

Une Brève Introduction à Matlab

Gaëtan Hello

Université d'Evry Val d'Essonne
UFR S&T - Laboratoire de Mécanique et d'Energétique d'Evry
gaetan.hello@ufrst.univ-evry.fr

Semestre de Printemps 2014

- 1 Le logiciel
- 2 Variables et fonctions
- 3 Boucles et instructions conditionnelles
- 4 Graphiques

1 Le logiciel

- Généralités
- Interface graphique
- Editeur de programme

- ▶ Le logiciel Matlab est un environnement de simulation numérique convivial doté de nombreuses fonctionnalités adaptées aux besoins en sciences et techniques,
- ▶ Matlab est également un langage informatique interprété de haut niveau,
- ▶ Le logiciel est largement utilisé tant dans le monde industriel qu'académique en raison principalement de sa souplesse de programmation et de la richesse des bibliothèques disponibles,
- ▶ Matlab étant un logiciel propriétaire payant, on pourra éventuellement se tourner vers les alternatives free et open-source que sont Scilab et Octave.

Interface graphique Matlab R2012b

MATLAB R2012b

HOME PLOTS APPS

New Script New Open Find Files Import Data Save Workspace Clear Workspace Analyze Code Run and Time Clear Commands Simulink Library Layout Preferences Set Path

FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES

C:\Documents and Settings\NETPCD1\Bureau

Current Folder

- MonPremierProgramme.m
- Untitled.m

Command Window

```
>> MonPremierProgramme
la somme des 10 premiers entiers vaut : 55
>>
```

Workspace

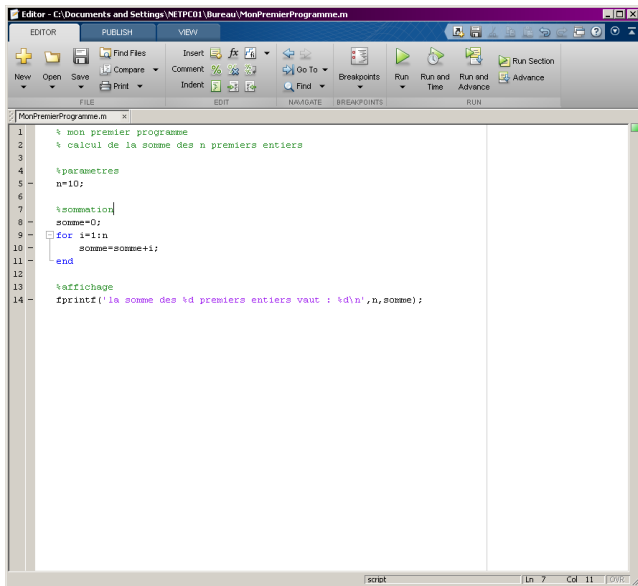
Name	Value
i	10
n	10
somme	55

Command History

- 19/11/2012 13:00
- 19/01/2013 14:00
- 06/03/2014 13:00
 - Untitled
 - Untitled
 - Untitled
 - MonPremierProgramme.m

- ▶ *Command Window* : fenêtre dans laquelle l'utilisateur peut donner des instructions à Matlab sous forme de commandes tapées au clavier (ex : $a=3;b=a+2;$),
- ▶ *Workspace* : propose une interface permettant d'explorer les différents objets que l'utilisateur a créés en mémoire,
- ▶ *Current Folder* : donne accès aux fichiers contenus dans le repertoire de travail actuel de Matlab,
- ▶ *Command History* : liste les commandes successivement tapées par l'utilisateur dans la *command window*.

