



Chapitre 4 : Dessiner des Graphes avec Matlab

Université Alger I, Dept MI

2° année Maths, Semestre 3, 2016

Matière : Outils de Programmation 2

Contact : fodil.laib@hotmail.com

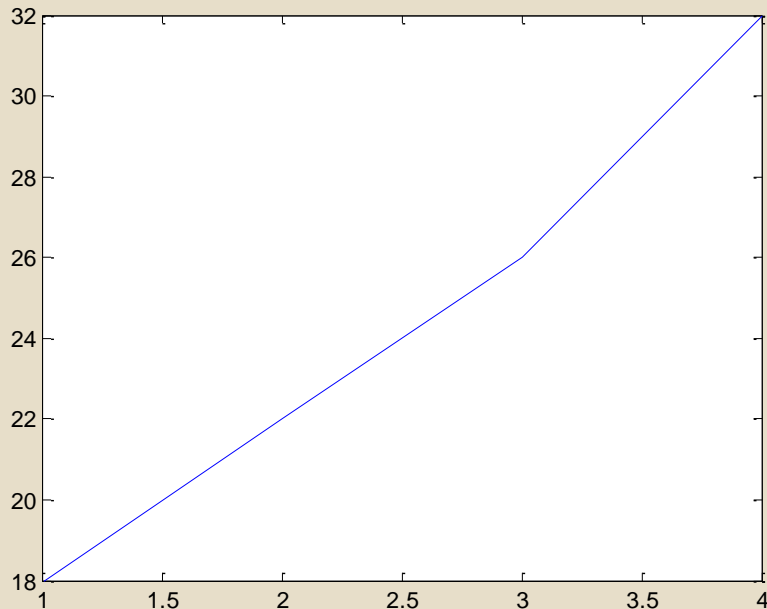
La Fonction *plot*

plot permet de dessiner une ou plusieurs courbes avec plusieurs motifs.

Cas d'un vecteur (valeurs des ordonnées) :

```
>> H = [18 22 26 32];
```

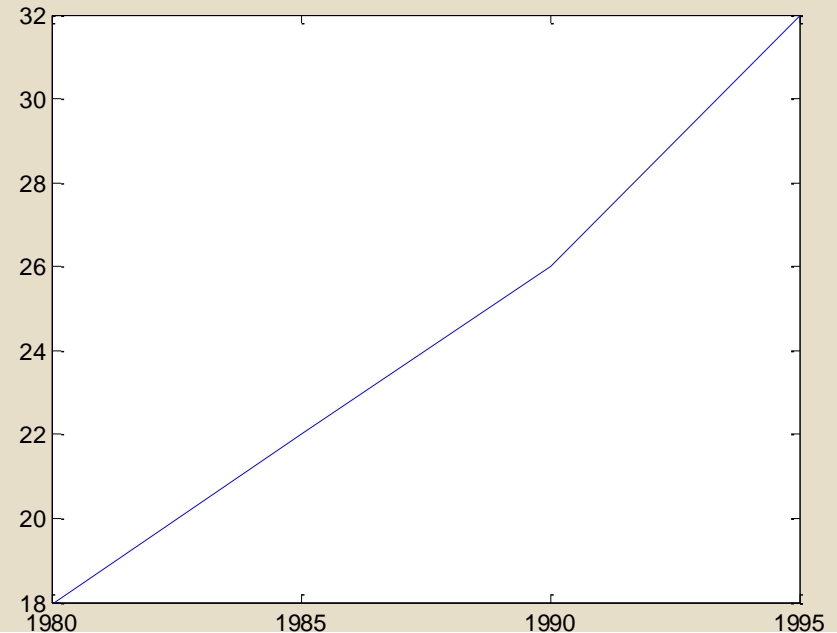
```
>> plot(H)
```



Cas de 2 vecteurs (valeurs des abscisses et des ordonnées) :

```
>> A = [1980 1985 1990 1995];
```

```
>> plot(A, H)
```



Avec les commandes *grid*, *title*, *xlabel*, *ylabel* on ajoute respectivement une grille, un titre, un label pour les x et un label pour les y.

...

