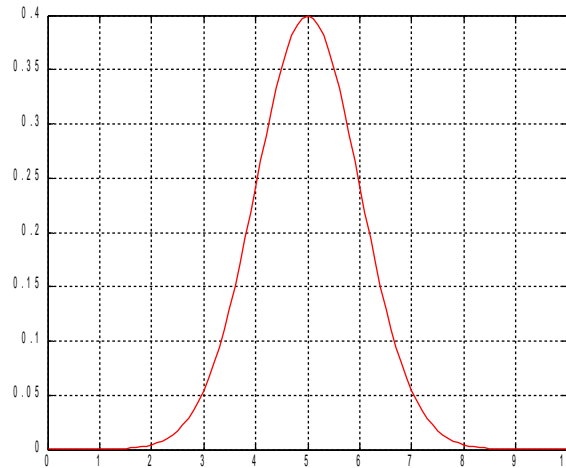


Tracé d'une fonction

Tracé d'une fonction :

- exprimer les valeurs prises par l'abscisse sous forme d'une liste de valeurs
- calculer les valeurs de la fonction en utilisant directement la liste des valeurs (attention aux carrés et multiplications entre éléments de type liste!)



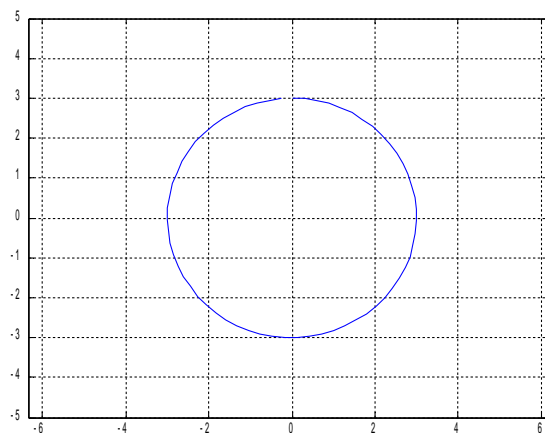
Exemple : tracé d'une gaussienne

```
X = [ 0.:1:10];
sigma = 1; mu = 5;
Gauss = exp( -(X - mu) .*(X - mu) / (2*sigma^2) ) / (sqrt(2*pi) * sigma);
plot( X , Gauss,'r')
grid on
```

Tracé d'une courbe paramétrique

Chacun des points d'une courbe paramétrique est décrit par un couple de valeurs (x, y) représentant l'abscisse et l'ordonnée du point. L'abscisse et l'ordonnée sont fonction d'un paramètre t qui parcourt un ensemble de valeurs.

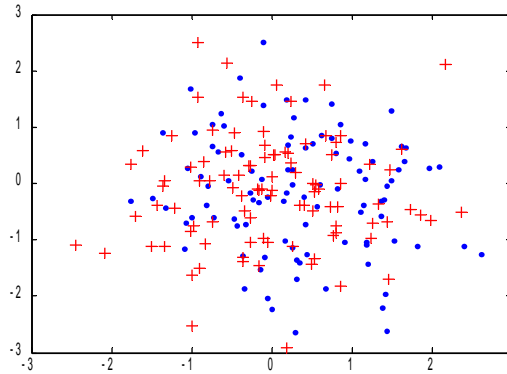
Exemple : le cercle est décrit par $x(t)=r \cos t$ et $y(t)=r \sin t$ avec $t=[0, 2\pi]$



```
t = [0 :0.1: 2*pi]; radius = 3;
X = radius * sin(t); Y= radius * cos(t);
plot( X , Y , 'b')
axis([-5 5 -5 5],'equal')
```

Tracé d'un nuage de points 2D

Chacun des points d'un nuage est décrit par un couple de valeurs (x, y) représentant l'abscisse et l'ordonnée. Le tracé d'un nuage de points ne diffère du tracé paramétrique que par la présentation (pas de lien implicite entre les points).

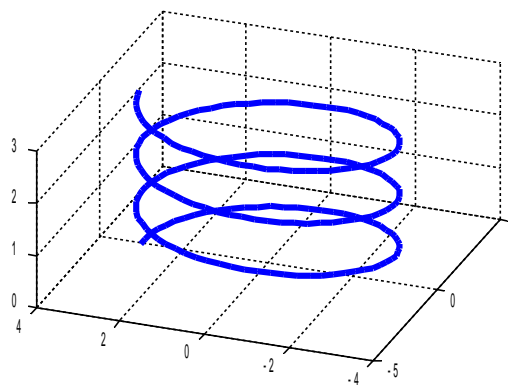


```
Abcisse_Nuage1 = randn(1,100); Ordonnee_Nuage1= randn(1,100);
plot ( Abcisse_Nuage1 , Ordonnee_Nuage1 , '.' )
hold on
Nuage2=randn(2,100);
plot( Nuage2( 1 , : ) , Nuage2( 2 , : ) , '+r' )
hold off
```

Tracé d'une courbe 3D

La courbe 3D a généralement une expression paramétrique $x(t), y(t), z(t)$ avec t paramètre.

