Initiation à Matlab

Florence Hubert

Table des matières

1 Matlab : un système de calcul matriciel

Matlab est l'abrévation en anglais de "matrix laboratory" et par suite tous les objets en Matlab sont des matrices, y compris les scalaires (matrices 1×1).

1.1 Matlab ou Maple ?

Matlab est un programme de calcul numérique, tandis que Maple est un programme de calcul formel et symbolique. Autrement dit,

- si on a besoin d'inverser une matrice comportant des inconnues ou des paramètres, si on doit calculer le discriminant d'un polynôme dont les coefficients sont des paramètres, si on veut obtenir l'expression explicite d'une primitive d'une fonction, on utilisera **Maple**.
- Si au contraire, on doit inverser une matrice composée de nombres, si on veut résoudre une équation aux dérivées partielles, utiliser un schéma numérique d'intégration ou de recherche de minimum, en résumé si la réponse est un nombre décimal et non une expression formelle, on utilisera **Matlab**.

Il est possible d'effectuer des calculs numériques sous Maple, mais l'expérience montre que Matlab est plus adapté. Par contre, il est nécessaire d'utiliser la Toolbox *Calcul symbolique* (qui n'est pas autorisée à l'agrégation) pour faire des calculs formels sous Matlab.

1.2 Première utilisation de Matlab

Sous Unix (Sun) ou Linux (PC), le lancement de Matlab se fait simplement :

- on se place dans une fenêtre (xterm par exemple)
- on se place dans le répertoire dans lequel on veut travailler (par exemple le répertoire Agregation) en tapant **cd Agregation**
- on tape alors **matlab**.

Apparait dans la fenêtre :

```
                    < M A T L A B >
        Copyright 1984-1999 The MathWorks, Inc.
               Version 5.3.0.10183 (R11)
                      Jan 21 1999


  To get started, type one of these: helpwin, helpdesk, or demo.
  For product information, type tour or visit www.mathworks.com.

>>
```

On peut commencer.
Pour sortir de Matlab, il suffit d'utiliser les commandes **quit** ou **exit**. Lorque l'on veut interrompre un programme, la commande **ĉ** permet de récupérer la main.

1.3 Aide en ligne

Matlab contient beaucoup de commandes différentes. De plus la syntaxe des matrices, des boucles, des tests est sensiblement différente de la syntaxe en Maple, Scilab, ... Il parait très utile d'apprendre à se servir de **l'aide en ligne** de Matlab.

- **help** : "help" tout seul donne la liste des aides générales possibles.
- **helpwin** : ouvre une fenêtre et donne accès à une aide détaillée, en gros à tout le manuel de Matlab.
- **help** *nom de commande* : exemple, **help plot** indique la syntaxe des graphes en 2D.
- **lookfor** *nom de commande* : exemple, **lookfor plot** donne une liste de toutes les commandes qui ont un rapport avec **plot**.
- **demo** : lance une démo générale de Matlab.
- **help demos** : donne une liste des demos existantes.

1.4 Premiers calculs dans Matlab

Quelques exemples :

```
>>1+1
ans =
          2
>>pi
ans =
          3.1416
>>eps
ans =
          2.2204e-16
>>i
ans =
          0+ 1.0000i
>>j
ans =
          0+ 1.0000i
>>t=1+1
t =
          2
>>t=1+1;
>>t
t=
          2
>>u=sin(t)
u =
          0.9093
>>v=exp(u)
v =
          2.4826
>>format long
>>v
v =
```

```
          2.48257772801500
>>format short
>>v
v =
          2.4826
>>who
Your variables are:
ans       t         u         v
>>whos
Name      Size         Bytes  Class
ans       1x1              8  double array
t         1x1              8  double array
u         1x1              8  double array
v         1x1              8  double array
Grand total is 4 elements using 32 bytes
leaving 14998496 bytes of memory free.
```

On remarque que :

- par défaut, tout calcul est affecté dans la variable **ans**
- les variables **pi, eps, i, j** sont déja affectées, les variables **i, j** peuvent être réaffectées
- l'affectation se fait grâce au sigle =
- le résultat d'une affectation est imprimé sauf si cette affectation est suivie du sigle **;**
- la commande **format** permet de modifier l'affichage du format des différentes variables
- les commandes **who, whos** permettent de lister l'ensemble des variables utilisées
- la commande **clear** efface le contenu de tous les variables utilisées

Attention : il très fortement déconseillé d'utiliser des noms de variables déja utilisées par Matlab.

1.5 Les matrices dans Matlab

1.5.1 Création de matrices

- $A = [a_{11} \dots a_{1m}; \dots; a_{n1} \dots a_{nm}]$
  Exemple :

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

A =

     1     2     3     4
     2     3     4     1
     3     4     1     2
     4     1     2     3
```

- quelques matrices prédéfinies

```
>>B=zeros(2,3)
B =
     0     0     0
     0     0     0
>>C=ones(3,2)
C =
     1     1
     1     1
     1     1
>>Id=eye(2,3)
Id =
     1     0     0
     0     1     0
>>D=rand(4,4)
D =
    0.8913    0.8214    0.9218    0.9355
    0.7621    0.4447    0.7382    0.9169
    0.4565    0.6154    0.1763    0.4103
    0.0185    0.7919    0.4057    0.8936
>>E=randn(3,4)
E =
   -0.4326    0.2877    1.1892    0.1746
   -1.6656   -1.1465   -0.0376   -0.1867
    0.1253    1.1909    0.3273    0.7258
```

zeros(n,m) construit une matrice nulle de taille $n \times m$,

ones(n,m) construit une matrice dont tous les éléments sont égaux à un et de taille $n \times m$,

eye(n,m) construit une matrice de taille $n \times m$ dont tous les éléments diagonaux sont égaux à un,

rand(n,m) construit une matrice de taille $n \times m$ dont tous les éléments sont choisis aléatoirement avec la loi uniforme sur $[0, 1]$,

randn(n,m) construit une matrice de taille $n \times m$ dont tous les éléments sont choisis aléatoirement avec la loi normale.

- Construction élément par élément

  Exemple :