On peut ajuster les axes avec la commande `axis([xmin xmax ymin ymax])`

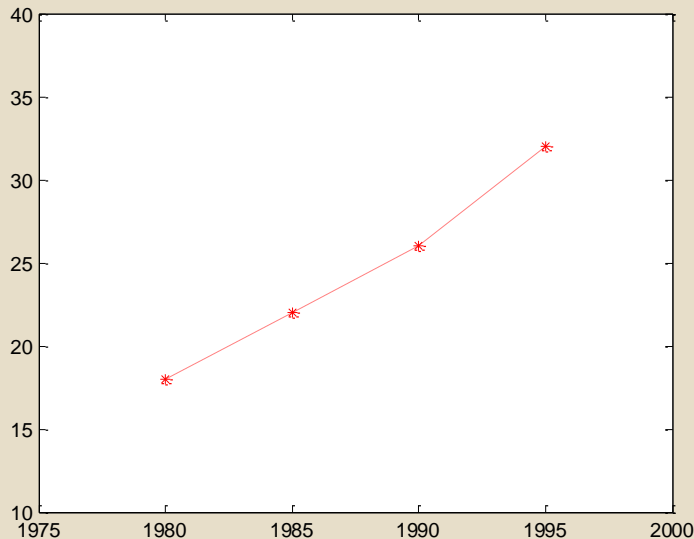
Changer la couleur avec les lettres **r**, **b**, **g** ...
pour red, blue, green, ...

Marquer les points par *****, **+**, **o**, **x**, ...

Type de trait : **-**, **--**, **-.**, **:**,

```
>> plot(A, H, 'r*')
```

```
>> axis([1975 2000 10 40])
```



Cas d'une matrice :

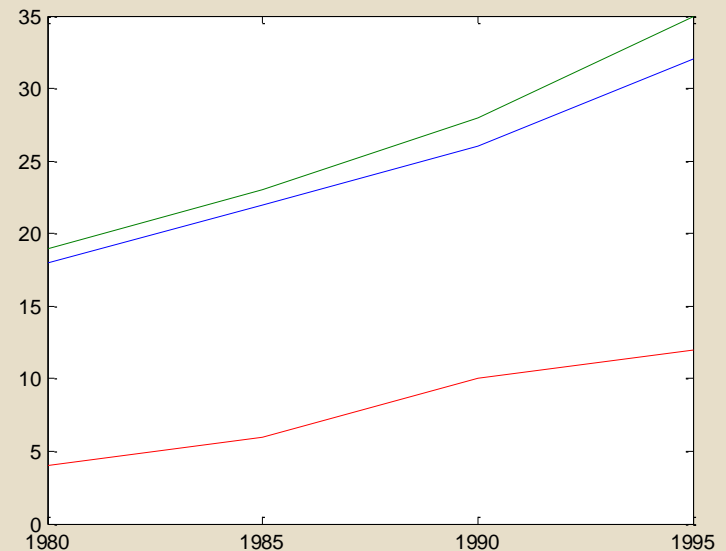
Chaque colonne sera représentée par une courbe :

```
>> HH = [ H ; 19 23 28 35 ; 4 6 10 12 ]'
```

HH =

18	19	4
22	23	6
26	28	10
32	35	12

```
>> plot(A, HH)
```



Cas de courbes n'ayant pas les mêmes abscisses :

Définissons les vecteurs :

```
>> x1 = [10 20 30 40];
```

```
>> y1 = [ 500 350 600 430];
```

```
>> x2= [8 18 32 45 ];
```

```
>> y2 = [ 150 180 210 200];
```