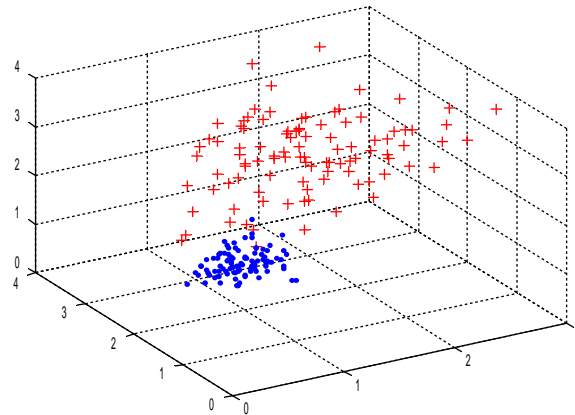
Exemple: l'hélice est une courbe ayant pour base (x, y) le cercle et dont l'élévation est proportionnelle au paramètre de description du cercle (l'angle θ).



```
theta= [0 :0.1: 6*pi] ; % 3 spires d'hélice
radius = 3 ;
pas = 1 / (2*pi) ;
X = radius * sin(theta) ; Y= radius * cos(theta) ; Z = pas * theta ;
plot3(X , Y , Z , 'b')
grid on
```

Tracé d'un nuage de points 3D

Le principe est le même que pour un nuage 2D: les points possèdent 3 coordonnées (x, y, z) représentés par des marqueurs non liés



```
T1=ones(1,100);
% Premier nuage de valeurs
A1 = 0.2 * randn(3,100) + [1*T1 ; 2*T1 ; 1*T1] ;
plot3( A1(1, :), A1(2, :), A1(3, :), 'b')
hold on
% Deuxième nuage de valeurs
A2 = 0.6 * randn(3,100) + [2*T1 ; 3*T1 ; 2*T1] ;
plot3( A2(1, :), A2(2, :), A2(3, :), '+r')
grid on ; axis([0 3 0 4 0 4])
```

Tracé d'une surface

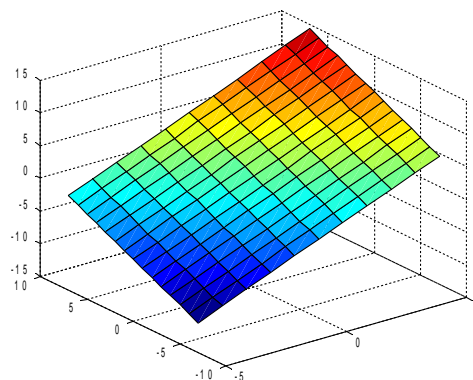
Une surface est une application de (x, y) vers z de la forme $z = f(x, y)$.

Exemple: le plan est décrit par l'équation $z = ax + by + z_0$

La représentation est calculée pour un *maillage* des valeurs de l'espace de départ; le tracé de ce maillage forme une grille rectangulaire ou carrée. Chaque point du maillage est caractérisé par le couple des coordonnées (x, y) .

La fonction bidimensionnelle est rendu visible par la projection de cette grille sur sa surface (courbes *iso-x* et *iso-y*).

Exemple du plan :



```
a = 1.5 ; b = .7 ; z0 = 2 ;
X = [-5 : .5 : 5] ;
Y = [-7 : .3 : 7] ;
% génération de la grille X par produit extérieur
GrilleX = ones(length(Y),1) * X ;
GrilleY = Y' * ones(1, length(X)) ;
Z = a * GrilleX + b * GrilleY + z0 * ones(length(Y), length(X));
surf(Z)
```

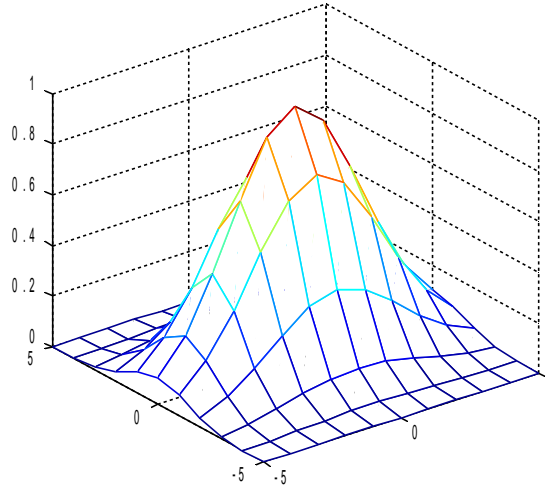
✓ Utiliser meshgrid pour générer directement les valeurs de grille.

Tracé d'une fonction séparable

Le tracé est simplifié pour les fonctions séparables, c'est à dire celle qui sont le produit d'une fonction en x par une fonction en y .

$$f(x, y) = f_x(x) \times f_y(y)$$

Exemple : fonction du type gaussienne bidimensionnelle.

