

# Le logiciel de simulation MATLAB Partie 2 Graphiques

Professeur Ali Tahri  
Université des sciences et de la technologie d'Oran  
Mohamed Boudiaf

# 1. Graphique simple

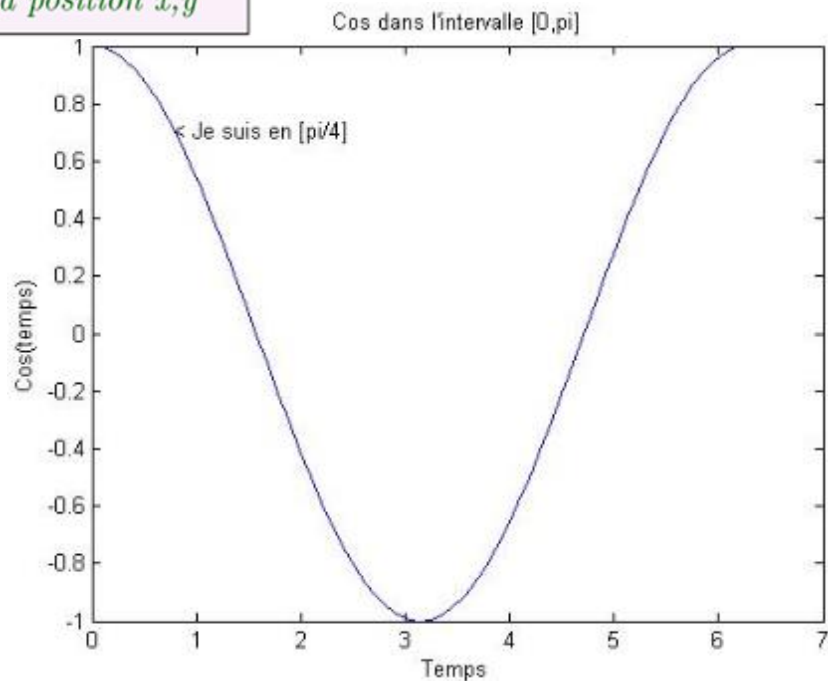
Les commandes suivantes montrent comment afficher un graphique simple. Matlab traite que des valeurs discrètes, il faut d'abord commencer par définir l'intervalle de valeurs de l'abscisse puis on calcule la valeur de la fonction (dans notre cas un cosinus) pour chaque échantillon de temps et l'on arrive au vecteur  $y$ .

```
% — Graphique simple ————— %  
  
t = 0 :pi/30 :2*pi;           % Abscisse de 0 à 2π par pas de π/30  
y = cos(t);                  % Valeurs de la fonction dans le temps  
  
figure;                       % Nouvelle fenêtre  
plot(t,y)                     % Tracer  
title('Cos dans l''intervalle [0,pi]'); % Titre  
xlabel('Temps');              % Légende en abscisse  
ylabel('Cos(temps)');        % Légende en ordonnée  
text(pi/4,cos(pi/4),'i Je suis en [pi/4]') % Chaîne de caractères à la position x,y
```

*% — Graphique simple — %*

```
t = 0 :pi/30 :2*pi;           % Abscisse de 0 à 2π par pas de π/30
y = cos(t);                   % Valeurs de la fonction dans le temps

figure;                       % Nouvelle fenêtre
plot(t,y)                     % Tracer
title('Cos dans l''intervalle [0,pi]'); % Titre
xlabel('Temps');              % Légende en abscisse
ylabel('Cos(temps)');        % Légende en ordonnée
text(pi/4,cos(pi/4),'i Je suis en [pi/4]') % Chaîne de caractères à la position x,y
```



```
>> help plot
```

```
PLOT Linear plot.
```

```
PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix, then the vector is plotted versus the rows or columns of the matrix, whichever line up. If X is a scalar and Y is a vector, disconnected line objects are created and plotted as discrete points vertically at X.
```

```
PLOT(Y) plots the columns of Y versus their index.
```

```
If Y is complex, PLOT(Y) is equivalent to PLOT(real(Y),imag(Y)).
```

```
In all other uses of PLOT, the imaginary part is ignored.
```

Various line types, plot symbols and colors may be obtained with PLOT(X,Y,S) where S is a character string made from one element from any or all the following 3 columns:

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

The X,Y pairs, or X,Y,S triples, can be followed by parameter/value pairs to specify additional properties of the lines. For example, PLOT(X,Y,'LineWidth',2,'Color',[.6 0 0]) will create a plot with a dark red line width of 2 points.