L'éditeur de programme



The screenshot shows a MATLAB editor window titled "Editor - C:\Documents and Settings\NETPC01\Bureau\MonPremierProgramme.m". The window has a menu bar with "EDITOR", "PUBLISH", and "VIEW". Below the menu bar is a toolbar with icons for New, Open, Save, Find Files, Compare, Print, Insert, Comment, Indent, Go To, Find, Breakpoints, Run, Run and Time, Run and Advance, Run Section, and Advance. The main editing area contains the following code:

```
1 % mon premier programme
2 % calcul de la somme des n premiers entiers
3
4 %parametres
5 n=10;
6
7 %somme
8 somme=0;
9 for i=1:n
10     somme=somme+i;
11 end
12
13 %affichage
14 fprintf('la somme des %d premiers entiers vaut : %d\n',n,somme);
```

The status bar at the bottom of the window shows "script", "Ln 7", "Col 11", and "LIVE".

L'éditeur de programme

- ▶ lorsque l'utilisateur souhaite réaliser des traitements qu'il serait trop fastidieux de définir dans la *command window*, il est possible de regrouper les instructions dans un fichier de commande (fichier `.m`),
- ▶ chaque script/programme/fonction créé par l'utilisateur est stocké dans un fichier ASCII `.m` qu'il sera possible de lire/modifier avec un éditeur de texte,
- ▶ l'exécution des commandes contenues dans un script `.m` se fait en "appelant" ce script par son nom dans la *command window*,
- ▶ il est également possible de "lancer" le programme à partir de l'éditeur en cliquant sur *RUN* (ou en pressant F5).

2 Variables et fonctions

- Variables
- Fonctions et opérations

Variables

- ▶ des nombres

```
a=10;
```

```
b=a*3.5;
```

- ▶ des chaînes de caractères

```
nomCoursPreferé='Modélisation et Simulation';
```

- ▶ des tableaux (de nombres par exemple)

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad (1)$$

```
maMatrice=[1,2,3;4,5,6];
```

- ▶ lecture/écriture dans un tableau

```
a=maMatrice(2,3);%a vaut donc 6 ici
```

```
maMatrice(2,3)=a+1;%la case 2,3 contient désormais 7
```

- ▶ *zeros* : permet d'initialiser un tableau de dimensions données en le remplissant de 0,

```
maMatrice=zeros(2,3);
```

- ▶ *rand* : permet d'initialiser un tableau de dimensions données en le remplissant aléatoirement de valeurs $\in]0, 1[$,

```
maMatrice=rand(2,3);
```

- ▶ *sqrt* : square root - racine carrée

```
a=sqrt(2);
```

- ▶ *cos*, *sin...* : fonctions trigonométriques

```
a=cos(pi/4);
```

Opérations

- ▶ somme et produits de matrices et vecteur

```
A=[1,2;3,4];
```

```
b=[1;2];
```

```
c=A*b;
```

```
B=A*A;
```

- ▶ transposition $B = A^T$

```
A=[1,2;3,4];
```

```
B=A';
```

- ▶ la résolution d'un système linéaire $A \cdot x = b$

```
A=[1,2;3,4];
```

```
b=[5;11];
```

```
x=zeros(2,1);
```

```
x=A\b;
```

3 Boucles et instructions conditionnelles

- Boucle *for*
- Boucle *while*

Boucle *for*

On souhaite calculer la somme S_n des n premiers entiers :

$$S_n = \sum_{i=1}^n i \quad (2)$$

```
n=10;  
somme=0;  
for i=1:n  
    somme=somme+i;  
end
```

Boucle *while* (1)

On souhaite calculer la somme S_n des n premiers entiers :

$$S_n = \sum_{i=1}^n i \quad (3)$$

```
n=10;
somme=0;
i=0;
while(i<n)
    i=i+1;
    somme=somme+i;
end
```

Boucle *while* (2)

On souhaite calculer la somme S_n des n premiers entiers :

$$S_n = \sum_{i=1}^n i \quad (4)$$

```
n=10;
somme=0;
test=false;
i=0;
while(test==false)
    i=i+1;
    somme=somme+i;
    if(i==n)
        test=true;
    end
end
```