Déssinons les courbes :

```
>> plot(x1,y1,'-r*',x2,y2,':b+')
```

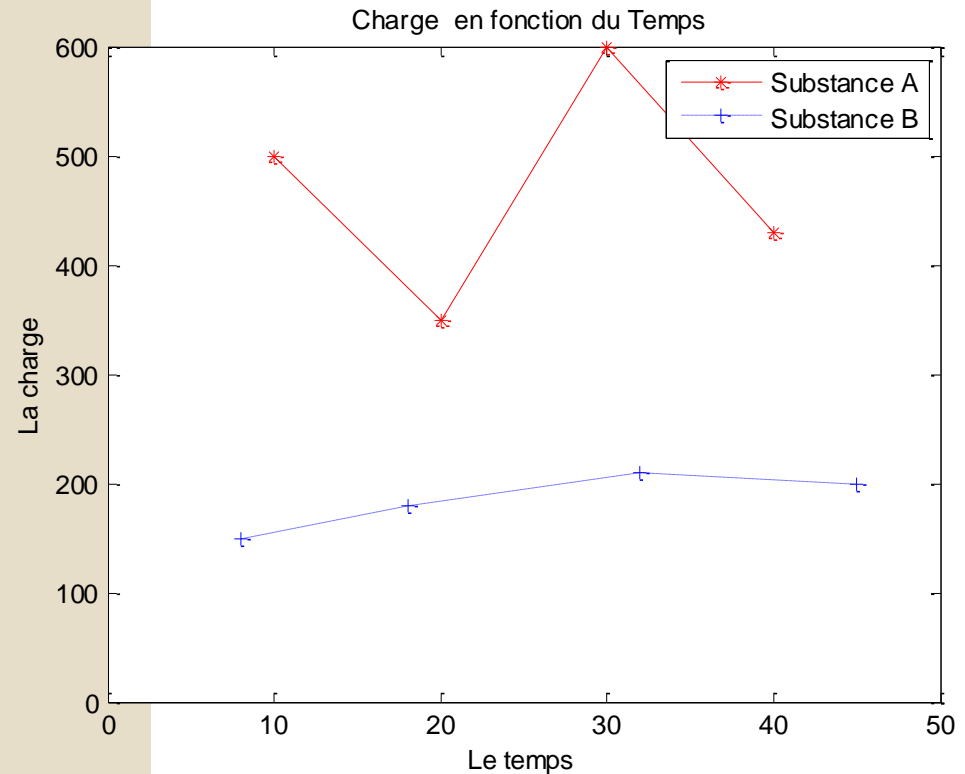
Puis,

```
>> hold on
```

```
>> axis([0 50 0 600])
```

```
>> xlabel('Le temps'), ylabel('La charge'),  
title('Charge en fonction du Temps')
```

```
>> legend('Substance A','Substance B')
```



La Fonction *bar*

bar(x,y) : dessine des bars, chacune d'elle correspond à un couple $x(i), y(i)$ des vecteurs x et y .

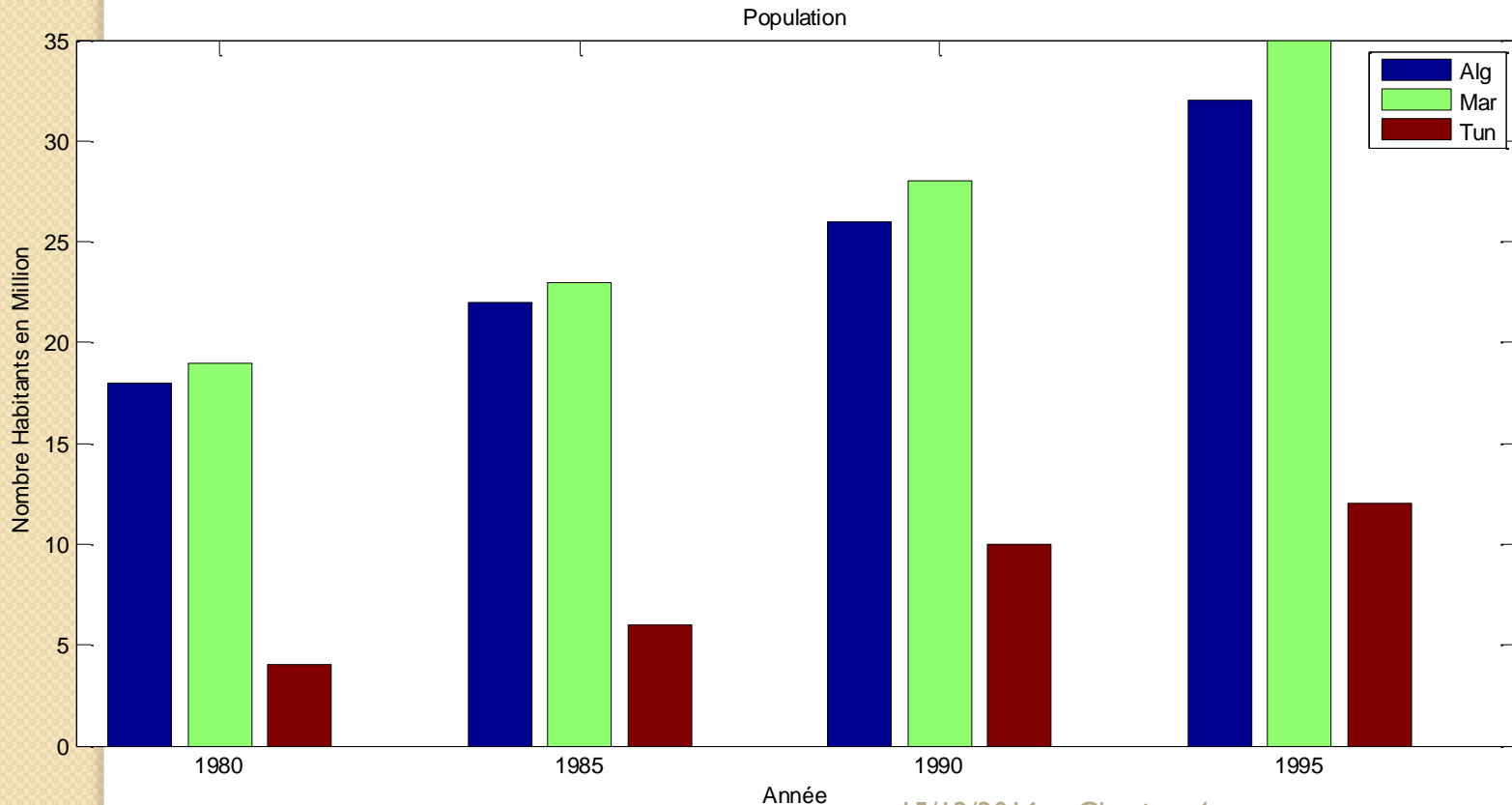
Traçons les bars

```
>> bar(A,HH)
```

```
>> hold on
```

```
>> legend('Alg','Mar','Tun')
```

```
>> title('Population'), xlabel('Année'),  
ylabel('Nombre Habitants en Million')
```