For example, `PLOT(X,Y,'c+:')` plots a cyan dotted line with a plus at each data point; `PLOT(X,Y,'bd')` plots blue diamond at each data point but does not draw any line.

`PLOT(X1,Y1,S1,X2,Y2,S2,X3,Y3,S3,...)` combines the plots defined by the (X,Y,S) triples, where the X's and Y's are vectors or matrices and the S's are strings.

For example, `PLOT(X,Y,'y-',X,Y,'go')` plots the data twice, with a solid yellow line interpolating green circles at the data points.

The PLOT command, if no color is specified, makes automatic use of the colors specified by the axes ColorOrder property. The default ColorOrder is listed in the table above for color systems where the default is blue for one line, and for multiple lines, to cycle through the first six colors in the table. For monochrome systems, PLOT cycles over the axes LineStyleOrder property.

If you do not specify a marker type, PLOT uses no marker.

If you do not specify a line style, PLOT uses a solid line.

`PLOT(AX,...)` plots into the axes with handle AX.

PLOT returns a column vector of handles to lineseries objects, one handle per plotted line.

## 2. Graphique avancé

Matlab permet de personnaliser les graphiques. Par exemple, il est possible de changer la couleur du trait, de marquer les points avec différents symboles tels que x, O, \*,... etc. Voici un aperçu simple de ce qui est possible de faire

```
% — Graphique avancé ————— %

figure;                               % Nouvelle fenêtre

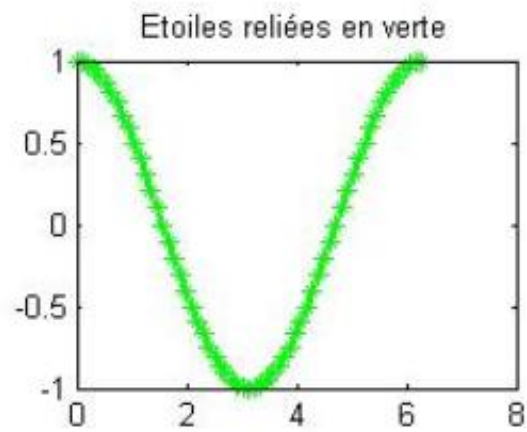
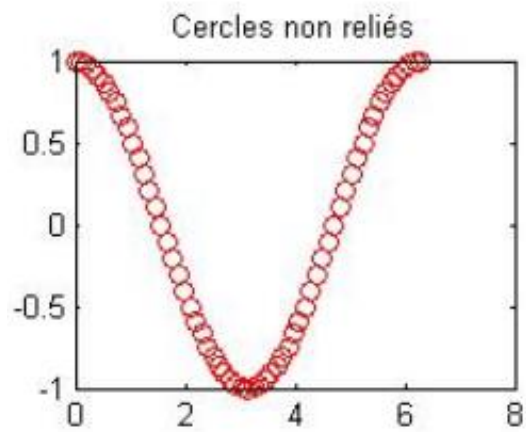
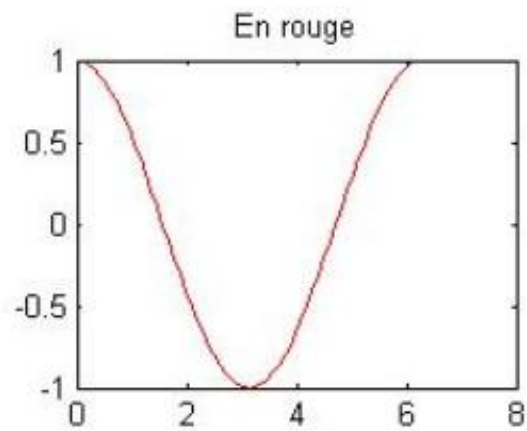
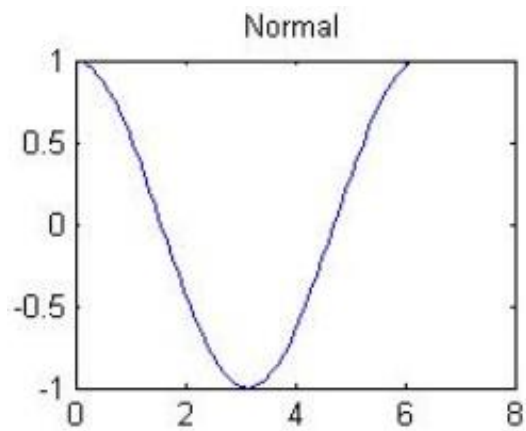
t = 0 :pi/30 :2*pi;                   % Définition de temps
y = cos(t);                            % Fonction du temps

subplot(2,2,1)                         % Zone supérieure gauche
plot(t,y);                             % Tracer
title('Normal');                       % Titre

subplot(2,2,2)                         % Zone supérieure droit
plot(t,y,'r');                         % Tracé en rouge
title('En rouge');                    % Titre

subplot(2,2,3)                         % Zone inférieur gauche
plot(t,y,'or');                       % Tracé points ronds en rouge
title('Cercles non reliés');         % Titre

subplot(2,2,4)                         % Zone inférieur droite
plot(t,y,'*-g');                      % Tracé point étoiles relié en vert
title('Etoiles reliées en verte');   % Titre
```