4 Graphiques

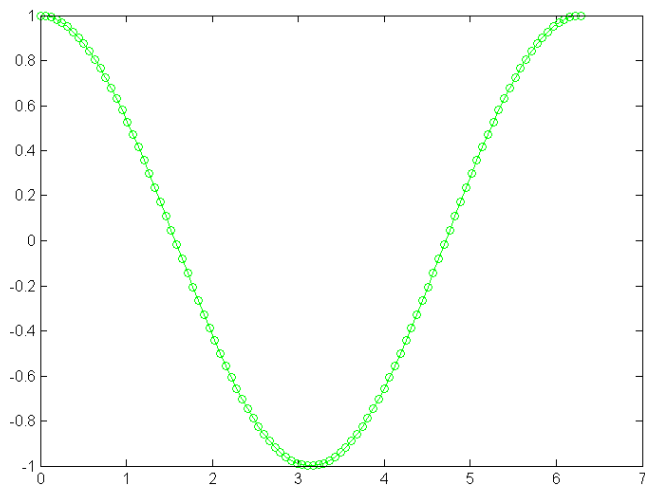
- Figure
- Affichage de multiples entités
- Entités graphiques et paramètres

Figure

On souhaite afficher la courbe représentative de la fonction cosinus pour $\theta \in [0, 2\pi]$.

```
nTheta=100;
theta=zeros(nTheta,1);
cosDeTheta=zeros(nTheta,1);
for i=1:nTheta
    theta(i,1)=(i-1)*2*pi/(nTheta-1);
    cosDeTheta(i,1)=cos(theta(i,1));
end
figure(1)
plot(theta,cosDeTheta,'marker','o','color',[0,1,0],'linestyle','-')
```

Figure

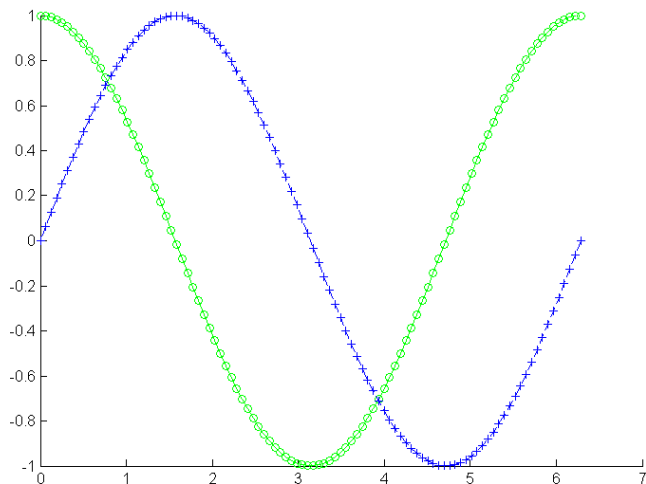


Affichage de multiples entités

On souhaite afficher les courbes représentatives des fonctions cosinus et sinus pour $\theta \in [0, 2\pi]$.

```
nTheta=100;
theta=zeros(nTheta,1);
cosDeTheta=zeros(nTheta,1);
sinDeTheta=zeros(nTheta,1);
for i=1:nTheta
    theta(i,1)=(i-1)*2*pi/(nTheta-1);
    cosDeTheta(i,1)=cos(theta(i,1));
    sinDeTheta(i,1)=sin(theta(i,1));
end
figure(1)
hold on
plot(theta,cosDeTheta,'marker','o','color',[0,1,0],'linestyle','-')
plot(theta,sinDeTheta,'marker','+','color',[0,0,1],'linestyle','--')
hold off
```

Affichage de multiples entités



Entités graphiques et paramètres

- ▶ *grid* : superpose une grille sur la figure courante,
- ▶ *box on* : pour afficher une boîte autour de la figure courante,
- ▶ *xlim, ylim* : pour borner les abscisses et ordonnées de la figure,
- ▶ *legend* : pour afficher la légende,
- ▶ *xlabel, ylabel* : pour donner des titres aux axes,
- ▶ *title* : pour donner un titre au graphique.

```
grid;  
box on;  
xlim([-pi/2,pi/2]);  
ylim([-1,1]);  
legend('courbe1','courbe2');  
xlabel('axe des abscisses');  
ylabel('axe des ordonnées');  
title('Mon graphique');
```