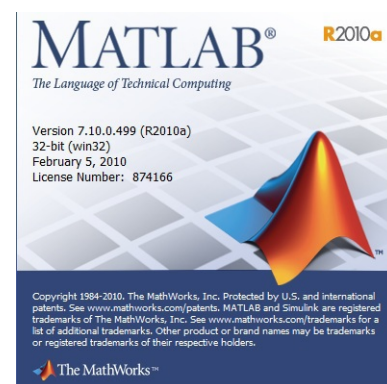


# Chapitre 02 : Les graphiques et la visualisation des données en MATLAB

*Université de M'sila*



Dr. Chouder. R

M.C.B. université de M'sila

# Table des matières



<b>Introduction</b>	<b>3</b>
<b>I - Graphiques 2D</b>	<b>4</b>

# Introduction



MATLAB fournit une grande variété de techniques pour l'affichage des données sous forme graphique.

Des outils interactifs vous permettent de manipuler les graphiques et révéler la plupart des informations contenues dans les données.

Le processus de visualisation des données implique une série d'opérations.

Ce chapitre fournit une vue générale sur les possibilités que fournit Matlab pour la représentation graphiques.



# Graphiques 2D

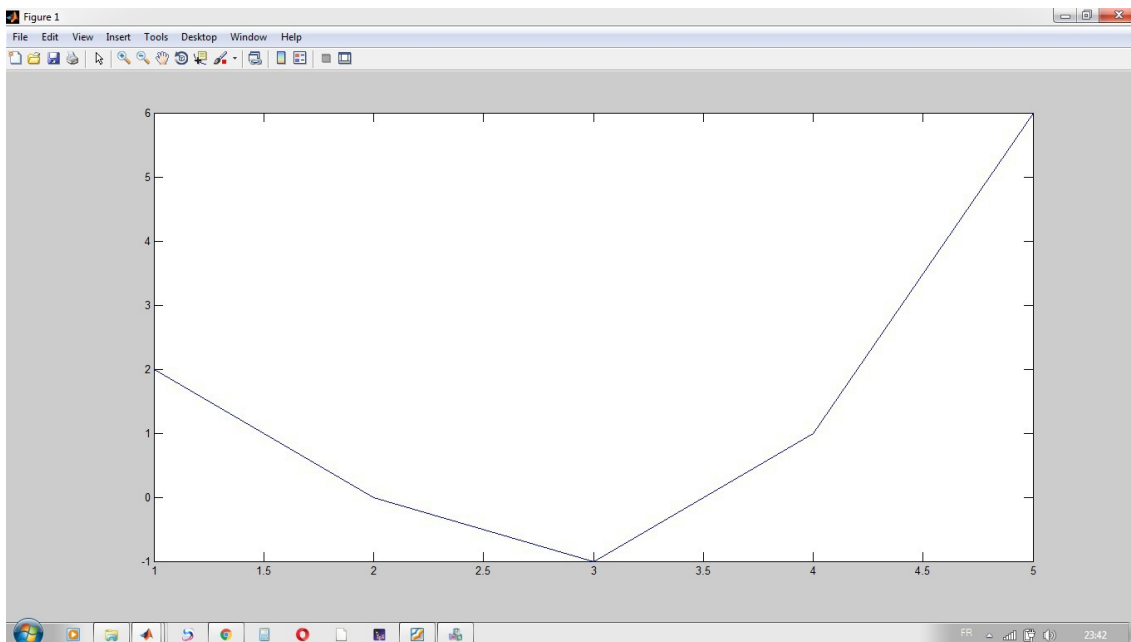
## La Fonction plot

C'est la fonction la plus utilisée dans la représentation graphique 2D. La fonction `plot` permet de dessiner une ou plusieurs courbes avec plusieurs motifs.

La commande `plot(x,f)` permet de tracer les valeurs de  $f$  en fonction de valeurs de  $x$ . En toute rigueur, il faut définir auparavant les valeurs de  $x$  et  $f$  qui doivent être de même dimension.