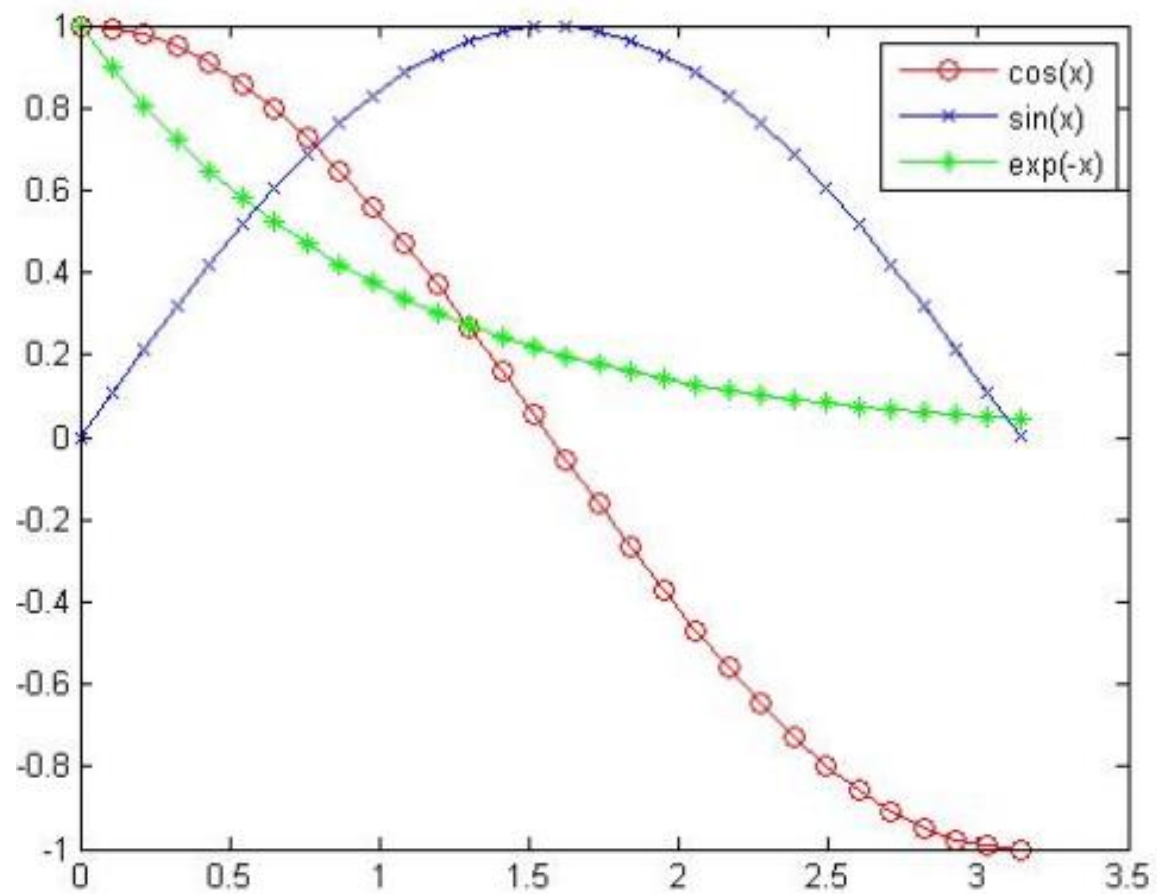




### 3. Graphique multiple 1

Il est possible de tracer plusieurs courbes sur le même graphique. Pour ce faire, une méthode consiste à mettre l'ensemble des fonctions à tracer dans les parenthèses qui suivent la commande plot.

```
% — Graphique multiple 1 ————— %  
  
figure; % Nouvelle fenêtre  
  
x = linspace(0,pi,30); % Fonction du temps  
plot(x,cos(x),'o-r',x,sin(x),'x-b',x,exp(-x),'*-g') % Tracer  
legend('cos(x)', 'sin(x)', 'exp(-x)') % Légende sur graphique
```