```
>>for i=1:3,for j=1:4, F(i,j)=i+(j-1)*3;end;end;
>>F
F =
     1     4     7    10
     2     5     8    11
     3     6     9    12
```

- la commande **size** permet d'obtenir la taille de la matrice :

```
>>size(A)
ans =
     4     4
>>size(F)
ans =
```

```
     3     4
>>size(F,1)
ans =
     3
>>size(F,2)
ans =
     4
```

1.5.2 Matrices et vecteurs

Les vecteurs dans Matlab sont écrits en ligne :

```
>>for i=1:3, V(i)=i;end;
>>V
V =
     1     2     3
>>size(V)
ans =
     1     3
```

Une matrice $n \times m$ est stockée colonnes après colonnes dans Matlab comme un vecteur de taille $n * m$:

```
>>for i=1:3,for j=1:4, F(i,j)=i+(j-1)*3;end;end;
>>F
F =
     1     4     7    10
     2     5     8    11
     3     6     9    12
>>F(4)
ans =
       4
```

De même, on peut affecter directement les éléments d'une matrice, après avoir défini sa taille :

```
>>F=zeros(3,4);
>>for i=1:12,F(i)=i;end
>>F
F =
     1     4     7    10
     2     5     8    11
     3     6     9    12
```

1.5.3 Les premières opérations sur les matrices

- Transposition :

```
>>F'
ans =
          1     2     3
          4     5     6
          7     8     9
         10    11    12
```

- La diagonale, la trace, le rang, le déterminant

```
>>diag(A)
ans =
     1
     3
     1
     3
>>trace(A)
ans =
     8
>>rank(A)
ans =
     4
>>det(A)
ans =
     160
```

- Addition ou soustraction

```
>>F+E
ans =
    0.5674    4.2877    8.1892   10.1746
    0.3344    3.8535    7.9624   10.8133
    3.1253    7.1909    9.3273   12.7258
```

- Multiplication, la puissance $n^{ième}$

```
>>F*A
ans =
    70    52    46    52
    80    62    56    62
    90    72    66    72
 >>A^3
ans =
   240   244   256   260
   244   256   260   240
   256   260   240   244
   260   240   244   256
```

- L'inverse

```
>>inv(A)
ans =
   -0.2250    0.0250    0.0250    0.2750
    0.0250    0.0250    0.2750   -0.2250
    0.0250    0.2750   -0.2250    0.0250
    0.2750   -0.2250    0.0250    0.0250
```

1.5.4 Matrices et tableaux

Une matrice peut être considérée élément par élément, c'est alors un tableau. Les opérations élémentaires sur les tableaux sont alors les suivantes :

- Addition ou soustraction : pas de changements

- Multiplication, la puissance $n^{ième}$

```
>>D.*A
ans =
    0.8913    1.6428    2.7654    3.7419
    1.5242    1.3341    2.9528    0.9169
    1.3694    2.4617    0.1763    0.8205
    0.0740    0.7919    0.8114    2.6809
>>A.^3
ans =
     1     8    27    64
     8    27    64     1
    27    64     1     8
    64     1     8    27
```

Remarque : Si les tailles des tableaux ou matrices ne sont pas adaptées, Matlab donne des messages d'erreurs de la forme :

```
>>F.*A
??? Error using ==> .*
Matrix dimensions must agree.
```

1.6 Le calcul vectoriel

Il est souvent intéressant de travailler sur les lignes ou les colonnes d'une matrice avec Matlab.

- **A(i, :)** désigne la $i^{ième}$ ligne de la matrice A
- **A(2:3, :)** désigne les $2^{ième}$ et $3^{ième}$ lignes de la matrice A
- **A(:, j)** désigne la $j^{ième}$ colonne de la matrice A
- **A(:, 1:2:m)** désignent les colonnes impaires de la matrice A

Remarque :

- **xmin : dx : xmax** désigne une discrétisation de pas dx entre xmin et xmax, les deux extrémités étant incluses.
- **xmin : xmax** idem avec par défaut dx=1.

2 Programmation en Matlab

2.1 Créer un programme

2.1.1 Les fonctions élémentaires classiques

Un certain nombre de fonctions élémentaires sont prédéfinies par Matlab : **exp, sin, cos, abs, ...** (voir le paragraphe 4.3 pour plus de détails). La plupart de ces fonctions agissent également sur des matrices ou vecteurs :

```
>>x=[0:0.5:pi];
>>sin(x)
ans =
         0    0.4794    0.8415    0.9975    0.9093    0.5985    0.1411
```

2.1.2 Syntaxe des opérations logiques classiques

- == égalité
- >= supérieur ou égal
- ~= différent
- | ou
- & et
- ...

2.1.3 Syntaxe des boucles classiques

- **for** i=imin : imax,
  < commandes >,
  end
  ou
  for i=imin : h : imax,
  < commandes >,
  end
  Exemples :

```
>>for i=1:2:10,X(i)=i;end
>>X
X =
     1     0     3     0     5     0     7     0     9
>>for i=10:-1:1,Y(i)=10-i;end
>>Y
Y =
     9     8     7     6     5     4     3     2     1     0
```

- **while** < test >,
  < commandes >
  end

- **if**< test >,
  < commandes >
  elseif < test >,
  < commandes >
  else < commandes >,
  end

2.1.4 Les fonctions ou procédures

On va créer à l'extérieur de matlab un fichier "texte" à l'aide de n'importe quel éditeur de texte **emacs, nedit, Text editor, ...** que l'on appelera *nomdefonction*.**m** ou *nomdeprocedure*.**m**. On va décrire la syntaxe de ces fichiers à l'aide de deux exemples.

- **Les fonctions**

  On va créer une fonction $f : (x, l) \to \frac{1}{x^2+l}$. Le premier exemple est édité sous le nom **f1.m**.

