*g*tab_t[i]**2+v0*math.sin(alpha)*tab_t[i]+y0;  
  
#=== affichage de la courbe x(t) =====  
matplotlib.pyplot.plot(tab_x,tab_y)
```

Programmation avec Python - Anaconda : exemple



1 Organisation

2 Introduction à la Modélisation

- Concepts
- Exemples

1 Organisation

2 Introduction à la Modélisation

- Concepts
- Exemples

Modèles et modélisations

Une définition d'un modèle : il s'agit d'un instrument d'intelligibilité du phénomène étudié constituant une représentation formelle de celui-ci s'appuyant sur une syntaxe symbolique.

On distingue plusieurs fonctions pour le modèle :

- ▶ fonction de *substitution* : l'étude d'un phénomène par nature trop complexe sera réalisée au moyen d'un modèle, la simplification opérée rend alors accessibles des éléments de compréhension,
- ▶ fonction de *représentation* : à partir du tout, des éléments pertinents sont extraits et leurs inter-relations explicitées,
- ▶ fonction d'*explication* : les paramètres du modèle peuvent être adaptés pour corrélérer des données existantes,
- ▶ fonction de *prédiction* : un modèle aux paramètres identifiés pour une instance donnée du phénomène étudié doit permettre la connaissance de l'évolution du phénomène.

La compréhension effective d'un problème du monde réel passe souvent par sa traduction sous forme de modèles successifs :

- 1 Problème réel (données empiriques)
- 2 Modèle physique (symboles : langage courant, schémas, mathématiques,...)
- 3 Modèle numérique, si besoin (mathématiques appliquées)
- 4 Modèle informatique (algorithmique)

La modélisation constitue l'acte de générer un modèle à partir d'informations pré-existantes.

1 Problème du monde réel (données empiriques)

- mouvement des planètes,
- comportement d'une voiture lors d'un crash-test,
- prévisions météorologiques,
- comportement d'un échangeur thermique,
- signature radar d'un avion de combat,
- évolution du PIB d'un état,
- comportement des joueurs "bot" dans un jeu vidéo,
- installation de relais de téléphonie mobile,
- ...

2 Modèle physique (symboles : langage courant, dessins, mathématiques,...)

- équations de la mécanique céleste,
- équations de la mécanique des milieux-continus solides,
- équations de la mécanique des milieux-continus fluides,
- équations de la thermique,
- équations de l'électromagnétisme,
- outils conceptuels d'économétrie,
- outils d'intelligence artificielle,
- équations en optimisation hétérogène multi-critères,
- (des couplages peuvent exister)
- ...

3 Modèle numérique, si besoin (mathématiques appliquées)

- méthode des différences-finies (FD - Finite Difference),
- méthode des éléments-finis (FEM - Finite Element Method),
- méthode des volumes-finis (FV - Finite Volume),
- méthode des éléments de frontière (BEM - Boundary Element Method),
- approches de Monte-Carlo,
- réseau de neurones (*deep learning...*),
- logique floue,
- chaînes de Markov,
- (des couplages peuvent exister)
- ...

4 Modèle informatique (algorithmique, hardware, software)

- hardware
 - unité de calcul : processeur multi-coeurs (CPU), GPU, cluster, grid,
 - mémoire partagée/distribuée ...
- système d'exploitation
 - unix, linux, OS/X, android...
 - windows ...
- paradigme de programmation
 - procédurale,
 - orientée-objet ...
- langage de programmation
 - interprété : matlab, python, PHP...
 - compilé : C, C++, fortran...
 - intermédiaire : java, C#, LISP...
- (des couplages peuvent exister)
- ...

Notion de modélisations

Le passage d'un type de modèle au suivant repose sur une action de modélisation :

- 1 Problème physique (ie réel - données empiriques)
↓ **Modélisation " physique"**
- 2 Modèle physique (langage symbolique)
↓ **Modélisation numérique (FD, FEM, FV, BEM...)**
- 3 Modèle numérique (mathématiques appliquées)
↓ **Modélisation informatique (paradigme, langage...)**
- 4 Modèle informatique (algorithmique, hardware, software)

1 Organisation

2 Introduction à la Modélisation

- Concepts
- Exemples

Exemple 1 : Comportement d'une classe durant un cours

On souhaite modéliser l'évolution de l'état des étudiant durant le cours afin d'ajuster la pratique pédagogique pour s'assurer qu'un maximum d'étudiants soit attentif.

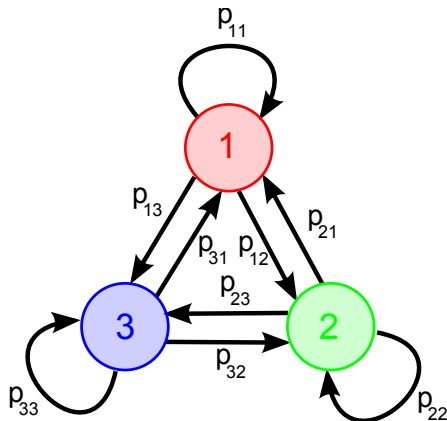
On fait l'hypothèse que chaque étudiant à un instant donné ne peut être que dans l'un des ces 3 états :

- ▶ état 1 : **écoutant**,
- ▶ état 2 : **bavardant**,
- ▶ état 3 : **dormant**.

Les probabilités de transition d'état à l'issue d'une période fixe (1 min par exemple) sont supposées connues. La quantité p_{ij} indique alors la probabilité de passage de l'état i vers l'état j .