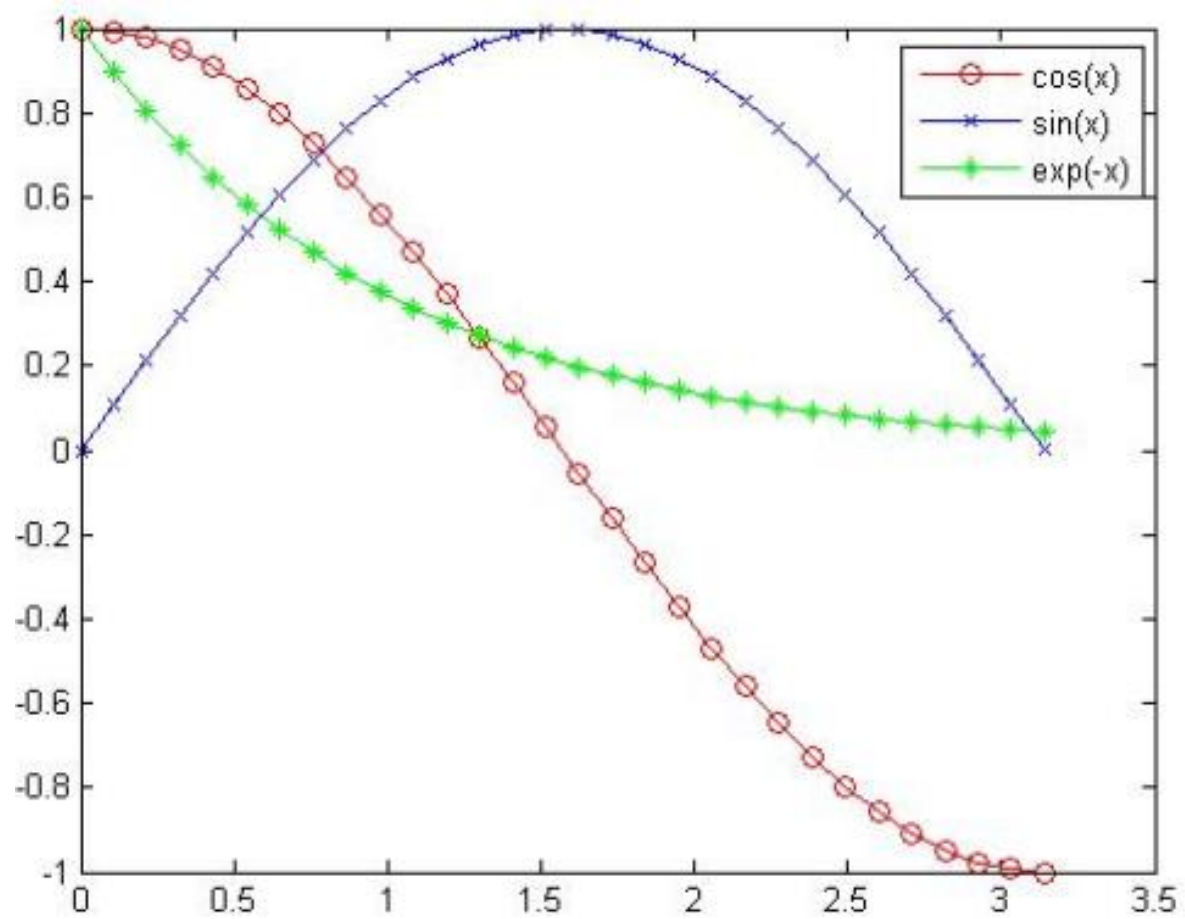


# 4. Graphique multiple 2

Une autre méthode pour tracer plusieurs courbes sur un même graphique consiste à activer la commande `hold on`, ce qui stipule à Matlab de dessiner les graphiques les uns sur les autres.

```
% — Graphique multiple 2 — %  
  
figure; % Nouvelle fenêtre  
  
x = linspace(0,pi,30); % Fonction du temps  
  
hold on % Maintient du graphique  
plot(x,cos(x),'o-r') % Tracer points ronds reliés en rouge  
plot(x,sin(x),'x-b') % Tracer points croix reliés en vert  
plot(x,exp(-x),'*-g') % Tracer points étoiles reliés en bleu  
legend('cos(x)', 'sin(x)', 'exp(-x)') % Légende sur graphique
```