```
function [y]=f1(x,l)
y=1/(l+x^2);
% on peut agrémenter tous ces fichiers par des commentaires
% qui seront précédés du sigle %
```

  Une autre possibilité permet de travailler avec des variables x qui seront des tableaux (fichier **f2.m**):

```
function [y]=f2(x,l)
y=1./(l+x.^2);
% il suffit de remplacer les  opérateurs de multiplication
% classique par les opérateurs de multiplication des tableaux
```

- **Les procédures**

  On se donne une discrétisation de l'intervalle $[0, 1]$ de pas h, et on crée une procédure qui calcule les vecteurs X et F de composantes $X(i) = (i-1) * h, F(i) = f((i-1) * h, l)$.

  Le premier programme est sauvegardé sous le nom essai1.m et **fait appel à la fonction f1.m** qui doit se trouver dans le même répertoire.

```
%essai1.m
l=1;
h=0.1;
N=floor(1/h); % partie entière
for i=1:N+1, X(i)=(i-1)*h;end
for i=1:N+1, F(i)=f1((i-1)*h,l);end
X % affichage de X
F % affichage de F
```

  Le deuxième programme est sauvegardé sous le nom essai2.m, fait appel à la fonction f2.m qui doit se trouver dans le même répertoire et utilise le calcul vectoriel, la variable h est laissée comme paramètre à choisir ultérieurement.

```
%essai2.m
l=1;
X=[0:h:1];
F=f2(X,l);
X
F
```

Remarques :

- les commentaires situés juste après la première ligne d'une fonction **lambda.m** ou qui sont en tête d'une procédure **lambda.m** sont accessibles dans matlab par la commande **help lambda**.
- On peut accéder directement au listing d'une fonction ou procédure **lambda.m** en utilisant la commande **type lambda**.

2.1.5 Exécuter une procédure par Matlab

Il suffit de taper le nom du fichier après avoir défini les paramètres qu'il utilise.
Exemple :

```
>>essai1
X =
  Columns 1 through 7
         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
  Columns 8 through 11
    0.7000    0.8000    0.9000    1.0000
F =
  Columns 1 through 7
    1.0000    0.9901    0.9615    0.9174    0.8621    0.8000    0.7353
  Columns 8 through 11
    0.6711    0.6098    0.5525    0.5000
```

ou encore, comme h est laissé comme paramètre dans le programme essai2.m

```
>>h=0.2;
>>essai2
X =
    0    0.2000    0.4000    0.6000    0.8000    1.0000
F =
    1.0000    0.9615    0.8621    0.7353    0.6098    0.5000
```

2.1.6 Conseils de programmation

Pour gagner du temps en Matlab, il faut éviter le plus possible les boucles dans les programmes.
Illustrons ce fait sur un exemple :

```
%test.m
a=1000*ones(20);z=zeros(20);
tic % initialise le temps
for i=1:20, for j=1:20, z(i,j)=a(i,j);end;end;
toc % affiche le temps écoulé
tic % initialise le temps
z=a;
toc % affiche le temps écoulé


>>test
elapsed_time =
              0.0501
elapsed_time =
              9.1100e-04
```

2.2 Enregistrer ou lire des données créées par Matlab

- **save a** :

  sauve toutes les variables existantes dans le fichier a.mat

- **save a 'F' 'X'**

  sauve les variables F et X dans le fichier a.mat

- **nomdufichier=['ah=' num2str(h) '.mat'];save(nomdufichier,'F','X')**

  sauve les variables F et X dans le fichier nomdufichier.mat

- **save resultat.dat -ascii -double**

  sauve toutes les variables existantes sous forme ascii dans le fichier resultat.dat

- **clear**

  efface toutes les variables et données existantes.

- **load a** ou **load nomdufichier**

  charge les données qui sont dans chacun de ses fichiers (Matlab affecte à nouveau automatiquement les variables F et X).

- **load resultat.dat** charge les données qui sont stockées dans resultat.dat (Matlab ne connait plus dans ce cas les variables F et X).

3 Graphisme dans Matlab

3.1 Graphisme 2D

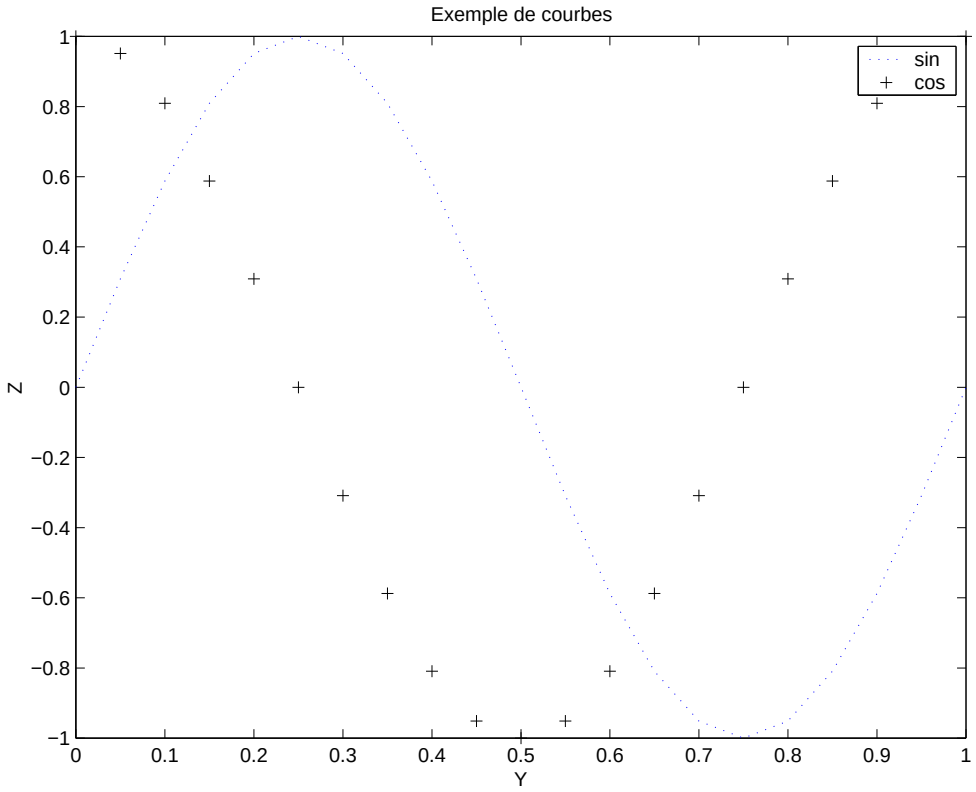
3.1.1 Graphe de points

Soit Y et Z deux vecteurs de taille n, avec $Y = [1, \ldots, n]$.

- Définir une courbe, première possibilité :
  plot(Z) relie les points $(i, Z(i))$
- Définir une courbe, deuxième possibilité :
  plot(Y,Z) relie les points $(Y(i), Z(i))$
- Définir deux ou plusieurs courbes, première possibilité :
  plot(Y,Z,Y,cos(Z)), graphes des points $(Y(i), Z(i))$ et $(Y(i), \cos Z(i))$
- Définir deux ou plusieurs courbes, deuxième possibilité :
  W=[Z',cos(Z)']; plot(Y,W)
- Définir le style des courbes :
  plot(Y,Z,'-r',Y,cos(Z),':') on peut pour chacune des courbes imposer un style de ligne, un style de point, une couleur.
  - Quelques style de lignes :
    - ligne pleine,**:** pointillés, **none** pas de ligne entre les points, ...
  - Quelques couleurs :
    y jaune, **r** rouge, **b** bleu, **g** vert, **w** blanc, **k** noir, ...
  - Quelques styles de points :
    +,**o**, **.**, **square**, **diamond**, **none**, ...
- Contrôler les axes des courbes : le choix de la taille des axes est géré par défaut par Matlab. Pour imposer la taille des axes, il suffit d'utiliser : **axis([xmin,xmax,ymin,ymax])**. Pour obtenir la même échelle sur les deux axes, taper **axis square**.
- Rajouter des annotations sur le graphe :
  - **title** ajoute un titre au graphe,
  - **xlabel** ajoute une légende à l'axe horizontal du graphe,
  - **ylabel** ajoute une légende à l'axe vertical du graphe,
  - **text** ajoute un texte à l'emplacement précisé.
  - **legend** ajoute des légendes au différentes courbes.
  - ...

Pour plus de précision utiliser la commande **help plot**.
Exemple :

