

TP N° 3 : Représentation graphique

Le but de ce TP est de vous donner les outils Matlab permettant de réaliser des représentations graphiques afin de présentant clairement vos données.

I- Graphisme 2D

Principales fonctions de graphisme 2D :

Fonction	Description
plot	Graphe en 2D avec une échelle linéaire
plotyy	Graphe avec deux axes y différents à gauche et à droite
loglog	Graphe en 2D avec une échelle logarithmique pour les deux axes
semilogx	Graphe en 2D avec une échelle logarithmique pour l'axe des x
semilogy	Graphe en 2D avec une échelle logarithmique pour l'axe des y
subplot	Tracer plusieurs graphes alignés sur une même figure
figure	Ouvrir une figure
close	Fermer une figure

Pour bien comprendre ces fonctions nous proposons les activités suivantes :

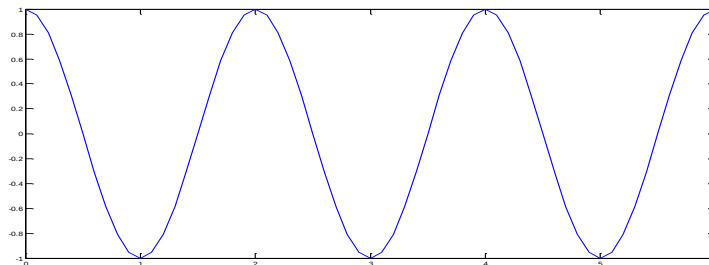
Activité 1 : Tracer la courbe d'une fonction

Nous volons tracer la fonction $f(x) = \cos(\pi x)$ dans l'intervalle $[0, 6]$; pour cela nous allons définir une subdivision (vecteur) x sur l'intervalle $[0, 6]$ et calculer le vecteur $y = f(x)$ en fin nous allons tracer la fonction en utilisant la fonction `plot` et les deux vecteur x et y :

```
>> X = [0:0.1:6];  
>> Y = cos(X*pi);  
>> figure();  
>> plot(X,Y)
```

Code 1

Nous allons obtenir le graphe suivant



NB : dans le code précédant la ligne de commande

```
>> figure();
```

est facultative ; c'est-à-dire que vous pouvez avoir le même résultat en appliquant seulement les commandes :

```
>> X = [0:0.1:6];  
>> Y = cos(X*pi);  
>> plot(X,Y)
```

Code 2

En effet la commande plot trace le graphe dans une figure ; dans le code 1 la commande figure() va créer une nouvelle figure puis la commande plot trace le graphe dans la figure active. Dans le code 2, la commande plot appelle implicitement la commande figure() puisque elle ne trouve aucune figure active.

Si, dans un projet, vous avez plusieurs figures : Figure 1, Figure 2, ... et vous voulez ajouter le graphe de la fonction $f(x) = \cos(\pi x)$ à la figure Figure 2, vous devez d'abord faire appel à la figure 2 en utilisant la commande

```
>> figure(2);
```

En suite applique la commande

```
>> plot(X,Y)
```

Un autre exemple : la fonction $f(x) = \sin(x)$

```
>> x = -pi:.1:pi;
>> y = sin(x);
>> plot(x,y)
>> set(gca,'XTick',-pi:pi/2:pi)%Permet de définir une subdivision sur l'axe OX
>> set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'}) %d'ajouter les labels
pour la subdivision
>> title('Fonction Sinus');% ajouter un titre pour le graphe
>> xlabel('Radians'); % ajouter un label pour l'axe OX
>> ylabel('Valeur de la Fonction'); % ajouter un label pour l'axe Oy
```

Code 3

Activité 2 : Représentation graphique de plusieurs courbes

Méthode 1

Décrivons maintenant quelques fonctions très utiles pour les graphes. Si l'on souhaite placer deux courbes sur le même graph on utilise la commande hold on (à utiliser avec la commande hold off.)

```
x = -pi:.1:pi;
y = sin(x);
z = cos(x);
p1=plot(x,y);
hold on
p2=plot(x,z);
set(p1,'Color','red','LineWidth',2)
set(p2,'LineWidth',2)
set(gca,'XTick',-pi:pi/2:pi)
set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'})
xlabel('Radians');
ylabel('Function Value');
```

