Représentations graphiques

Les bibliothèques de Matlab proposent un très grand nombre de fonctions pour la manipulation d'objets graphiques. Nous ne présentons ici que quelques principes de base, utiles pour la visualisation de courbes. Si nous nous concentrons particulièrement sur la représentation graphique à 2 dimensions, il est possible d'aller bien plus loin: graphismes 3D (courbes, maillages, surfaces...) \\
L'interface graphique de MATLAB est sans conteste l'un des points forts du logiciel et facilite le tracé de courbes et l'obtention de graphiques 2D ou 3D de grande qualité (plot, stairs, stem, hist, mesh, surf, plot3).

Matlab possède un vaste ensemble de fonctionnalités graphiques. Nous ne présenterons seulement que quelques principes de base qui serviront à visualiser courbes et surfaces. Que ce soit pour tracer le graphe d'une fonction dans le plan ou dans l'espace, la technique peut se décomposer en trois temps.

- Discrétiser le domaine de représentation.
- Evaluer la fonction en chaque point de ce domaine discrétisé.
- Exécuter l'instruction graphique avec les données précédentes.

La fonction plot :

La fonction plot est sans aucun doute la plus utilisée pour tracer simplement des données contenus dans des tableaux.

le tracé d'une courbe s'effectue à partir de la commande plot().Celle-ci prend en paramètres deux vecteurs et affiche sur un graphique à deux axes chaque couple de points (de même indice). Par exemple, plot(x,y) marquera un point pour chaque couple[x(i),y(i)] avec i allant de 0 à length(x). On représente ainsi les valeurs de y en fonction des valeurs de x. La fonction renvoie une erreur si x et y ne sont pas de même longueur. Si le premier vecteur x est omis, y est tracé en fonction de son indice i.

La fonction plot est utilisable avec des vecteurs ou des matrices. Elle trace des lignes en reliant des points de coordonnées définis dans ses arguments, et elle a plusieurs formes : Si elle contient deux vecteurs de la même taille comme arguments : elle considère les valeurs du premier vecteur comme les éléments de l'axe X (les abscisses), et les valeurs du deuxième vecteur comme les éléments de l'axe Y (les ordonnées).

En tapant >>help plot on a accès à toutes les options disponibles.

L'utilisation la plus simple de la commande plot est la suivante :

>>plot(Vx,Vy)

- %où Vx=[x1 x2 ... xn] est le vecteur d'abscisses,
- %et Vy=[y1 y2 ... yn] le vecteur d'ordonné

Exemple

- Si l'on veut tracer sin(x) sur l'intervalle [-2pi,\; 2pi], on commence par définir une série raisonnable de valeurs équidistantes sur cet intervalle (pas = 2pi/100):

```
x = -2*pi:2*pi/100:2*pi;
plot(x,sin(x));
```

- Pour x variant de -4 jusqu'à 4, avec un pas = 0.5, dessinons la fonction:
- $y = \quad 2x^3 + x^2 \quad 2x + 4$

```
x = -4:0.5:4;
y = -2*x.^3+x.^2-2*x+4;
figure(1)
h=plot(x,y)
title('Dessiner une courbe')
xlabel('l''axe des abscisses')
ylabel('l''axe des ordonnées')
% modification de la largeur du trait (valeur 2)
set(h, 'linewidth',2)
% ajout d'une grille
grid on
```

Annotation d'une figure

Dans une figure, il est préférable de mettre une description textuelle aidant l'utilisateur à comprendre la signification des axes et de connaître le but ou l'intérêt de la visualisation concernée. Il est très intéressant également de pouvoir signaler des emplacements ou des points significatifs dans une figure par un commentaire signalant leurs importances.

- L'instruction title, à laquelle il faut fournir une chaîne de caractères, permet de spécifier le titre du graphique. Le titre apparaît simplement en haut de la fenêtre graphique :

```
>> title('titre de la figure')
```

- Il s'agit d'afficher du texte sous les abscisses (xlabel) et à coté de l'axe des ordonnées (ylabel), afin de spécifier de quoi il s'agit :

Pour donner un titre pour l'axe vertical des ordonnées y, nous utilisons la fonction ylabel comme ceci :

```
>> ylabel('Ceci est l''axe des ordonnées Y')
```

- Pour donner un titre pour l'axe horizontal des abscisses x, nous utilisons la fonction xlabel comme ceci :

```
>> xlabel('Ceci est l''axe des abscisses X')
```

- Dès que l'on trace plusieurs courbes sur le même graphique, il devient indispensable d'ajouter une légende, pour spécifier à quoi correspond chacune des courbes.

- L'instruction legend permet d'ajouter cet élément. Il faut lui communiquer autant de chaînes de caractères que de courbes tracées. Un cadre est alors ajouté sur le graphique, et affiche en face du style de chaque courbe, le texte correspondant. Par exemple :

>> legend('fonction1','fonction2');

- La fonction axis permet de préciser explicitement la zone graphique que l'on souhaite tracer. Cette fonction prend pour argument un vecteur donnant les valeurs extrêmes des abscisses et ordonnées [xmin,xmax,ymin,ymax].

- Lorsque l'on veut faciliter la lecture des valeurs des points d'un tracé, il peut être utile d'ajouter un quadrillage avec l'instruction grid. Un second appel à l'instruction grid fait disparaître le quadrillage.