La Fonction *hist*

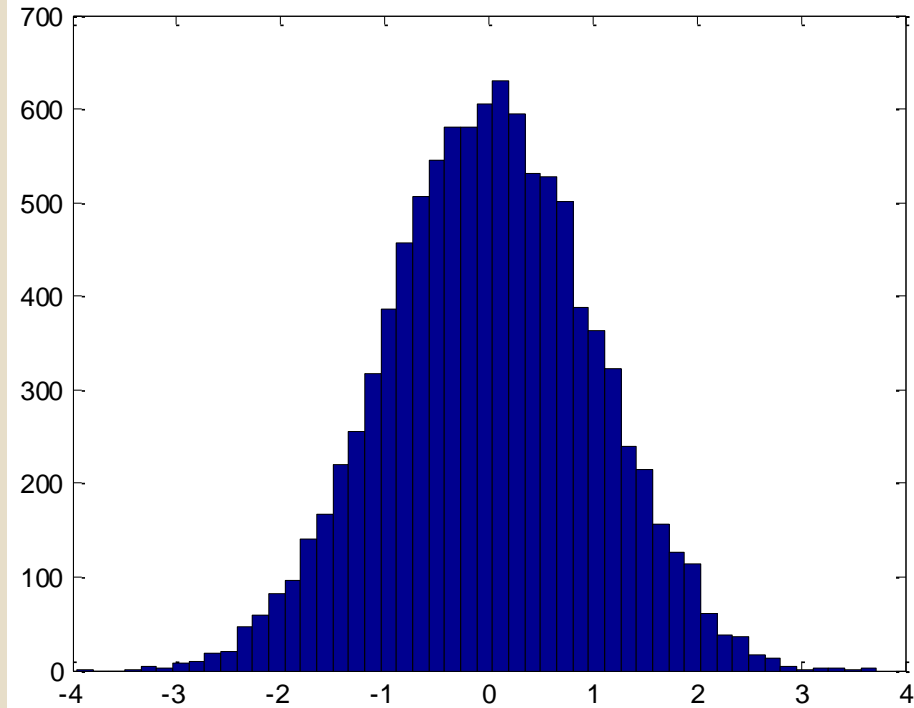
hist(y,n) : dessine un histogramme qui repartit les valeurs de *y* en *n* classes.

Créons un vecteur de nombres aléatoires suivant une loi normale :

```
>> y = randn(10000,1);
```

Dessinons l'histogramme correspondant à 50 classes :

```
>> hist(y,50)
```



La Fonction *gplot*

En théorie des graphes, un graphe est défini par des nœuds et des arcs.

La matrice d'adjacence MA montre les arcs reliant les nœuds.