### Exemple

```
>> x=[1,2,3,4,5];  
>> f=[2,0,-1,1,6];  
>> plot(x,f)
```



Au graphe obtenu, on peut ajouter la légende des axes en exécutant les commandes :

```
>> xlabel('abscisse');  
>> ylabel('ordonnee');
```

On peut aussi donner un titre à la courbe à l'aide de la commande :

```
>> title ('Exemple de graphique');
```

Et on peut aussi ajouter un quadrillage :

```
>> grid on
```

On peut utiliser la commande `grid on` ou tout simplement `grid`. Pour enlever le quadrillage, on utilise la commande :

```
>> grid off
```

### **Exemple**

---

```
>> x=-2:0.1:2;
```

```
>> f=exp(-x.^2);
```

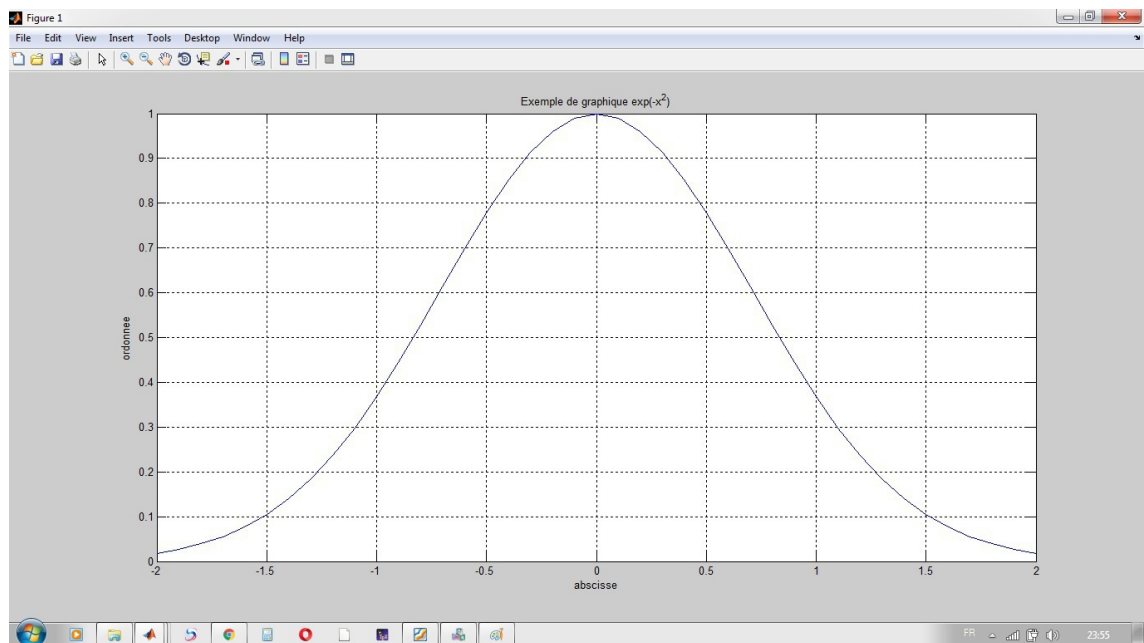
```
>> plot(x,f)
```

```
>> xlabel('abscisse');
```

```
>> ylabel('ordonnee');
```

```
>> title ('Exemple de graphique exp(-x^2)');
```

```
>> grid
```



On peut ajuster les axes avec la commande `axis([xmin xmax ymin ymax])`, où `Xmin`, `Xmax` : les graduations minimales et maximales de l'axe des abscisses et `Ymin`, `Ymax` : les graduations minimales et maximales de l'axe des ordonnées.

La commande : `axis('square')` présente le graphe dans un carré au lieu du rectangle habituel.

Pour utiliser des couleurs différentes on peut utiliser la syntaxe générale de la fonction `plot` : `plot(x,`

