

5. Le Graphique

5.1 Les Graphiques 2-D

5.1.1 Fonctions de tracés

Les commandes de base

figure	Création d'une fenêtre graph
plot	Tracé de courbes lineaires et de points
semilogx	Tracé semi- logarithmique
semilogy	
loglog	Tracé logarithmique
polar	Tracé en coordonnées polaires
plotyy	Tracé de deux courbes dans deux systèmes d'axes superposés

Les utilitaires - pour changer l'aspect

subplot	Création de plusieurs axes
grid	Affichage d'une grille
zoom	Zoom !
axis	Définition des limites des axes
hold on/off	Superposition de tracés

Les annotations

title	Titre du graphique
xlabel	Libellé des axes
ylabel	
text	Commentaires à une position spécifique
legend	

MATLAB dispose d'un interpréteur Te χ (lettres grecques,symbols) Les codes sont disponibles dans l'aide HTML en tapant **>>helpdesk** puis Handle Graphics Object /text/string /Te χ codes

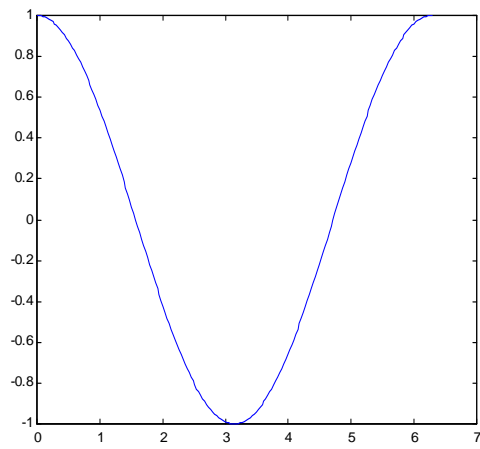
Syntaxe des commandes **plot**, **semilogx** , **semilogy** et **loglog**

plot (x1,y1,s1,x2,y2,s2,x3.....)

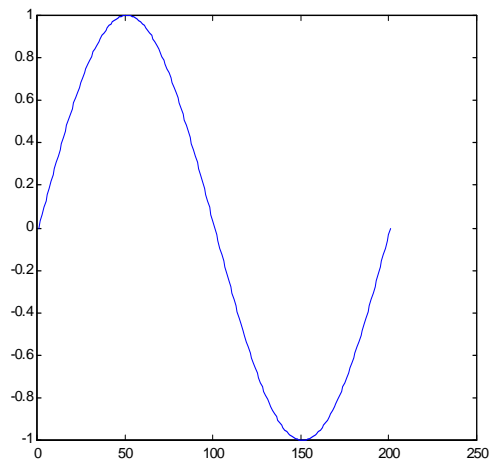
ou s = chaîne de caractères définissant la couleur, le style, le marqueur

Exemples

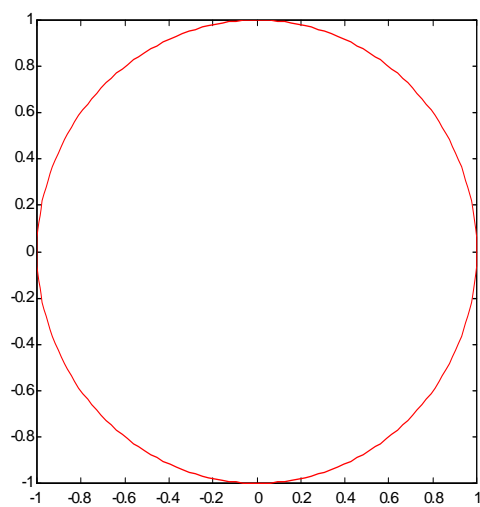
```
>> t=0:pi/100:2*pi;  
>> x = sin(t);  
>> y = cos(t);  
>> figure  
>> plot(t,y)
```



» plot(x)



» plot (x,y,'r')