Les coordonnées cartésiennes des nœuds sont données par la matrice Coord.

gplot(MA, Coord, Type) dessine le graphe correspondant

```
>> MA = [0 1 0 1 0; 1 0 1 1 0; 0 1 0 0 1; 1 1 0 0 0; 0 0 1 0 0]
```

MA =

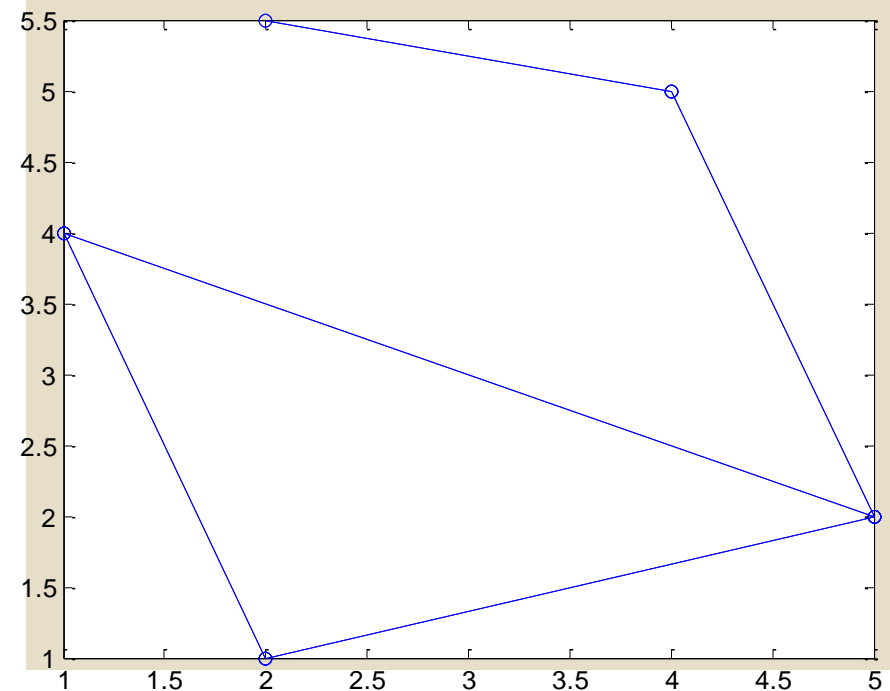
0	1	0	1	0
1	0	1	1	0
0	1	0	0	1
1	1	0	0	0
0	0	1	0	0

```
>> Coord = [2 1; 5 2; 4 5; 1 4; 1 3]
```

Coord =

2	1
5	2
4	5
1	4
1	3

```
>> gplot(MA, Coord, '-o')
```



Ajustons les axes du graphe

```
>> axis([0 6 0 6])
```

Définissons des étiquettes pour les noeuds

```
>> Etiq = ['ALGER '; 'ROME '; 'PARIS ';  
'LONDRES'; 'MADRID ']
```

Etq =

ALGER

ROME

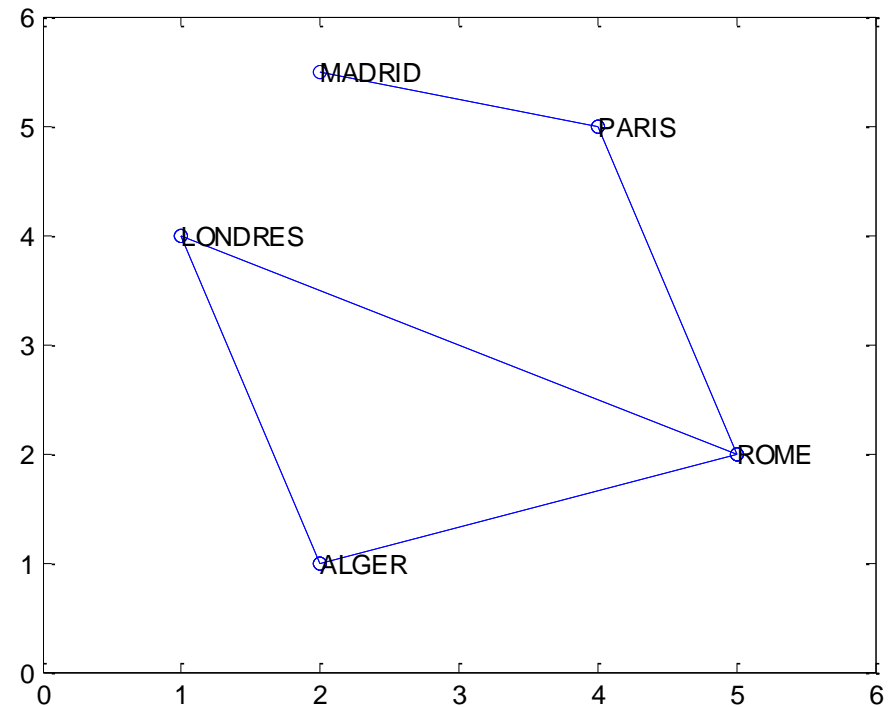
PARIS

LONDRES

MADRID

Affichons les étiquettes sur le graphe

```
>> text(Coord(:,1), Coord(:,2), Etiq)
```



Remarque :

Pour vérifier que MA est bien une matrice d'adjacence, on doit avoir

```
>> all(all(MA == MA'))
```

ans =

1

La Fonction *plot3*

plot3(x,y,z) : dessine une courbe en 3 dimensions.