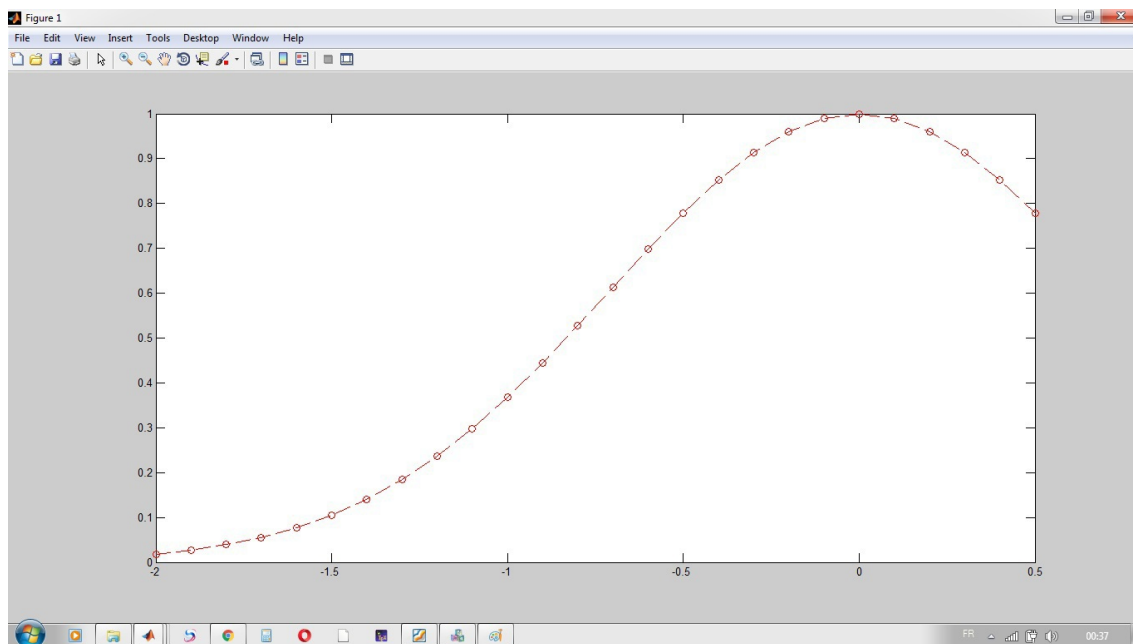
```
f, 'c+:' )
```

Le tableau suivant donne les différents styles, motifs et couleurs supportés par la commande plot :

Couleurs		Marqueurs		Styles de lignes	
<i>y ou yellow</i>	<i>jaune</i>	.	<i>point</i>	-	<i>ligne solide</i>
<i>m ou magenta</i>	<i>magenta</i>	o	<i>cercle</i>	:	<i>pointillée</i>
<i>c ou cyan</i>	<i>cyan</i>	*	<i>étoile</i>	-.	<i>Trait d'union-point</i>
<i>r ou red</i>	<i>rouge</i>	+	<i>plus</i>	--	<i>coupée</i>
<i>g ou green</i>	<i>vert</i>	d	<i>diamant</i>		
<i>b ou blue</i>	<i>bleu</i>	s	<i>carré</i>		
<i>k ou black</i>	<i>noir</i>	v	<i>Triangle (bas)</i>		
<i>w ou white</i>	<i>blanc</i>	^	<i>Triangle (haut)</i>		

### Exemple

```
>> x=-2:0.1:2;
>> f=exp(-x.^2);
>> plot(x,f,'r--o')
>> axis([-2 0.5 0 1])
```



## Plusieurs courbes dans une même figure

Dans certains cas, on est amené à tracer plusieurs courbes dans une même figure. Dans MATLAB, ceci peut se faire des deux manières.

La première est d'utiliser une seule instruction `plot` : `plot(x,f1,x,f2,x,f3,x,f4)`.

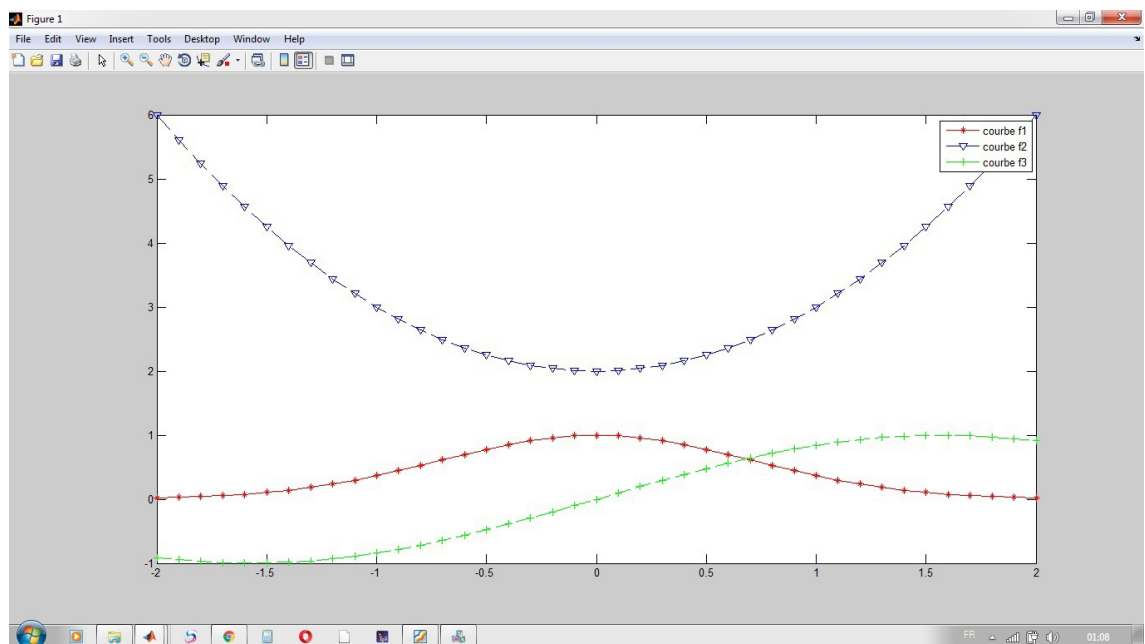
On peut ajouter une légende à cette courbe, en utilisant la commande suivante :

```
>> legend('courbe f1', 'courbe f2', 'courbe f3', 'courbe f4')
```