(Sur l'ecran la courbe est une ellipse)

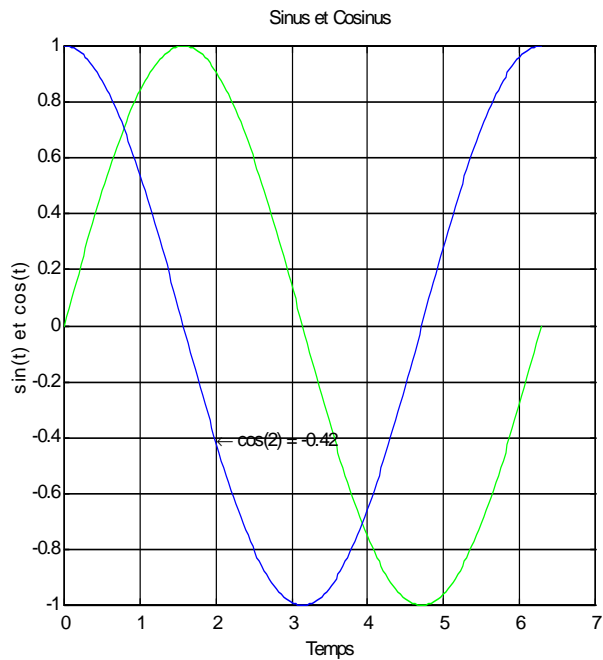
Les codes pour les couleurs et les styles

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

```

» plot(t,[x;y])
» plot(t,x,'g-',t,y,'b-')
» grid on
» title('Sinus et Cosinus')
» xlabel('Temps')
» ylabel('sin(t) et cos(t)')
» text(2,cos(2),'\leftarrow cos(2) = -0.42')

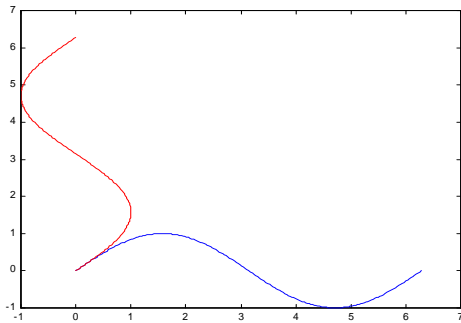
```



Question : Si par exemple A est une matrice (7,12) et on tape **plot(A)** qu'est ce qu'il va se passer?

Pour superposer des tracés, soit on définit plusieurs dans la commande de **plot** soit on utilise la commande **hold**

- » `plot(t,x)`
- » `hold on`
- » `plot(x,t,'r--')`
- » `hold off`



Pour effacer une figure on utilise `>> clf` et pour fermer les fenêtres en 'soft' on peut utiliser la commande **close**.

- La commande **zoom** active un zoom interactif sur un graphique 2D

Syntaxe :

- * **zoom on** pour activer **zoom off** pour désactiver
- * avec le bouton gauche de la souris, sélectionner une zone ou cliquer sur un point
- * avec le bouton droit de la souris cliquer une fois pour 'de-zoomer' avec un rapport 2, double-cliquer pour 'de-zoomer' complètement

- * La commande **axis** permet de changer les limites des axes

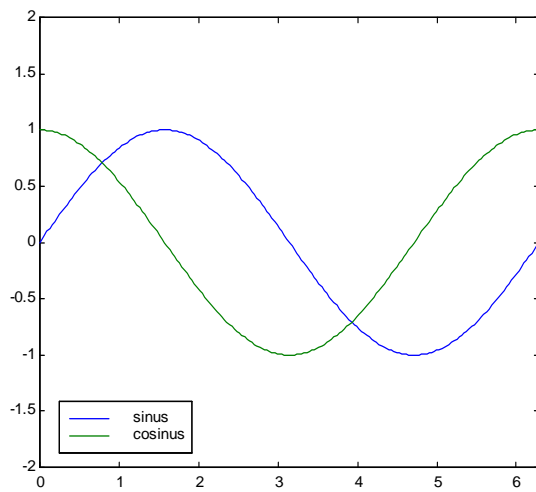
Syntaxe :

axis ([xmin xmax ymin ymax])