Définissons les vecteurs

```
>> t = -3*pi : pi/10 : 3*pi;
```

```
>> x = 4*sin(t);
```

```
>> y = 4*cos(t);
```

```
>> z = 2*t;
```

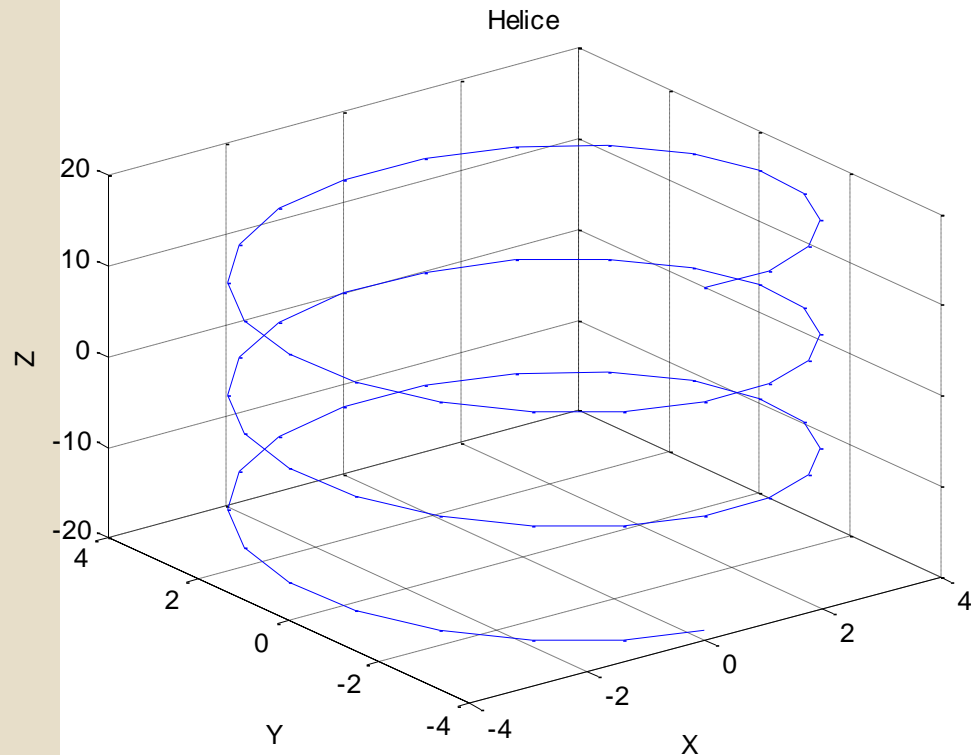
Dessinons la courbe 3D:

```
>> plot3(x,y,z)
```

Ajoutons

```
>> grid, xlabel('X'), ylabel('Y'), zlabel('Z')
```

```
>> title('Helice')
```



La Fonction *meshgrid*

$[A,B] = \text{meshgrid}(a,b)$: une fonction utilisée dans les calculs intermédiaires.

a est un vecteur à m composantes

b est un vecteur à n composantes

meshgrid renvoi les matrices A et B tel

le vecteur a est répété n fois dans A

le vecteur b est répété m fois dans B

```
>> a = 1:4
```

```
a =
```

```
1    2    3    4
```

```
>> b = 10:10:30
```

```
b =
```

```
10    20    30
```

```
>> [A,B] = meshgrid(a,b)
```

```
A =
```

```
1    2    3    4
```

```
1    2    3    4
```

```
1    2    3    4
```

```
B =
```

```
10    10    10    10
```

```
20    20    20    20
```

```
30    30    30    30
```

La Fonction *mesh*

mesh(x,y,z) : dessine la surface d'une fonction à 2 variables :