### Exemple : Plusieurs courbes dans une même figure

```
>> x=-2:0.1:2;
>> f1=exp(-x.^2);
>> f2=x.^2+2;
>> f3=sin(x);
>> plot(x,f1,'r*-',x,f2,'bv--',x,f3,'g+--')
>> legend('courbe f1', 'courbe f2', 'courbe f3')
```



## Plusieurs courbes dans une même figure

La deuxième manière de tracer plusieurs courbes dans une même figure consiste à tracer les figures une par une.

L'utilisation de la commande `plot` plusieurs fois va utiliser la même fenêtre et va effacer l'ancienne courbe pour tracer la récente. Pour garder l'ancienne courbe et tracer dessus la nouvelle courbe, il faut utiliser la commande MATLAB `hold on` (l'effet de cette commande est annulé à l'aide de la commande

hold off).

### Exemple : Plusieurs courbes dans une même figure

```
>> x=-2:0.1:2;

>> f1=exp(-x.^2);

>> f2=x.^2+2;

>> f3=sin(x);

>> hold on

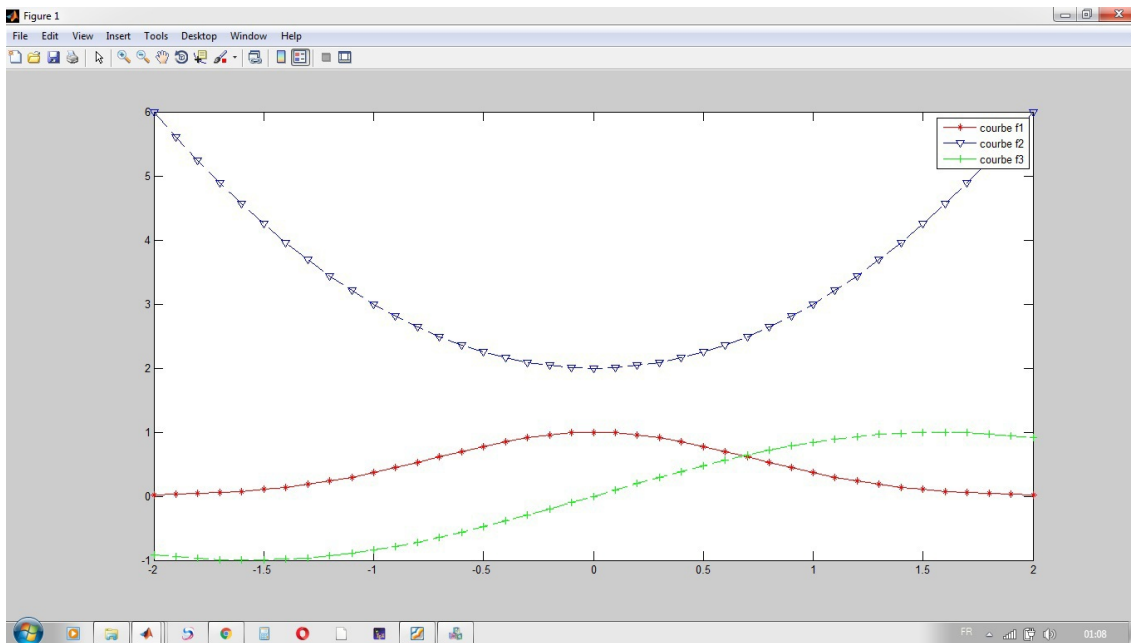
>> plot(x,f1,'r*-')

>> plot(x,f2,'bv--')

>> plot(x,f3,'g+--')

>> legend('courbe f1', 'courbe f2', 'courbe f3')

>> hold off
```



On peut tracer les courbes sur des fenêtres graphiques séparées à condition de créer ces fenêtres à l'aide de la commande *figure*.