Exemple:

- » `plot(t,x,t,y,'--')`
- » `axis([min(t) max(t) -2 2])`
- » `legend('sinus','cosinus')`

Celle legende peut être déplacée avec la souris

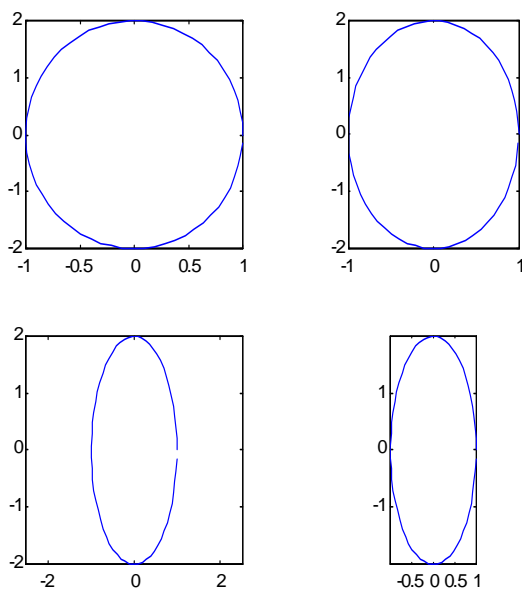


Par défaut, MATLAB crée des axes dont la taille est proportionnelle à la figure. La commande **axis** permet de modifier la taille des axes pour qu'un cercle, par exemple, apparaisse réellement sur l'écran comme un cercle et non pas comme une ellipse

```

» t=0:0.1:2*pi;
» x = cos(t);y=2*sin(t);
» plot(x,y)
» clf
» plot(x,y)
» axis normal
» axis square
» axis equal
» axis equal tight

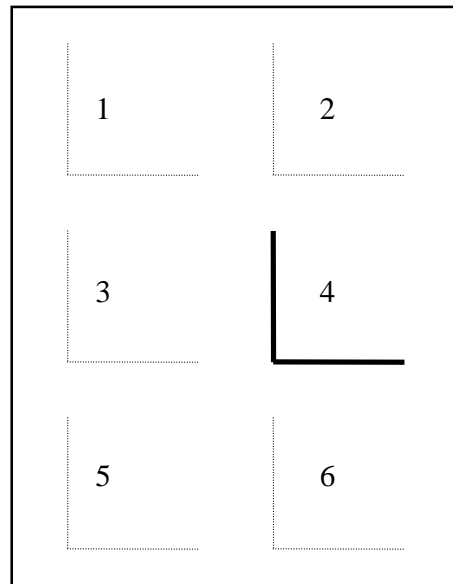
```



- La commande **subplot** permet de mettre plusieurs axes sur la même figure

Syntaxe : **subplot** (nblignes, nbcolonnes, indice)

Exemples : la commande **subplot(3,2,4)** sépare la figure en 3 lignes et 2 colonnes et crée le 4ème axe



Pour générer la graphique de la forme du repère le code suivant était utilisé.

```
» clf
» subplot(2,2,1)
» plot(x,y)
» axis normal
» subplot(2,2,2)
» plot(x,y)
» axis square
» subplot(2,2,3)
» plot(x,y)
» axis equal
» subplot(2,2,4)
» plot(x,y)
» axis equal tight
```

La règle générale !

Les commandes telles que **plot**, **text**, **title** permettent de placer des lignes et des textes.

Ces commandes sont toujours appliquées sur l'axe courant. Un axe est 'courant' lorsqu'on clique dessus ou lorsqu'on l'appelle avec la commande **subplot** (ou **plot** ...)