Soit :

$$z = x^2 + y^2$$

Définissons les vecteurs

```
>> x = -10 : 0.8 : 10;
```

```
>> y = x;
```

Calculons :

```
>> [X,Y] = meshgrid(x,y);
```

```
>> Z = X.^2 + Y.^2;
```

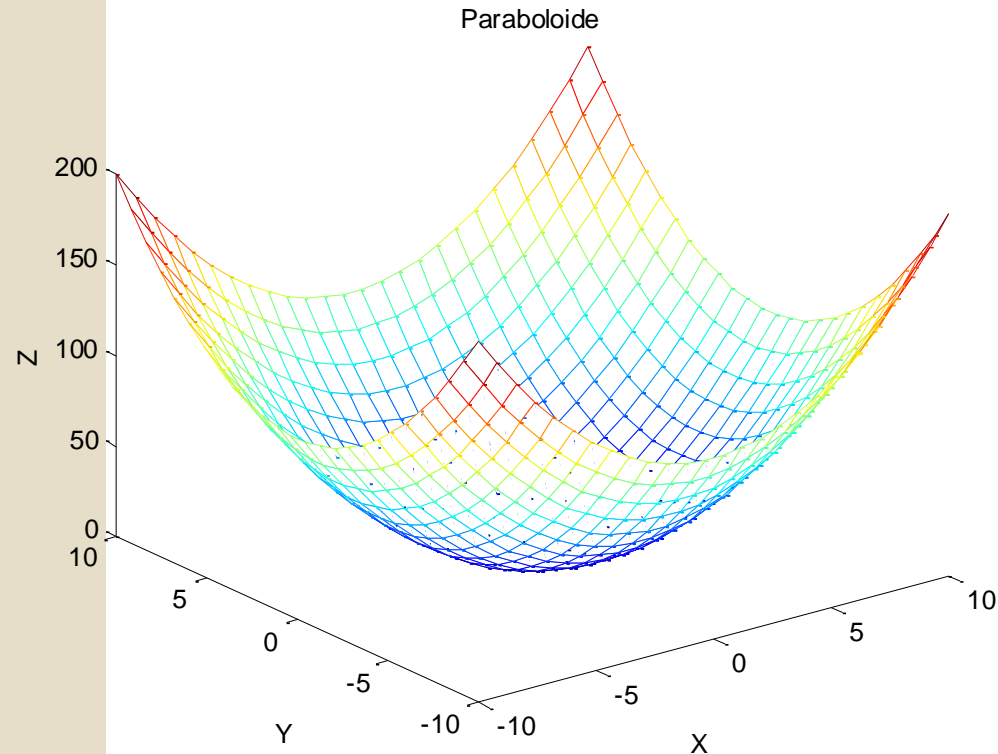
Traçons la surface :

```
>> mesh(X,Y,Z)
```

Ajoutons

```
>> grid, xlabel('X'), ylabel('Y'), zlabel('Z')
```

```
>> title('Paraboloid')
```