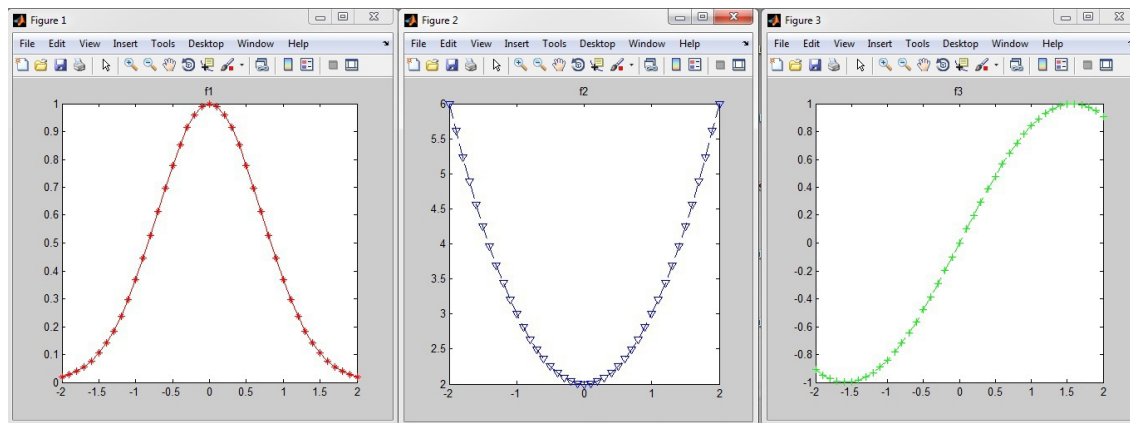
### Exemple

```
>> figure(1),plot(x,f1,'r*-'),title('f1')

>> figure(2),plot(x,f2,'bv--'),title('f2')

>> figure(3),plot(x,f3,'g+--'),title('f3')
```





### Plusieurs courbes dans la même fenêtre

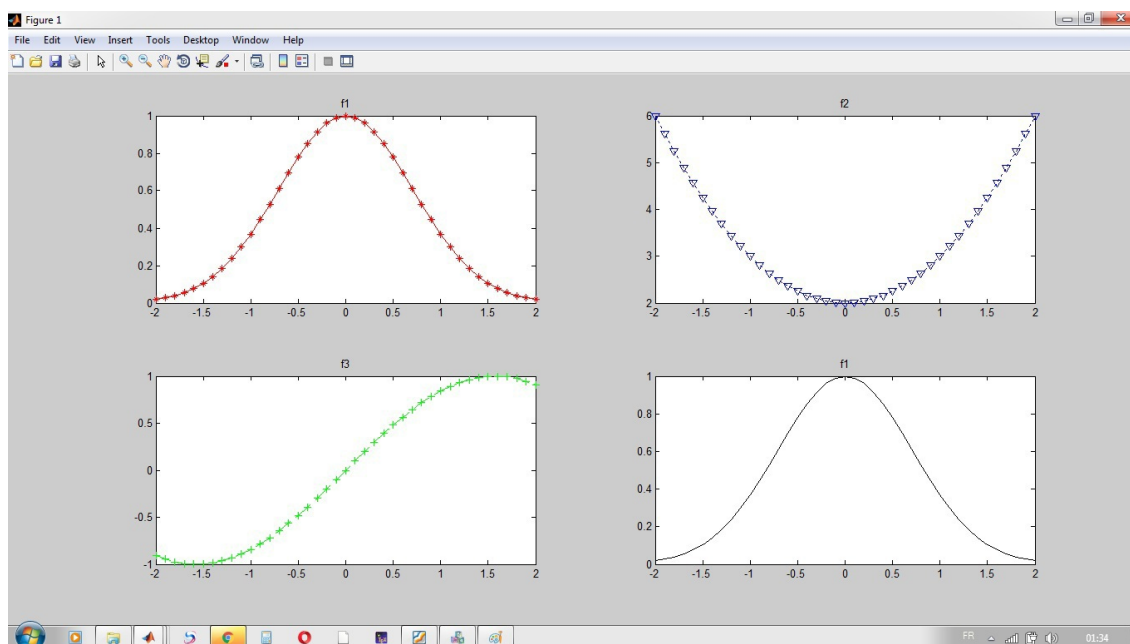
Si on veut tracer plusieurs courbes, dans la même fenêtre, mais dans des cadres séparés, alors on utilise la fonction MATLAB `subplot`.