```
>> Y=[0:0.05:1];Z1=sin(2*pi*Y);Z2=cos(2*pi*Y);
>> plot(Y,Z1,':b',Y,Z2,'+k');
>> title('Exemple de courbes');
>> xlabel('Y');ylabel('Z');
>> legend('sin','cos');
```



3.1.2 Graphe d'une fonction

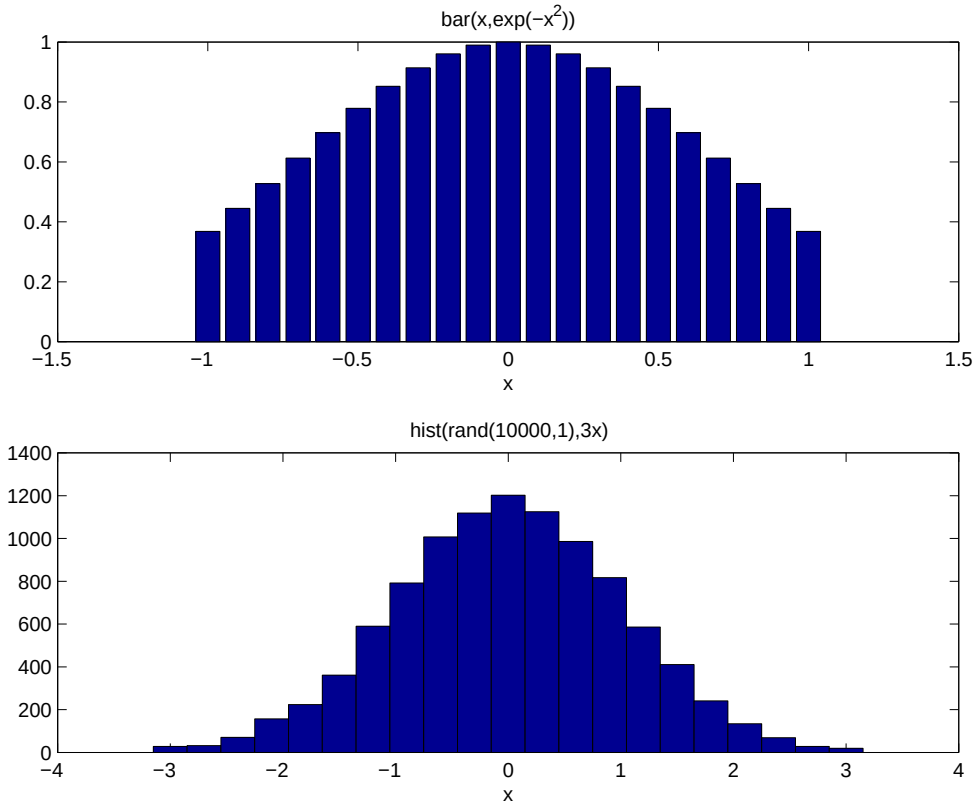
Le graphe d'une fonction sur un intervalle I se ramène au graphe d'un vecteur, en choisissant une discrétisation de cet intervalle I.
Exemple :

```
>>x=[0:0.5:pi];
>>plot(sin(x));
>>plot(x,sin(x));
>>plot(f2(x,1));
```

3.1.3 Histogrammes

L'histogramme d'une fonction est obtenu à l'aide des commandes **hist** ou **bar**.
Exemple :