Exemple 1 : Comportement d'une classe durant un cours

La modélisation du problème s'appuie ici sur la notion de chaînes de Markov (cours 5). Chaque étudiant évoluera aléatoirement en respectant les probabilités prescrites



les probabilités doivent satisfaire la condition :

$$\sum_{j=1}^3 p_{ij} = 1, \forall i \in \{1, 2, 3\}.$$

Exemple 1 : Comportement d'une classe durant un cours

pour $n = 113$ étudiants

probabilités :

$$p_{11} = 0.9$$

$$p_{12} = 0.05$$

$$p_{13} = 0.05$$

$$p_{21} = 0.3$$

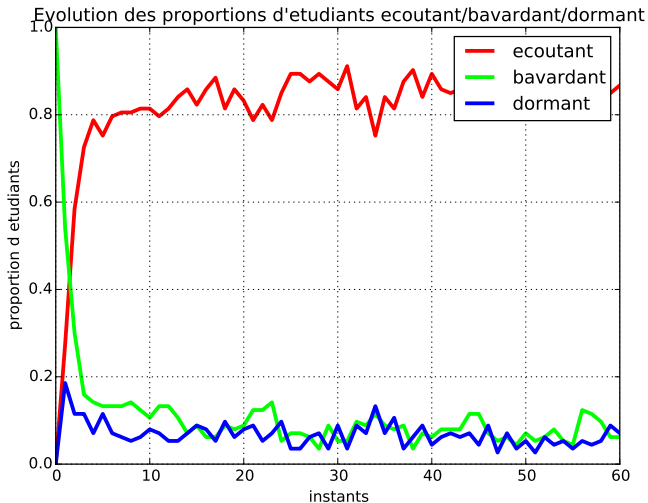
$$p_{22} = 0.5$$

$$p_{23} = 0.2$$

$$p_{31} = 0.8$$

$$p_{32} = 0.05$$

$$p_{33} = 0.15$$



Exemple 2 : trajectoire du boulet de canon avec frottement

On souhaite connaître la trajectoire d'un boulet de canon se déplaçant dans l'air. L'approche consistant à ne tenir compte que de l'effet du poids apparaît irréaliste au regard de constatations empiriques (volant de badminton). Il faut désormais également prendre en compte l'effet du frottement qu'exerce l'air sur le boulet.

Parmi les différents paramètres influents (forme du projectile, température de l'air ...), il ressort que le plus important est la vitesse relative \vec{v} entre le solide et le fluide.

En première approximation, l'action du fluide tendant à s'opposer à l'avancée du solide d'autant plus intensément que la vitesse relative est importante sera représentée par une force d'expression :

$$\vec{f} = -k \cdot \vec{v}$$

Exemple 2 : trajectoire du boulet de canon avec frottement

L'expression des forces de frottement selon $\vec{f} = -k \cdot \vec{v}$ constitue le modèle de Stokes (adapté aux "faibles" vitesses). Il existe d'autres modèles pertinents pour différents régime d'écoulement (turbulent \rightarrow modèle de Newton). Le paramètre k agrège les effets de l'ensemble des paramètres influents négligés.

En utilisant le principe fondamental de la dynamique, le phénomène peut être représenté par l'équation :

$$m \cdot \ddot{\vec{x}} = \vec{P} + \vec{f} \quad (1)$$

Ainsi en projetant dans un repère cartésien $(O, \vec{e}_1, \vec{e}_2)$ où $\vec{P} = -m \cdot g \cdot \vec{e}_2$ et $\vec{f} = -k \cdot \dot{\vec{x}}$, il vient :

$$m \cdot \ddot{x}_1(t) = -k \cdot \dot{x}_1(t)$$

$$m \cdot \ddot{x}_2(t) = -m \cdot g - k \cdot \dot{x}_2(t)$$

Exemple 2 : trajectoire du boulet de canon avec frottement

Equations du mouvement avec prise en compte du poids seul :

$$x_1(t) = \dot{x}_1^0 \cdot t + x_1^0$$

$$x_2(t) = -\frac{1}{2} \cdot g \cdot t^2 + \dot{x}_2^0 \cdot t + x_2^0$$

Equations du mouvement avec prise en compte du frottement selon le modèle de Stokes :

$$x_1(t) = \dot{x}_1^0 \cdot \frac{m}{k} \cdot \left[1 - \exp\left(-\frac{k}{m} \cdot t\right) \right] + x_1^0$$

$$x_2(t) = \left(\dot{x}_2^0 + \frac{m \cdot g}{k} \right) \cdot \frac{m}{k} \cdot \left[1 - \exp\left(-\frac{k}{m} \cdot t\right) \right] - \frac{m \cdot g}{k} \cdot t + x_2^0$$

Que se passe-t-il lorsque $t \rightarrow +\infty$ pour chacune des approches ?

Exemple 2 : trajectoire du boulet de canon avec frottement

paramètres
(unités SI) :

$$m = 10$$

$$g = 9.81$$

$$v_0 = 100$$

$$\alpha = 30^\circ$$

$$k = 1$$

$$x_0 = 0$$

$$y_0 = 0$$

$$\dot{x}_0 = v_0 \cdot \cos(\alpha)$$

$$\dot{y}_0 = v_0 \cdot \sin(\alpha)$$