La fonction `subplot(m,n,p)` ou `subplot(mnp)` subdivise la fenêtre graphique en  $m$  lignes et  $n$  colonnes cadres séparés. Le cadre est choisi de telle sorte que  $p=1$  correspond à  $(m=1,n=1)$ ,  $p=2 \rightarrow (m=1,n=2)$ , ....



### Exemple : Plusieurs courbes dans la même fenêtre

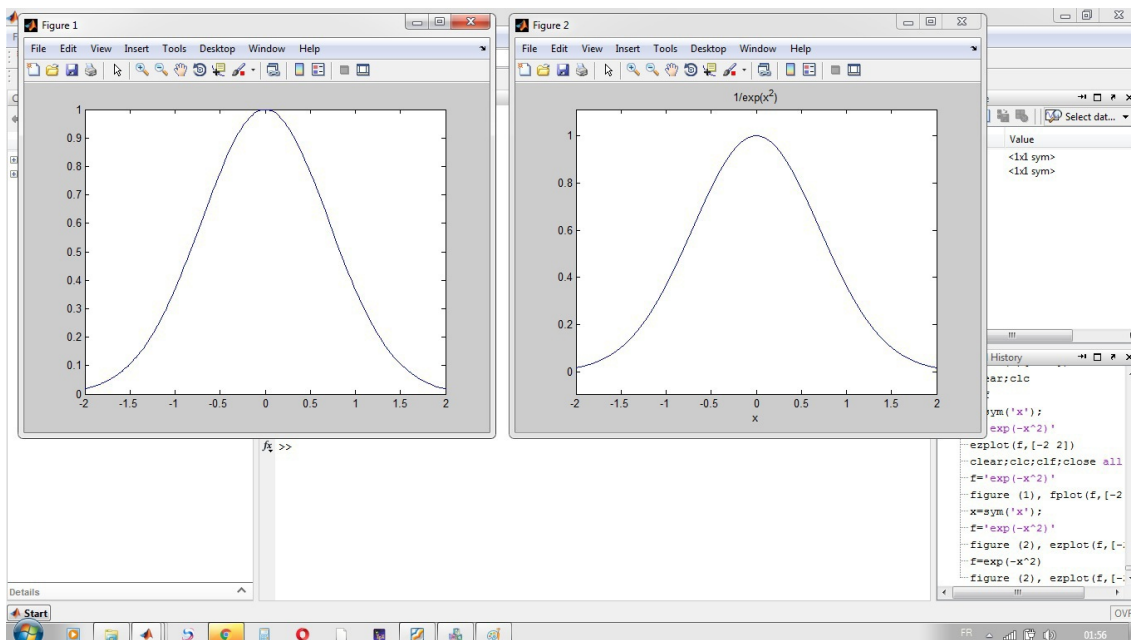
```
>> subplot(221),plot(x,f1,'r*-'),title('f1')
>> subplot(222),plot(x,f2,'bv:'),title('f2')
>> subplot(223),plot(x,f3,'g+--'),title('f3')
>> subplot(2,2,3),plot(x,f3,'g+--'),title('f3')
>> subplot(2,2,4),plot(x,f1,'k'),title('f1')
```



## Les fonctions *fplot*, *ezplot* :

On peut aussi utiliser les fonctions *fplot* et *ezplot* pour tracer une fonction :

```
>> f='exp(-x^2) '
f =
exp(-x^2)
>> figure (1), fplot(f,[-2 2])
>> x=sym('x');
>> f=exp(-x^2)
f =
1/exp(x^2)
>> figure (1), ezplot(f,[-2 2])
```



La commande *close all* permet de fermer toutes les fenêtres graphiques.

La commande *close(n)* permet de fermer la fenêtre graphique n.

Pour obtenir la liste des fonctions de représentation graphique 2D disponibles dans MATLAB, tapez :

```
>> help graph2d
```