```
>> x=-1:0.1:1;
>> subplot(2,1,1);bar(x,exp(-x.^2));
>> title('bar(x,exp(-x^2))');xlabel('x');
>> y = randn(10000,1);
>> subplot(2,1,2);hist(y,3*x);
>> title('hist(rand(10000,1),3x)');xlabel('x');
```



3.2 Graphisme 3D

3.2.1 Graphe d'une courbe dans l'espace

Soit X, Y et Z trois vecteurs de taille n.
plot3(X,Y,Z) graphe des points $(X(i), Y(i), Z(i))$

3.2.2 Graphe d'une surface

Soit X et Y deux vecteurs de tailles n et m, et Z une matrice de taille $n \times m$ (resp. soit X, Y et Z trois matrices de taille $n \times m$).
La génération des surfaces se fait à l'aide des deux commandes **surf** et **mesh**.

- **surf(Z)** trace une surface "pleine" passant par les points $(i, j, Z(i, j))$
- **surf(X,Y,Z)** trace une surface "pleine" passant par les points $(X(i), Y(j), Z(i, j))$ (resp. par les points $(X(i, j), Y(i, j), Z(i, j))$)
- **mesh(Z)** trace un maillage ("fil de fer") passant par les points $(i, j, Z(i, j))$ représentée par ses courbes de niveau
- **mesh(X,Y,Z)** trace un maillage ("fil de fer") passant par les points $(X(i), Y(j), Z(i, j))$ (resp. par les points $(X(i, j), Y(i, j), Z(i, j))$)
- **contour(Z)** surface passant par les points $(i, j, Z(i, j))$ représentée par ses courbes de niveau
- autres possibilités : **surfc**, **meshc**, **meshz**, **pcolor**, **surfl**,... Voir l'aide en ligne.

Remarque : La gestion des axes et des annotations graphiques est identique à celle du graphisme 2-D.

3.2.3 Graphe d'une fonction de deux variables réelles ou d'une surface paramétrée

De même que pour le graphisme 2D, on se ramène au graphe d'une courbe ou de matrices :

- Courbe dans l'espace : $t \to (x(t), y(t), z(t))$ pour $t \in I$. On se donne une discrétisation de I : T vecteur de taille n et on se ramène au graphe des vecteurs $(x(T), y(T), z(T))$.
- Surface dans l'espace donnée par $(x, y) \to z(x, y)$ pour $x \in I, y \in J$. On se donne une discrétisation de I et de J : I_x vecteur de taille n, J_y vecteur de taille m et on se ramène au graphe de la matrice $z(I_x, J_y)$.
- Surface paramétrée donnée par $(u, v) \to (x(u, v), y(u, v), z(u, v))$ pour $u \in I, v \in J$. On se donne une discrétisation de I et de J : I_u vecteur de taille n, J_v vecteur de taille m et on se ramène au graphe des matrices $(x(I_u, J_v), y(I_u, J_v), z(I_u, J_v))$.

3.3 Gestion des figures

3.3.1 Ouvrir une figure

Toute commande graphique (**plot**, **surf**,...) ouvre une fenêtre graphique si aucune n'est encore ouverte. On peut également ouvrir une fenêtre graphique à l'aide de la commande **figure**.

3.3.2 Ouvrir plusieurs figures

Toute nouvelle figure doit être ouverte à l'aide de la commande **figure**. Les commandes graphiques sont effectuées dans la dernière fenêtre activée.

3.3.3 La commande hold

Toute commande graphique (**plot**, **surf**,...) efface la commande précédente. Pour conserver plusieurs courbes ou surfaces sur la même figure, on peut utiliser la commande **hold on**.

3.3.4 Créer plusieurs graphes sur une figure

- Deux graphes côte à côte

```
>>subplot(1,2,1)
>>commande graphique 1
>>subplot(1,2,2)
>>commande graphique 2
```

- Deux graphes l'un au dessous de l'autre

```
>>subplot(2,1,1)
>>commande graphique 1
>>subplot(2,1,2)
>>commande graphique 2
```

- Quatre graphes

```
>>subplot(2,2,1)
>>commande graphique 1 (graphe situé en haut à gauche)
>>subplot(2,2,2)
>>commande graphique 2 (graphe situé en haut à droite)
>>subplot(2,2,3)
```

```
>>commande graphique 3 (graphe situé en bas à gauche)
>>subplot(2,2,4)
>>commande graphique 4 (graphe situé en bas à droite)
```

- ...

3.3.5 Effacer tout le contenu d'une figure

Utiliser la commande **clf**.

3.3.6 Impression et sauvegarde postcript d'une figure

La sortie postcript ou imprimante s'effectue directement à partir de la fenêtre graphique (file – print) ou en cliquant sur le bouton de l'imprimante. Sinon, on peut utiliser la commande **print**. Pour plus de détails **help print**.

3.4 Animation de graphisme

On peut créer une animation de deux façons différentes :

- soit en sauvegardant un certain nombre de figures et en les faisant défiler ultérieurement comme un film,
- soit en effaçant et redessinant au fur et à mesure les objects à l'écran.

3.4.1 Créer un film à l'aide des commandes movie et getframe

Exemple :

```
>>for i=1:5
>>plot(sin(i*pi*[0:0.025:2]));
>>M(:,i)=getframe;
>>end;
>>movie(M);
```

3.4.2 Créer une animation à l'aide des commandes hold on et pause

- Lorsque deux commandes **plot** se succédent, la deuxième efface sur la figure la première. La commande **hold on** permet de garder dans la figure le résultat des deux commandes **plot**.
- La commande **pause** permet d'effectuer un arrêt dans un programme, en particulier entre deux commandes **plot**. Pour relancer le programmer, il suffit d'appuyer sur n'importe quelle touche du clavier.
- La commande **pause(x)** permet d'effectuer un arrêt d'une durée fixée par **x** dans un programme. Plus **x** est grand, plus la pause est longue !

Exemples :

```
y=0:0.1:2;
plot(y,sin(y*pi),'-r');
hold on;
for x=0:0.1:2
plot(x,sin(x*pi),'*');
pause(1);
end;
```