Pour mettre un quadrillage (une grille), nous utilisons la commande grid (ou grid on). Pour l'enlever nous réutilisons la même commande grid (ou grid off).

>> grid on
>> grid off

- On peut dessiner plusieurs courbes sur un même graphique. L'instruction ***hold on*** permet d'ajouter un ou plusieurs tracés à un graphique déjà existant.

>> hold on
>> hold off

La commande plot prend un troisième argument permettant de spécifier la couleur du tracé et le symbole de représentation. Différentes options sont disponibles (consulter le help plot): Différents types de lignes, symboles de tracé et couleurs peuvent être obtenus avec PLOT (X, Y, S) où S est une chaîne de caractères composée d'un élément de la colonne suivante:

Remarque:

Les instructions pour ajouter le titre, le label des x et le label des y , ainsi que la modification de la largeur du trait (valeur 2) avec set(h, 'linewidh',2) seront ajouter dans cette exemple de script.

La fonction sinus cardinal est définie par :

$$sinc(x) = \frac{\sin(x)}{x}$$

Nous allons nous baser sur un exemple précis de tracé de la fonction sinus cardinal dans l'intervalle [-2pi ; 2pi] avec un pas = 2pi/100,

```
clear all; close all; clc;
x=-2*pi:2*pi/100:2*pi;
% fonction sinc à tracer
y= sinc(x);
% la fonction plot returne le pointeur h
h=plot(x,y,'m')
% fonction de tracer avec PLOT (X, Y, S), S='m' couleur  magenta
xlabel('x');
ylabel('y');
title('sinus cardinal');
% modification de la largeur du trait (valeur 2)
set(h, 'linewidth',2)
```

```
% choix des dimensions des axes
axis=([-7.5 7.5 -0.5 1.2])
% ajout d'une grille
grid on
```

Nous avons utiliser PLOT (X, Y, S), avec S='m' couleur magenta, et set(h, 'linewidth',2) pour modifier la largeur du trait qui vaut 2 dans le tracé gauche de la figure.
S='oc' symbole 'o' et couleur c 'cyan' , la largeur du trait vaut 3 dans le tracé droit de la figure

Tracé de plusieurs courbes } Dans une même fenêtre graphique, on peut tracer plusieurs courbe. On peut dessiner plusieurs courbes sur un même graphique.
L'instruction **hold on** permet d'ajouter un ou plusieurs tracés à un graphique déjà existant. Les tracés sont ajoutés jusqu'au retour du mode par défaut avec **hold off**. A chaque nouvelle commande plot, la figure est remplacée. Pour garder plusieurs courbes, il faut autoriser la superposition de graphique à l'aide de la commande **hold on**. Les plot suivants se superposeront jusqu'à la désactivation **hold off** ou la fermeture de la fenêtre.

Exemple
Si l'on veut tracer le graphe de sin(x) et le graphe de cos(x) sur la même figure sur l'intervalle [-2π,\; 2π],on peut écrire directement : \\

```
x = -2*pi:2*pi/100:2*pi;
y1 = sin(x);
plot(x,y1)
hold on
y2 = cos(x);
plot(x,y2)
hold off
```

Pour afficher la fonction xsin(x) dans l'intervalle [-5pi 5pi] avec un pas: pi/20, il suffit d'écrire

```
 clear all; close all; clc;
   x = -5*pi:*pi/20:5*pi;
  y1 = x.*sin(x);
  y2 = x;
  y3=-x;
plot(x,y1)
hold on
plot(x,y2)
hold on
plot(x,y3)
title('Représentation des trois fonctions y1, y2 et y3')
 xlabel(['x =[ -',num2str(n),' \pi : ',num2str(n),' \pi ]'])
 ylabel('y')
```

```
legend('y1 = x*sin(x)','y2 = x','y3 = -x')
```

Afficher plusieurs graphiques : subplot

Voilà une fonctionnalité très utile pour présenter sur une même zone graphique plusieurs tracés de résultats, par exemple pour les comparer.

L'idée générale est de découper la fenêtre graphique en zones, et d'afficher un graphe dans chacune des zones. On utilise l'instruction subplot en lui spécifiant le nombre de zones sur la hauteur, le nombre de zones sur la largeur, et le numéro de la zone que l'on considère (et dans laquelle on va tracer une courbe) :

subplot(NbZH, NbZL, NZ)

- où **NbZH** représente le nombre de zones sur la hauteur,
- **NbZL** le nombre de zones sur la largeur,
- et **NZ** le numéro de la zone à laquelle on s'intéresse.

Exemple: Découper la fenêtre graphique en zones,

```
clear all; close all; clc;
x = 0:2*pi/100:2*pi;
subplot(221);
plot(x,sin(x));
subplot(222);
plot(x,cos(x),x,sin(x),'-.');
subplot(223);
plot(cos(x),sin(x));
subplot(224);
plot(sin(2*x),sin(3*x));
```

```
clear all; close all; clc;
x = [-2*pi:2*pi/100:2*pi];
figure(1)
subplot(1,2,1)
plot(x, sin(x), 'r')
subplot(1,2,2)
plot(x, cos(x), 'g')
figure(2)
 subplot(2,1,1)
plot(x, x.*x, 'b')
 subplot(2,1,2)
plot(x, exp(-x.*x), 'm')
```