Exercice:

Soit dans la fenêtre de commande, soit dans un script :

- Charger le fichier de données exp_sq. mat
- Afficher 2 sous-plots (temps sur x, e sur y) sur la même figure -le premier avec des échelles linéaires, l'autre avec une échelle logarithmique en y

Pour voir l'effet de 'l'axe courant' sur cette exemple, taper

```
>>subplot(2,1,1)
```

```
>>xlabel('Echelle linéaire')
```

puis cliquer sur le repère du bas , puis écrire :

```
>>xlabel('Echelle logarithmique')
```

5.2 Les Graphiques 3-D

5.2.1 Fonctions de traces

Commandes de base

plot3	Tracé de courbe linéaires et de points
mesh	Maillage pour représenter des surfaces
surf	Représentation de surface

Utilitaires

hidden	Maillage transparent ou non
shading	Lissage des couleurs
light	ajouter un éclairage
colorbar	ajouter une barre de couleurs
colormap	définition de la table des couleurs
view	définir la direction du point de vue

Annotations

xlabel
ylabel
zlabel
text

5.2.1 Tracés linéaires en 3D

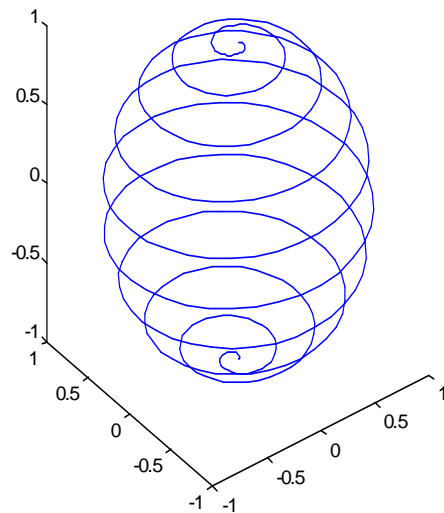
La commande **plot3** a la même syntaxe que la commande **plot**.

Exemple 1 : tracé avec des vecteurs

```

» az = (0:10:360*10)/180*pi;    azimuth
» el = (-90:180/360:90)/180*pi;  élévation
» r = 1;                          rayon
» [X,Y,Z]= sph2cart(az,el,r);    conversion en coordonnées cartésiennes
» plot3(X,Y,Z),axis square        trace en 3D

```



Exemple 2 : tracé avec des matrices.

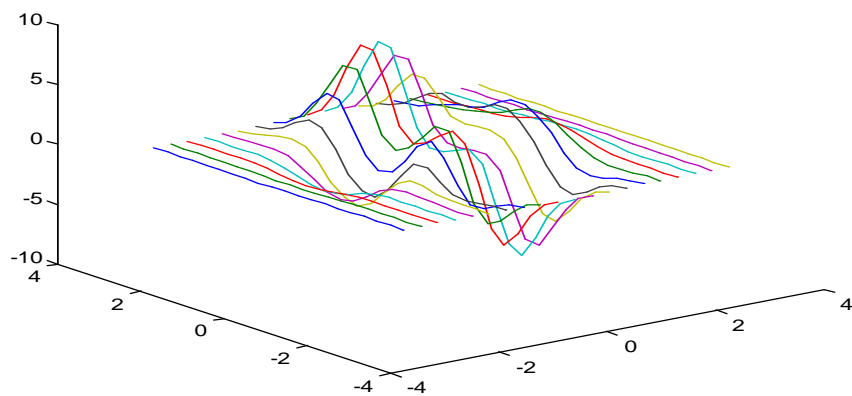
peaks est une fonction de démonstration du type $z = f(x,y)$