```
y=0:0.1:2;
for x=0:0.1:2
plot(y,sin(y*pi),'-r',x,sin(x*pi),'*');
pause(0.5);
end;
```

4 Annexe : Récapitulatif des commandes Matlab

4.1 Commandes générales

Informations générales	
help	On-line help, display text at command line
helpwin	On-line help, separate window for navigation
helpdesk	Comprehensive hypertext documentation and troubleshooting
demo	Run demonstrations
who	List current variables
whos	List current variables, long form
clear	Clear variables and functions from memory
pack	Consolidate workspace memory
load	Load workspace variables from disk
save	Save workspace variables to disk
quit	Quit Matlab session
exit	Quit Matlab session
what	List Matlab-specific files in directory
lookfor	Search all M-files for keyword
which	Locate functions and files

Eléments de programmation	
if	Conditionally execute statements
else	IF statement condition
elseif	IF statement condition
end	Terminate scope of FOR, WHILE, SWITCH and IF statements
for	Repeat statements a specific number of times
while	Repeat statements an indefinite number of times
break	Terminate execution of WHILE or FOR loop
switch	Switch among several cases based on expression
case	SWITCH statement case
otherwise	Default SWITCH statement case
return	Return to invoking function

Opérateurs arithmétiques		
plus	Plus	+
uplus	Unary plus	+
minus	Minus	-
uminus	Unary minus	-
mtimes	Matrix multiply	*
times	Array multiply	.*
mpower	Matrix power	^
power	Array power	.^
mldivide	Backslash or left matrix divide	\
mrdivide	Slash or right matrix divide	/
ldivide	Left array divide	.\
rdivide	Right array divide	./
kron	Kronecker tensor product	

Liens logiques		
eq	Equal	==
ne	Not equal	≃
lt	Less than	<
gt	Greater than	>
le	Less than or equal	<=
ge	Greater than or equal	>=
and	Logical AND	&
or	Logical OR	|
not	Logical NOT	~
xor	Logical EXCLUSIVE OR	
any	True if any element of vector is nonzero	
all	True if all elements of vector are nonzero	

Temps et dates	
calendar	Calendar
clock	Current time as a date vector
cputime	Elapsed CPU time
date	Current date string
datenum	Serial date number
datestr	Date string format
datevec	Date components
eomday	End of month
etime	Elapsed time
now	Current date and time
tic, toc	Stopwatch timer
weekday	Day of the week

4.2 Matrices et tableaux

Matrices élémentaires et tableaux	
blkdiag	Construct a block diagonal matrix from input arguments
eye	Identity matrix
linspace	Generate linearly spaced vectors
logspace	Generate logarithmically spaced vectors
ones	Create an array of all ones
rand	Uniformly distributed random numbers and arrays
randn	Normally distributed random numbers and arrays
zeros	Create an array of all zeros
: (colon)	Regularly spaced vector

Manipulation des matrices	
cat	Concatenate arrays
diag	Diagonal matrices and diagonals of a matrix
fliplr	Flip matrices left-right
flipud	Flip matrices up-down
repmat	Replicate and tile an array
reshape	Reshape array
rot90	Rotate matrix 90 degrees
tril	Lower triangular part of a matrix
triu	Upper triangular part of a matrix
: (colon)	Index into array, rearrange array

Fonctions de vecteurs	
cross	Vector cross product
intersect	Set intersection of two vectors
ismember	Detect members of a set
setdiff	Return the set difference of two vector
setxor	Set exclusive or of two vectors
union	Set union of two vectors
unique	Unique elements of a vector

Matrices creuses élémentaires	
spdiags	Extract and create sparse band and diagonal matrices
speye	Sparse identity matrix
sprand	Sparse uniformly distributed random matrix
sprandn	Sparse normally distributed random matrix
sprandsym	Sparse symmetric random matrix

Conversion matrices creuses - matrices pleines	
find	Find indices and values of nonzero elements
full	Convert sparse matrix to full matrix
sparse	Create sparse matrix
spconvert	Import matrix from sparse matrix external format

Eléments non nuls des matrices creuses	
nnz	Number of nonzero matrix elements
nonzeros	Nonzero matrix elements
nzmax	Amount of storage allocated for nonzero matrix elements
spalloc	Allocate space for sparse matrix
spfun	Apply function to nonzero sparse matrix elements
spones	Replace nonzero sparse matrix elements with ones

Visualisation des matrices creuses	
spy	Visualize sparsity pattern

Algorithmes de réarangement sur les matrices creuses	
colmmd	Sparse column minimum degree permutation
colperm	Sparse column permutation based on nonzero count
dmperm	Dulmage-Mendelsohn decomposition
randperm	Random permutation
symmmd	Sparse symmetric minimum degree ordering
symrcm	Sparse reverse Cuthill-McKee ordering

4.3 Fonctions élémentaires

Fonctions mathématiques élémentaires	
abs	Absolute value and complex magnitude
acos, acosh	Inverse cosine and inverse hyperbolic cosine
acot, acoth	Inverse cotangent and inverse hyperbolic cotangent
acsc, acsch	Inverse cosecant and inverse hyperbolic cosecant
angle	Phase angle
asec, asech	Inverse secant and inverse hyperbolic secant
asin, asinh	Inverse sine and inverse hyperbolic sine
atan, atanh	Inverse tangent and inverse hyperbolic tangent
atan2	Four-quadrant inverse tangent
ceil	Round toward infinity
complex	Construct complex data from real and imaginary components
conj	Complex conjugate
cos, cosh	Cosine and hyperbolic cosine
cot, coth	Cotangent and hyperbolic cotangent
csc, csch	Cosecant and hyperbolic cosecant
exp	Exponential
fix	Round towards zero
floor	Round towards minus infinity
gcd	Greatest common divisor
imag	Imaginary part of a complex number
lcm	Least common multiple
log	Natural logarithm
log2	Base 2 logarithm and dissect floating-point numbers into exponent
log10	Common (base 10) logarithm
mod	Modulus (signed remainder after division)
nchoosek	Binomial coefficient or all combinations
real	Real part of complex number
rem	Remainder after division
round	Round to nearest integer
sec, sech	Secant and hyperbolic secant
sign	Signum function
sin, sinh	Sine and hyperbolic sine