Code 4

Méthode 2

Une seconde méthode permet également d'afficher plusieurs courbes sur la même figure. Il suffit d'appeler ces différentes courbes dans la même fonction plot en les séparant par des virgules :

```
figure(1);
y = sin(x);
z = cos(x);
h = plot(x,y,x,z);%tracer à la fois la courbe (x,y) et la courbe (x,z)
xlabel('Radians');
ylabel('Function Value');
```

Activité 3 : Autres exemples de la fonction plot

1. Représentation graphique des colonnes d'une matrice

```
>> X = [16 2 3 13]
>> plot(X)           % c'est équivalent à la commande plot(1:length(X), X)

>> Y = [16; 2; 3; 13]
plot(Y)

>> Z = magic(4)
Z =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
>> plot(Z) % représentation des vecteurs colonnes de la matrice (l'axe des x est toujours 1:n ; avec n le nombre
des lignes
```

2. Tracer le cercle de centre M(4,3) et de rayon r=2

```
r = 2;
xc = 4;
yc = 3;

theta = linspace(0,2*pi);
x = r*cos(theta) + xc;
y = r*sin(theta) + yc;
plot(x,y)
axis equal
```

3. Create line plot using specific line width, marker color, and marker size.

```
x = -pi:pi/10:pi;
y = tan(sin(x)) - sin(tan(x));
plot(x,y, '--rs', 'LineWidth', 2, ...
      'MarkerEdgeColor', 'k', ...
      'MarkerFaceColor', 'g', ...
      'MarkerSize', 10)
```

4. Modify axis tick marks and tick labels and annotate the graph.

```
x = -pi:.1:pi;
y = sin(x);
plot(x,y)
set(gca, 'XTick', -pi:pi/2:pi)
set(gca, 'XTickLabel', {'-pi', '-pi/2', '0', 'pi/2', 'pi'})
title('Sine Function');
xlabel('Radians');
ylabel('Function Value');
```

5. Add a plot title, axis labels, and annotations.

```
x = -pi:.1:pi;
y = sin(x);
p = plot(x,y)
set(gca, 'XTick', -pi:pi/2:pi)
set(gca, 'XTickLabel', {'-pi', '-pi/2', '0', 'pi/2', 'pi'})
xlabel('-\pi \leq \Theta \leq \pi')
ylabel('sin(\Theta)')
title('Plot of sin(\Theta)')
% \Theta appears as a Greek symbol (see String)
```

```
% Annotate the point (-pi/4, sin(-pi/4))
text(-pi/4, sin(-pi/4), '\leftarrow sin(-\pi\div4)', ...
     'HorizontalAlignment', 'left')
% Change the line color to red and
% set the line width to 2 points
set(p, 'Color', 'red', 'LineWidth', 2)
```

Activité 2 : manipulation de la fonction fplot

Exercice 1

En utilisant la fonction fplot tracer la courbe de la fonction sin sur l'intervalle [0 4]

Solution

```
>> fplot(@(x) sin(x))
```

Exercice 2

Tracer la courbe paramétrique

$$f(x) = \begin{cases} x(t) = \cos(3t) \\ y(t) = \sin(2t) \end{cases}$$

Solution

```
xt = @(t) cos(3*t);
yt = @(t) sin(2*t);
fplot(xt, yt)
```

Exercice 3

Tracez trois ondes sinusoïdales de différentes phases. Pour la première, utilisez une largeur de ligne de 2 points. Pour la seconde, spécifiez un style de ligne rouge pointillé avec des marqueurs de cercle. Pour la troisième, spécifiez un style de ligne cyan, pointillé pointillé avec des marqueurs d'astérisque.

Solution

```
fplot(@(x) sin(x+pi/5), 'LineWidth', 2);
hold on
fplot(@(x) sin(x-pi/5), '--or');
fplot(@(x) sin(x), '-.*c')
hold off
```

Exercice 4

Tracer la fonction tangente hyperbolique dans l'intervalle [-2 2]

Solution

```
fnch = @tanh;
fplot(fnch, [-2 2])
```

Activité 4 : Fonction subplot

Une autre approche consiste à créer des sous-figures. La commande `subplot` permet de diviser une figure en plusieurs sous figure ; en effet, la syntaxe `subplot (m,n,p)` va créer une matrice $m \times n$ de figures. Le `p` est un indice prenant les valeurs `1, 2, 3...` $m \times n$, qui permet d'indiquer la sous figure ou on veut positionner le graphe.