» `[X,Y,Z] = peaks(20);`

» `whos`

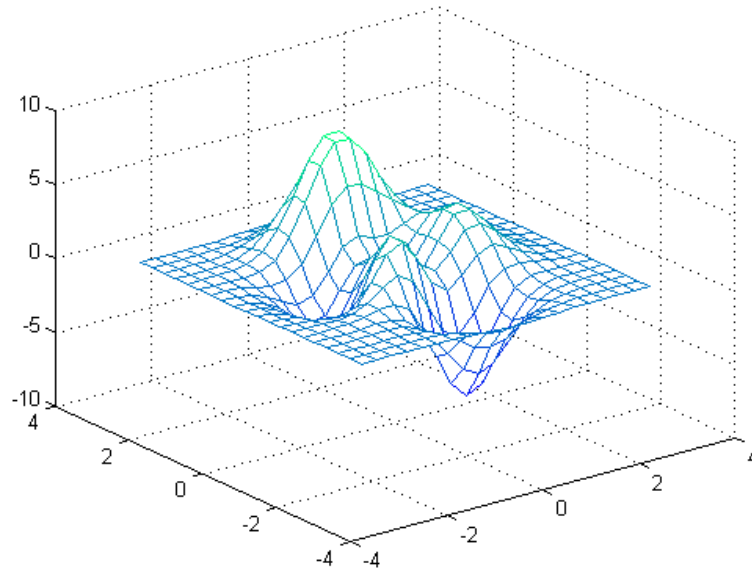
Name	Size	Bytes	Class
X	20x20	3200	double array
Y	20x20	3200	double array
Z	20x20	3200	double array

» `plot3(X,Y,Z)`

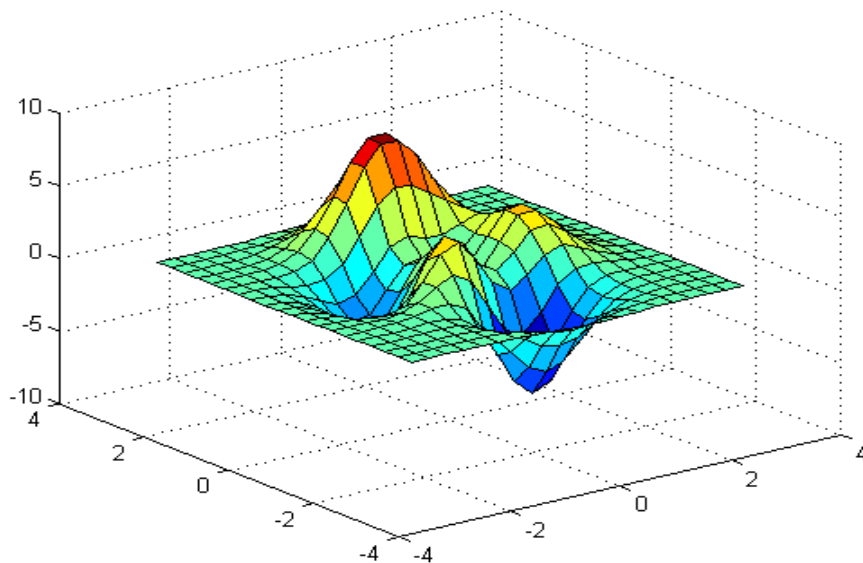


5.2.2 Surfaces

» `mesh(X,Y,Z)` représentation maillée de X,Y et Z calculés toute à l'heure
 » `colormap(winter);`



»
 »
 »