sqrt	Square root
tan, tanh	Tangent and hyperbolic tangent

Variables spéciales et constantes	
ans	The most recent answer
computer	Identify the computer on which Matlab is running
eps	Floating-point relative accuracy
flops	Count floating-point operations
i	Imaginary unit
Inf	Infinity
inputname	Input argument name
j	Imaginary unit
NaN	Not-a-Number
nargin, nargout	Number of function arguments
pi	Ratio of a circle's circumference to its diameter
realmax	Largest positive floating-point number
realmin	Smallest positive floating-point number
varargin, varargout	Pass or return variable numbers of arguments

Corrélation	
corrcoef	Correlation coefficients
cov	Covariance matrix

4.4 Polynômes

Polynômes	
conv	Convolution and polynomial multiplication
deconv	Deconvolution and polynomial division
poly	Polynomial with specified roots
polyder	Polynomial derivative
polyeig	Polynomial eigenvalue problem
polyfit	Polynomial curve fitting
polyval	Polynomial evaluation
polyvalm	Matrix polynomial evaluation
residue	Convert between partial fraction expansion and polynomial coefficients
roots	Polynomial roots

4.5 Analyse matricielle

Généralités	
cond	Condition number with respect to inversion
condeig	Condition number with respect to eigenvalues
det	Matrix determinant
norm	Vector and matrix norms
null	Null space of a matrix
orth	Range space of a matrix
rank	Rank of a matrix
rcond	Matrix reciprocal condition number estimate
rref, rrefmovie	Reduced row echelon form
subspace	Angle between two subspaces
trace	Sum of diagonal elements

Résolution de systèmes linéaires	
chol	Cholesky factorization
inv	Matrix inverse
lscov	Least squares solution in the presence of known covariance
lu	LU matrix factorization
lsqnonneg	Nonnegative least squares
pinv	Moore-Penrose pseudoinverse of a matrix
qr	Orthogonal-triangular decomposition

Valeurs propres, vecteurs propres	
balance	Improve accuracy of computed eigenvalues
cdf2rdf	Convert complex diagonal form to real block diagonal form
eig	Eigenvalues and eigenvectors
gsvd	Generalized singular value decomposition
hess	Hessenberg form of a matrix
poly	Polynomial with specified roots
qz	QZ factorization for generalized eigenvalues
rsf2csf	Convert real Schur form to complex Schur form
schur	Schur decomposition
svd	Singular value decomposition

Fonctions matricielles	
expm	Matrix exponential
funm	Evaluate functions of a matrix
logm	Matrix logarithm
sqrtm	Matrix square root

Autres fonctions	
qrdelete	Delete column from QR factorization
qrinsert	Insert column in QR factorization

Norme, conditionnement	
condest	1-norm matrix condition number estimate
normest	2-norm estimate

Résolution de systèmes linéaires creux	
bicg	BiConjugate Gradients method
bicgstab	BiConjugate Gradients Stabilized method
cgs	Conjugate Gradients Squared method
cholinc	Sparse Incomplete Cholesky and Cholesky-Infinity factorizations
cholupdate	Rank 1 update to Cholesky factorization
gmres	Generalized Minimum Residual method (with restarts)
luinc	Incomplete LU matrix factorizations
pcg	Preconditioned Conjugate Gradients method
qmr	Quasi-Minimal Residual method
qr	Orthogonal-triangular decomposition
qrdelete	Delete column from QR factorization
qrinsert	Insert column in QR factorization
qrupdate	Rank 1 update to QR factorization

Valeurs propres de matrices creuses	
eigs	Find eigenvalues and eigenvectors
svds	Find singular values

4.6 Approximation

Méthodes d'intégration numérique	
dblquad	Numerical double integration
quad, quad8	Numerical evaluation of integrals

Résolution numériques d'équations différentielles ordinaires	
ode45, ode23	Solve differential equations
ode113, ode15s	Solve differential equations
ode23s, ode23t, ode23tb	Solve differential equations
odefile	Define a differential equation problem for ODE solvers
odeget	Extract properties from options structure created with odeset
odeset	Create or alter options structure for input to ODE solvers

Résolution d'équations non linéaires	
fzero	Zero of a function of one variable
vectorize	Vectorize expression

Interpolation de données	
griddata	Data gridding
interp1	One-dimensional data interpolation (table lookup)
interp2	Two-dimensional data interpolation (table lookup)
interp3	Three-dimensional data interpolation (table lookup)
interpft	One-dimensional interpolation using the FFT method
interpn	Multidimensional data interpolation (table lookup)
meshgrid	Generate X and Y matrices for three-dimensional plots
ndgrid	Generate arrays for multidimensional functions and interpolation
spline	Cubic spline interpolation

Discrétisation d'équations aux dérivées partielles et minimisation	
del2	Discrete Laplacian
diff	Differences and approximate derivatives
fminbnd	Minimize a function of one variable
fminsearch	Minimize a function of several variables
gradient	Numerical gradient

4.7 Analyse des données et transformation de Fourier

Opérations basiques	
convhull	Convex hull
cumprod	Cumulative product
cumsum	Cumulative sum
cumtrapz	Cumulative trapezoidal numerical integration
delaunay	Delaunay triangulation
dsearch	Search for nearest point
factor	Prime factors
inpolygon	Detect points inside a polygonal region
max	Maximum elements of an array
mean	Average or mean value of arrays
median	Median value of arrays
min	Minimum elements of an array
perms	All possible permutations
polyarea	Area of polygon
primes	Generate list of prime numbers
prod	Product of array elements
sort	Sort elements in ascending order
sortrows	Sort rows in ascending order
std	Standard deviation
sum	Sum of array elements
trapz	Trapezoidal numerical integration
tsearch	Search for enclosing Delaunay triangle
var	Variance
voronoi	Voronoi diagram

Filtrage et convolution	
conv	Convolution and polynomial multiplication
conv2	Two-dimensional convolution
deconv	Deconvolution and polynomial division
filter	Filter data with an infinite impulse response (IIR) or finite impulse response