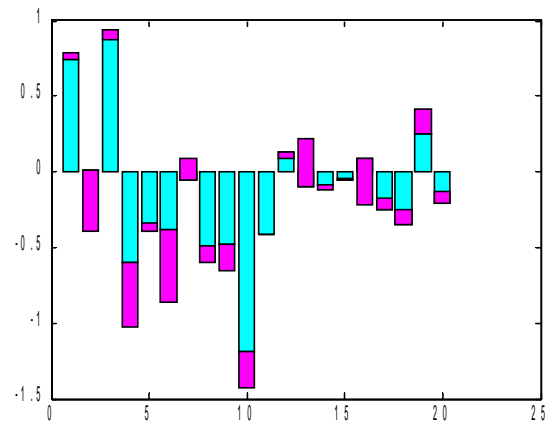
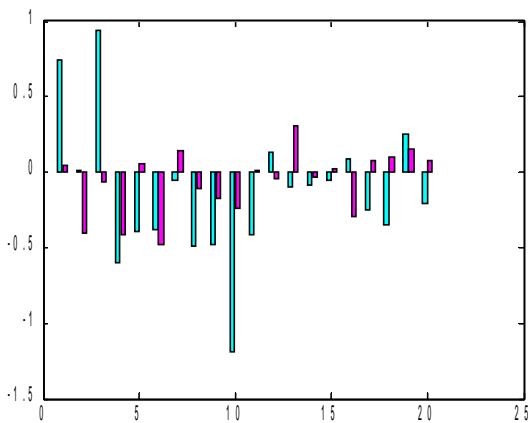


```
X = [-5 : 5]; Y = [-5 : 5];
GaussX = exp(-X.^2 * sigmaX^2);
GaussXY = GaussX * GaussY;
mesh(X, Y, GaussXY)
```

```
sigmaX = 0.5; sigmaY = 0.3;
GaussY = exp(-Y.^2 * sigmaY^2);
```

Tracé type barres

Tracé classique de représentation de résultats statistiques.



```
nb_barres=20;
A1 = 0.5*randn(nb_barres,1);
A2 = rand(nb_barres,1) - 0.5;
A = [A1 A2];
% Deux mises en forme différentes
bar(A)
colormap(cool)
```

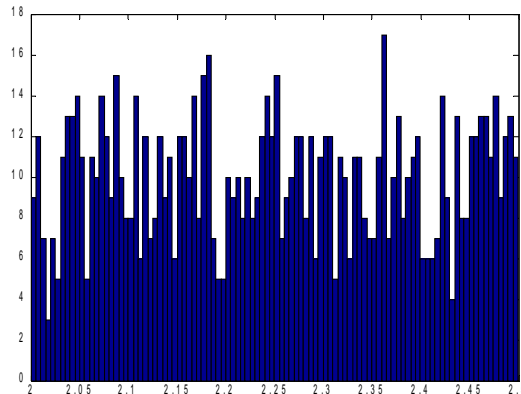
```
bar(A,'stacked')
colormap(cool)
```

Histogramme d'une variable

L'histogramme d'une variable est le dénombrement des valeurs prises par la variable à caractère aléatoire. Le graphe de l'histogramme est généralement du type *barres*.

Matlab propose une fonction de calcul et de tracé de l'histogramme : par défaut, l'espace de variation est découpé en 10 segments et l'histogramme comprend 10 raies. Le nombre de raies est modifiable, en le passant en paramètre dans la fonction. `hist` .

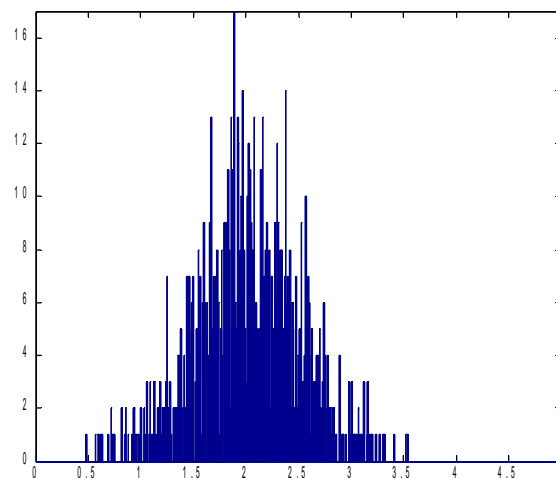
```
A = 0.5*rand(1,1000) + 2 ; %tirage de 1000 valeurs aléatoires comprises entre 2 et 2.5
hist ( A,100)             % histogramme sur 100 raies
```



```
std(A) % calcul de l'écart-type
0.1448
mean(A) % calcul de la moyenne
2.2524
```

Autre méthode : l'histogramme est un graphique du type "barres" . L'histogramme est calculé par la fonction `hist` avec retour (pas d'affichage si la fonction a une valeur de retour). Il est bien évidemment possible d'écrire soi-même la fonction `hist` !

```
A = 0.5*randn(1,1000) + 2 ;
X = [ 0:.01:5];
histo = hist(A,X);
bar(X,histo)
axis([min(X) max(X) 0 max(histo)])
```



```
std(A) : 0.4955
mean(A) : 2.021(valeurs différentes pour ce nouveau tirage)
```

* voir `randtool` pour la génération de variables statistiques.