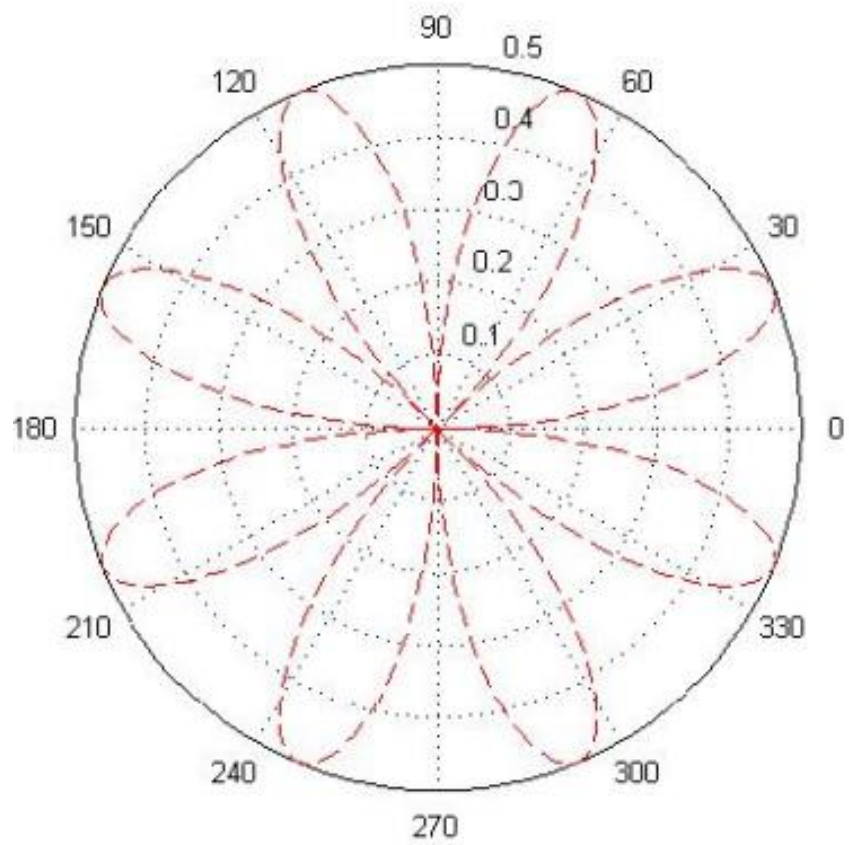
Il est à noter qu'à chaque fois `hold on` est activé, il faut la désactiver à la fin par `hold off`



# 5. Graphique polaire

Il est aussi possible de tracer des graphiques polaires en Matlab.