filter2	Two-dimensional digital filtering

Transformations de Fourier	
abs	Absolute value and complex magnitude
angle	Phase angle
cplxpair	Sort complex numbers into complex conjugate pairs
fft	One-dimensional fast Fourier transform
fft2	Two-dimensional fast Fourier transform
fftshift	Shift DC component of fast Fourier transform to center of spectrum
ifft	Inverse one-dimensional fast Fourier transform
ifft2	Inverse two-dimensional fast Fourier transform
ifftn	Inverse multidimensional fast Fourier transform
ifftshift	Inverse FFT shift
nextpow2	Next power of two
unwrap	Correct phase angles

4.8 Graphisme

Tracés de courbes et de graphes basiques	
bar	Vertical bar chart
barh	Horizontal bar chart
hist	Plot histograms
hold	Hold current graph
loglog	Plot using log-log scales
pie	Pie plot
plot	Plot vectors or matrices.
polar	Polar coordinate plot
semilogx	Semi-log scale plot
semilogy	Semi-log scale plot
subplot	Create axes in tiled positions

Graphe en 3D	
bar3	Vertical 3-D bar chart
bar3h	Horizontal 3-D bar chart
comet3	3-D comet plot
cylinder	Generate cylinder
fill3	Draw filled 3-D polygons in 3-space
plot3	Plot lines and points in 3-D space
quiver3	3-D quiver (or velocity) plot
slice	Volumetric slice plot
sphere	Generate sphere
stem3	Plot discrete surface data
waterfall	Waterfall plot

Annotation et grilles	
clabel	Add contour labels to a contour plot
datetick	Date formatted tick labels
grid	Grid lines for 2-D and 3-D plots
gtext	Place text on a 2-D graph using a mouse
legend	Graph legend for lines and patches
plotyy	Plot graphs with Y tick labels on the left and right
title	Titles for 2-D and 3-D plots
xlabel	X-axis labels for 2-D and 3-D plots
ylabel	Y-axis labels for 2-D and 3-D plots
zlabel	Z-axis labels for 3-D plots

Surfaces, maillages, contours	
contour	Contour (level curves) plot
contourc	Contour computation
contourf	Filled contour plot
hidden	Mesh hidden line removal mode
meshc	Combination mesh/contourplot
mesh	3-D mesh with reference plane
peaks	A sample function of two variables
surf	3-D shaded surface graph
surface	Create surface low-level objects
surfc	Combination surf/contourplot
surfl	3-D shaded surface with lighting
trimesh	Triangular mesh plot
trisurf	Triangular surface plot

Visualisation de volumes	
coneplot	Plot velocity vectors as cones in 3-D vector field
contourslice	Draw contours in volume slice plane
isocaps	Compute isosurface end-cap geometry
isonormals	Compute normals of isosurface vertices
isosurface	Extract isosurface data from volume data
reducepatch	Reduce the number of patch faces
reducevolume	Reduce number of elements in volume data set
shrinkfaces	Reduce the size of patch faces
smooth3	Smooth 3-D data
stream2	Compute 2-D stream line data
stream3	Compute 3-D stream line data
streamline	Draw stream lines from 2- or 3-D vector data
surf2patch	Convert srface data to patch data
subvolume	Extract subset of volume data set

Graphisme spécialisé	
area	Area plot
box	Axis box for 2-D and 3-D plots
comet	Comet plot
compass	Compass plot
errorbar	Plot graph with error bars
ezcontour	Easy to use contour plotter
ezcontourf	Easy to use filled contour plotter
ezmesh	Easy to use 3-D mesh plotter
ezmeshc	Easy to use combination mesh/contour plotter
ezplot	Easy to use function plotter
ezplot3	Easy to use 3-D parametric curve plotter
ezpolar	Easy to use polar coordinate plotter
ezsurf	Easy to use 3-D colored surface plotter
ezsurfc	Easy to use combination surface/contour plotter
feather	Feather plot
fill	Draw filled 2-D polygons
fplot	Plot a function
pareto	Pareto char
pie3	3-D pie plot
plotmatrix	Scatter plot matrix
pcolor	Pseudocolor (checkerboard) plot
rose	Plot rose or angle histogram
quiver	Quiver (or velocity) plot
ribbon	Ribbon plot
stairs	Stairstep graph
scatter	Scatter plot
scatter3	3-D scatter plot
stem	Plot discrete sequence data
convhull	Convex hull
delaunay	Delaunay triangulation
dsearch	Search Delaunay triangulation for nearest point
inpolygon	True for points inside a polygonal region
polyarea	Area of polygon
tsearch	Search for enclosing Delaunay triangle
voronoi	Voronoi diagram

Angle de vue	
camdolly	Move camera position and target
camlookat	View specific objects
camorbit	Orbit about camera target
campan	Rotate camera target about camera position
campos	Set or get camera position
camproj	Set or get projection type
camroll	Rotate camera about viewing axis
camtarget	Set or get camera target
camup	Set or get camera up-vector
camva	Set or get camera view angle
camzoom	Zoom camera in or out
daspect	Set or get data aspect ratio
pbaspect	Set or get plot box aspect ratio
view	3-D graph viewpoint specification
viewmtx	Generate view transformation matrices
xlim	Set or get the current x-axis limits
ylim	Set or get the current y-axis limits
zlim	Set or get the current z-axis limits

Luminosité	
camlight	Create or position a light
lightangle	Spherical position of a light
lighting	Lighting mode
material	Material reflectance mode

Opérations sur les couleurs	
brighten	Brighten or darken color map
caxis	Pseudocolor axis scaling
colorbar	Display color bar (color scale)
colordef	Set up color defaults
colormap	Set the color look-up table
graymon	Graphics figure defaults set for grayscale monitor
hsv2rgb	Hue-saturation-value to red-green-blue conversion
rgb2hsv	RGB to HSVconversion
rgbplot	Plot color map
shading	Color shading mode
spinmap	Spin the colormap
surfnorm	3-D surface normals
whitebg	Change axes background color for plots

Bibliographie

[1] P. Davis, *Differential equations modeling with matlab*, Prentice Hall, 1999.

[2] M. Mokhtari, A. Mesbah, *Apprendre et maîtriser Matlab*, Springer, 1997.

[3] K. Sigmon, *Matlab Aide mémoire*, Springer, 1999.

[4] C. F. Van Loan, *Introduction to scientific computing, a matrix-vector approach using matlab*, Prentice Hall, 2000.

[5] B. Ycart, *Démarrer en Matlab*, http://www.math-info.univ-paris5.fr/˜ ycart/polys_ ycart.html