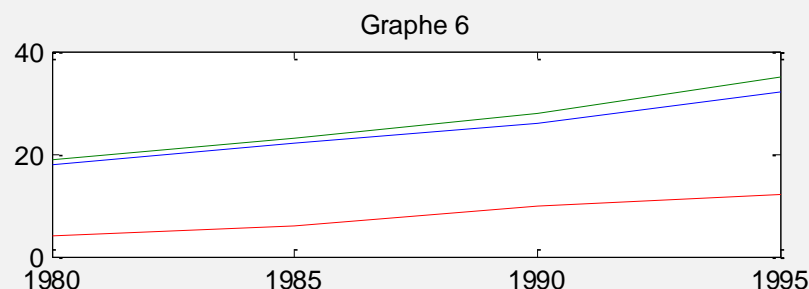
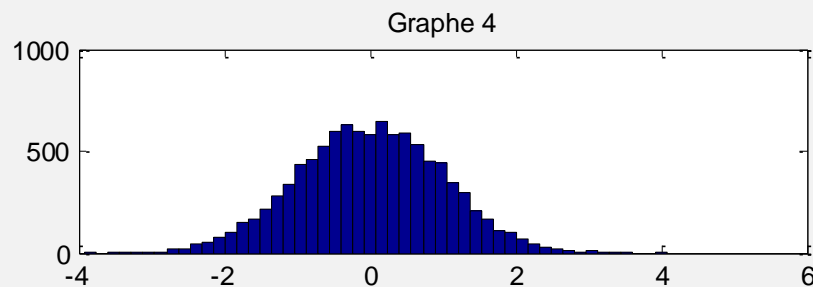
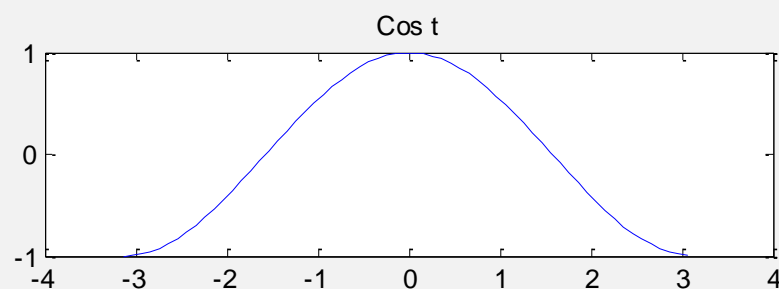
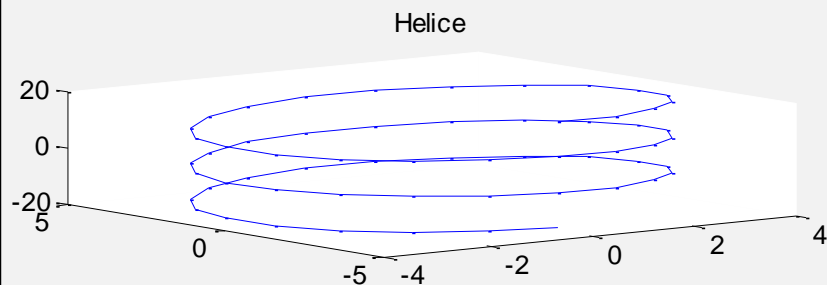
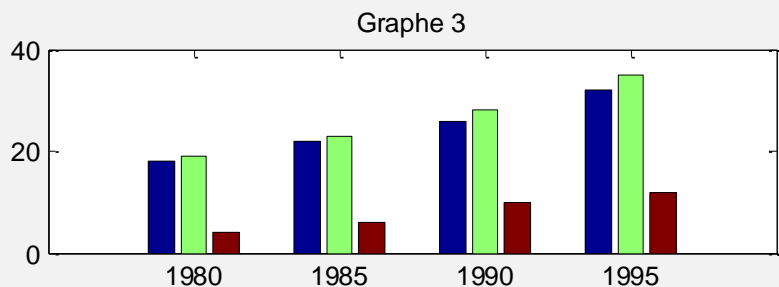
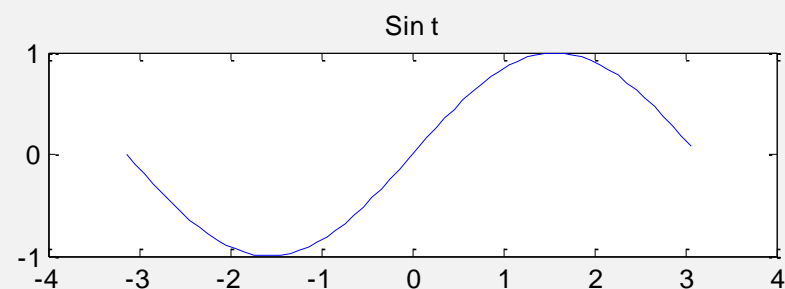


Subdiviser la Fenêtre Graphique

`subplot(n,m,i)` divise la fenêtre en n lignes et m colonnes d'espaces graphiques.

Elle parcourt ces espaces ligne par ligne, jusqu' au i -iem espace pour placer le graphique indiqué.

Soit à réaliser la fenêtre graphique ci-dessous composées de 3 lignes et 2 colonnes :



Les commandes à saisir sont :

Pour avoir le 1^{ier} sous graphe

```
>> subplot(3,2,1)
>> r = -pi : 0.1 : +pi ;
>> plot( r , sin(r))
>> title('Sin t')
```

Pour avoir le 2^{iem} sous graphe

```
>> subplot(3,2,2)
>> plot(r , cos(r))
>> title('Cos t')
```

Pour avoir le 3^{iem} sous graphe

```
>> subplot(3,2,3)
>> bar(A,HH)
>> title('graphe 3')
```

Pour avoir le 4^{iem} sous graphe

```
>> subplot(3,2,4)
>> hist(randn(10000,1),50)
>> title('Graphe 4')
```

Pour avoir le 5^{iem} sous graphe

```
>> subplot(3,2,5)
>> plot3(x,y,z)
>> title('Helice')
```

Les valeurs de x, y et z sont celles du slide
« Fonction plot3() »

Pour avoir le 6^{iem} sous graphe

```
>> subplot(3,2,6)
>> plot(A,HH)
>> title('Graphe 6')
```