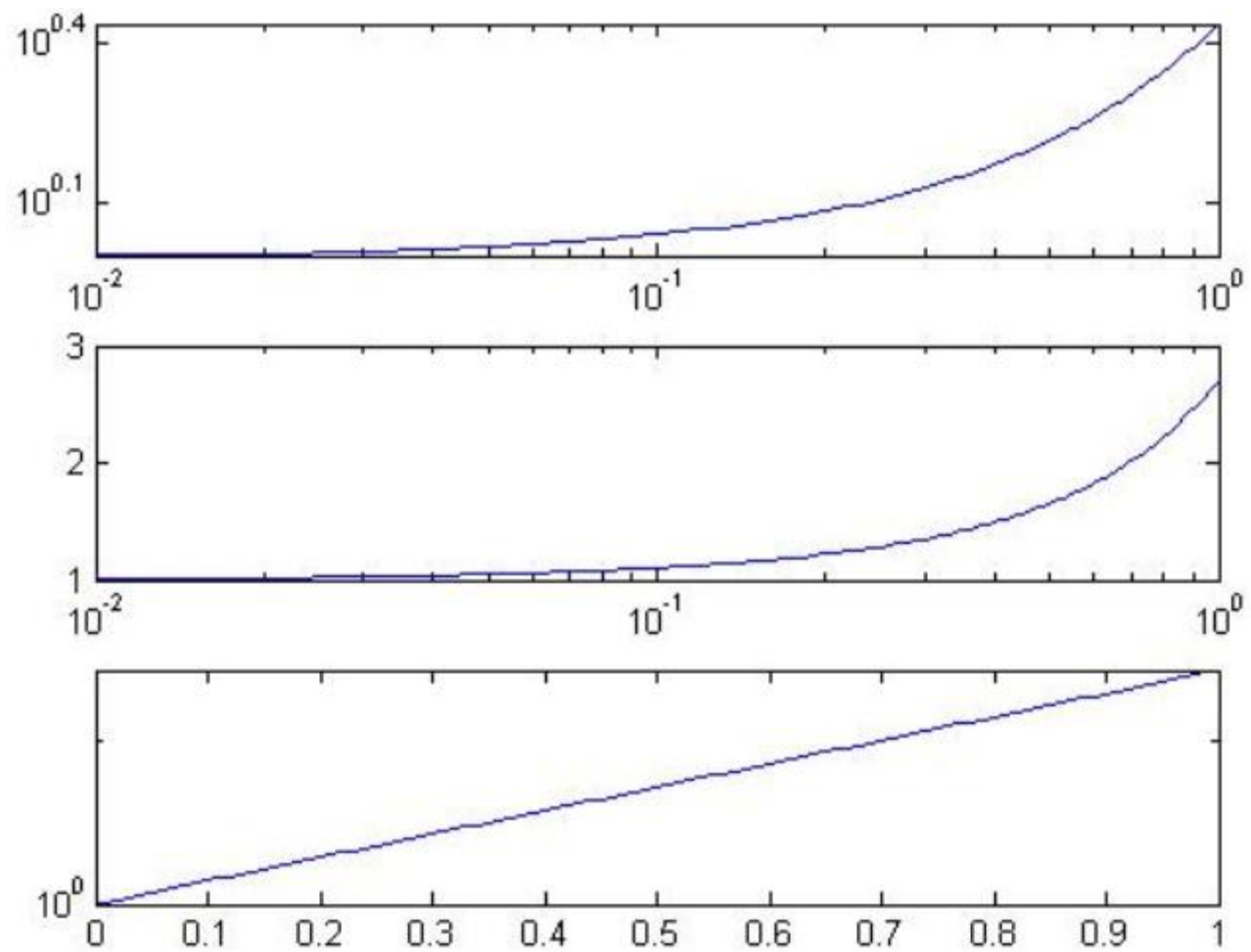
```
% — Graphique polaire ————— %  
  
figure; % Nouvelle fenêtre  
  
t = 0 :.01 :2*pi; % Échelle du temps  
polar(t,sin(2*t).*cos(2*t),'-r') % Graphique polaire en pointillé rouge
```



# 6. Graphique à échelle logarithmique

Il est aussi possible de tracer des graphiques avec un ou plusieurs axes sous forme logarithmiques en Matlab.

```
% — Graphique à échelles logarithmiques ————— %  
  
figure;           % Nouvelle fenêtre  
  
x = 0 :0.01 :1;  
subplot(3,1,1);  % Zone supérieure  
loglog(x,exp(x)); % Echelle logarithmique deux axes  
  
subplot(3,1,2)   % Zone centrale  
semilogx(x,exp(x)) % Echelle logarithmique sur l'axe Ox  
  
subplot(3,1,3)   % Zone inférieure  
semilogy(x,exp(x)) % Echelle logarithmique sur l'axe Oy
```





Il est aussi possible de tracer des graphiques à trois dimensions en Matlab. Voici quelques exemples de commandes que vous pouvez utiliser.

```
>> help plot3
```

```
PLOT3 Plot lines and points in 3-D space.
```

```
PLOT3() is a three-dimensional analogue of PLOT().
```

```
PLOT3(x,y,z), where x, y and z are three vectors of the same length,  
plots a line in 3-space through the points whose coordinates are the  
elements of x, y and z.
```

>> help mesh

MESH 3-D mesh surface.

MESH(X,Y,Z,C) plots the colored parametric mesh defined by four matrix arguments. The view point is specified by VIEW. The axis labels are determined by the range of X, Y and Z, or by the current setting of AXIS. The color scaling is determined by the range of C, or by the current setting of CAXIS. The scaled color values are used as indices into the current COLORMAP.

>> help surf

SURF 3-D colored surface.

SURF(X,Y,Z,C) plots the colored parametric surface defined by four matrix arguments. The view point is specified by VIEW. The axis labels are determined by the range of X, Y and Z, or by the current setting of AXIS. The color scaling is determined by the range of C, or by the current setting of CAXIS. The scaled color values are used as indices into the current COLORMAP. The shading model is set by SHADING.

Merci

Pour

Votre Attention