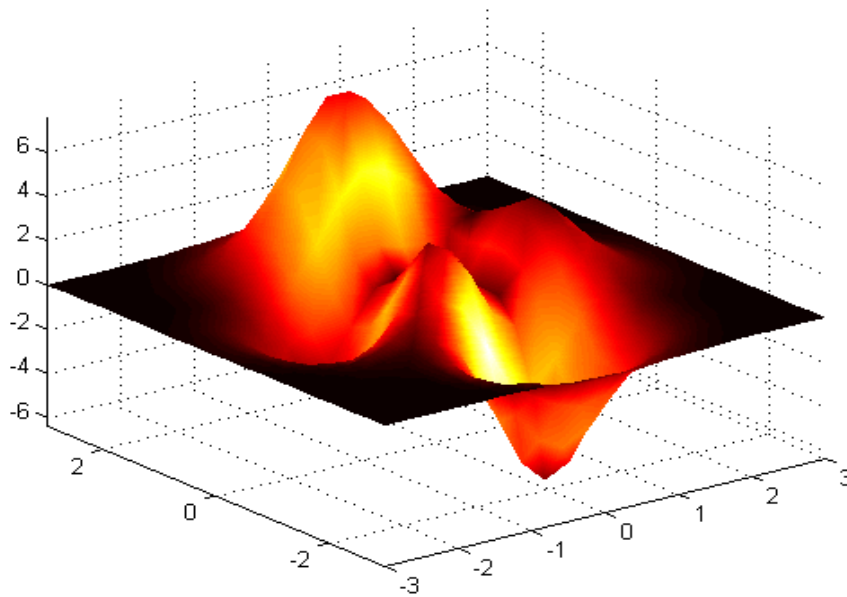
Par défaut la palette de couleur est associée à la matrice altitude. Cependant, la commande **surf** admet comme 4eme paramètre une matrice de taille identique aux trois autres. Lorsque ce 4eme paramètre est utilisé la table de couleurs est associée à ce parametre.

Dans l'exemple suivant, on associe le gradient de la surface à la couleur

```

» [dx,dy]=gradient(Z,0.2,0.2);
» P = sqrt(dx.^2 + dy.^2);
» surf(X,Y,Z,P)
» shading interp
» colormap(hot)
» axis tight

```



5.2.4 Source de lumière

MATLAB propose d'ajouter un éclairage particulier sur les surfaces (et patches) pour obtenir un image plus 'réaliste'. Cet éclairage est obtenu en utilisant la commande **light** qui génère un objet graphique (voir les détails sur les handles graphiques) proposant entre autres les propriétés suivantes:

- color
- style - source infinie ou locale
- position - direction (pour source infinie), position pour locale

L'exemple suivant illustre l'effet de l'ajout d'un éclairage sur une sphère de couleur rouge.

```

» [X,Y,Z] = sphere(36);
» surf(X,Y,Z)
» colormap([1 0 0]);
» shading interp
» axis equal
» light('posi',[-1 -1 0],'color','w')

```

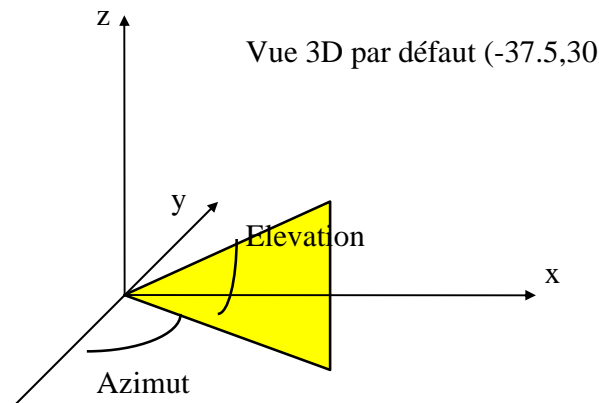
Les deux commandes suivantes permettent d'avoir un rendu encore plus réaliste:

lighting permet le choix de la méthode d'éclairage (algorithmes différents)
material agit sur les propriétés de réflectance de la surface (ou patch)

5.2.5 Point de vue

La commande **view** permet de définir la direction du point de vue sur la surface

Syntaxe : **view** (Azimut,Elevation) (tout en DEGREE)



- » [X,Y,Z] = peaks(20);surf(X,Y,Z)
- » view(0,90)
- » view(30,90)
- » view(-7,-100)

Une rotation commandée par le souris est possible avec la commande **rotate3d**.