Exemple

Dans cet exemple nous allons tracer les courbes des fonctions : $\sin(x)$, $\sin(x+\pi/3)$, $\sin(x+2\pi/3)$, $\cos(x)$, $\cos(x+\pi/3)$ et $\cos(x+2\pi/3)$.

```
x = -pi:.1:pi;
y0 = sin(x);
y1 = sin(x+pi/3);
y2 = sin(x+2*pi/3);
z0 = cos(x);
z1 = cos(x+pi/3);
z2 = cos(x+2*pi/3);

figure(1)
subplot(3,2,1) %diviser la figure en 3x2 sous figures et pointer sur la première
plot(x,y0)
subplot(3,2,2)
plot(x,y1)
subplot(3,2,3)
plot(x,y2)
subplot(3,2,4)
plot(x,z0)
subplot(3,2,5)
plot(x,z1)
subplot(3,2,6)
plot(x,z2)
```

Exercice 1

Créez une figure avec deux tracés empilés. Tracez une onde sinusoïdale dans chacun d'eux.

Solution

```
subplot(2,1,1);
x = linspace(0,10);
y1 = sin(x);
plot(x,y1)

subplot(2,1,2);
y2 = sin(5*x);
plot(x,y2)
```

Exercice 2

Créez une figure divisée en quatre sous-tracés. Tracez une onde sinusoïdale dans chacun d'eux et nommez chaque sous-trame.

Solution

```
subplot(2,2,1)
x = linspace(0,10);
y1 = sin(x);
plot(x,y1)
title('Subplot 1: sin(x)')
```

```

subplot(2,2,2)
y2 = sin(2*x);
plot(x,y2)
title('Subplot 2: sin(2x)')

subplot(2,2,3)
y3 = sin(4*x);
plot(x,y3)
title('Subplot 3: sin(4x)')

subplot(2,2,4)
y4 = sin(8*x);
plot(x,y4)
title('Subplot 4: sin(8x)')

```

Exercice 3

Créez une figure contenant trois tracés :

- deux tracés sur la moitié supérieure de la figure
- un troisième tracé sur la moitié inférieure de la figure.

Ajouter des titres à chacun.

Solution

```

subplot(2,2,1);
x = linspace(-3.8,3.8);
y_cos = cos(x);
plot(x,y_cos);
title('Subplot 1: Cosine')

subplot(2,2,2);
y_poly = 1 - x.^2./2 + x.^4./24;
plot(x,y_poly,'g');
title('Subplot 2: Polynomial')

subplot(2,2,[3,4]);
plot(x,y_cos,'b',x,y_poly,'g');
title('Subplot 3 and 4: Both')

```

Exercice 4

Créer une figure avec quatre tracés de tige de données aléatoires.

Solution

```

for k = 1:4
data = rand(1,10);
subplot(2,2,k)
stem(data)
end

```

Ensuite, remplacer la deuxième sous-trame par des axes vides.

Solution

```

subplot(2,2,2,'replace')

```

Créez une figure avec deux tracés non alignés avec les positions de la grille. Spécifiez une position personnalisée pour chaque tracé.

Solution

```
pos1 = [0.1 0.3 0.3 0.3];  
subplot('Position',pos1)  
y = magic(4);  
plot(y)  
title('First Subplot')  
  
pos2 = [0.5 0.15 0.4 0.7];  
subplot('Position',pos2)  
bar(y)  
title('Second Subplot')
```

Exercise 5

This example passes function handles to fplot, one created from a MATLAB® function and the other created from an anonymous function.

```
hmp = @humps;  
subplot(2,1,1);fplot(hmp,[0 1])  
sn = @(x) sin(1./x);  
subplot(2,1,2);fplot(sn,[.01 .1])
```

Exercise 6

Définissez t comme une subdivision de l'intervalle [0 10pi]. Définir st = sin(t) et ct = cos(t).

En utilisant ces vecteurs, tracer une hélice 3-D

Solution

```
t = 0:pi/50:10*pi;  
st = sin(t);  
ct = cos(t);  
  
figure  
plot3(st,ct,t)
```

Tester également le code suivant

```
t = 0:pi/50:10*pi;  
plot3(sin(t),cos(t),t)  
xlabel('sin(t)')  
ylabel('cos(t)')  
zlabel('t')  
grid on  
axis square
```