Les commandes **view** et **rotate3d** sont très simples à utiliser. Elles ont cependant des limitations. Elles permettent uniquement de définir la direction du point de vue. Il est donc impossible avec ces commandes de définir la position du point de vue, de faire des zooms sur la surface ou de définir des translations ou des rotations arbitraires.

Pour ce faire, les axes possèdent des propriétés 'camera' permettant de réaliser un contrôle plus complet.

5.3 Graphes Spécialisés

5.3.1 Barres et surfaces

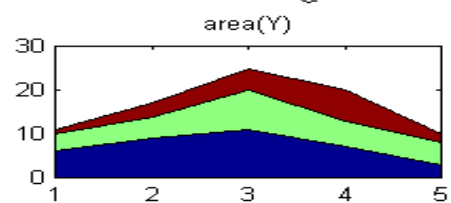
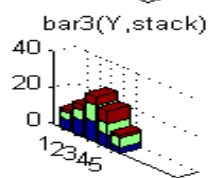
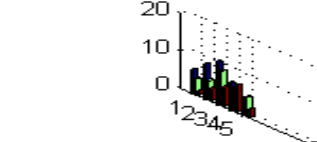
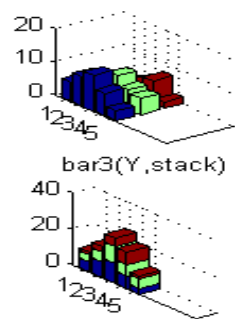
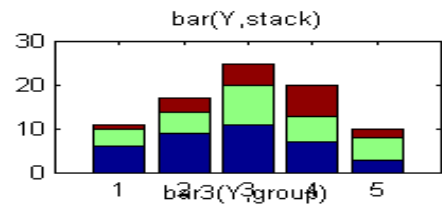
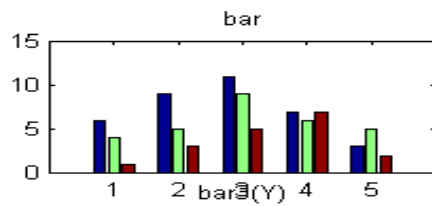
bar	barres verticales
barh	barres horizontales
bar3	barres verticales en 3D
bar3h	barres horizontales en 3D
area	Surfaces

Exemples :

```

» Y = [6 4 1; 9 5 3; 11 9 5; 7 6 7; 5 3 2];
» bar(Y)
» bar(Y,'stack')
» bar3(Y)
» bar3(Y,'group')
» bar3(Y,'stack')
» area(Y)

```



5.3.2. Camemberts

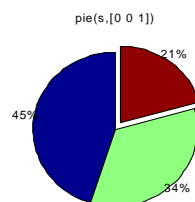
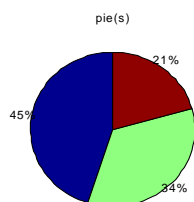
pie	Camembert
pie3	Camembert en 3D

Exemples:

```

» s=[33 25 15];
» pie(s)
» pie(s,[0 0 1])
» pie(s,[0 1 0])

```

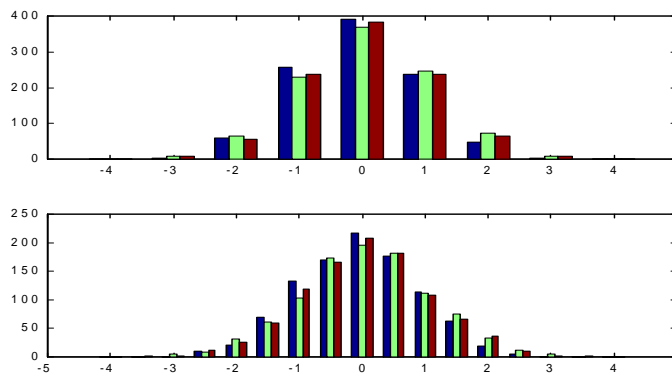


Lorsque la somme des elements du vecteur est superieure aa 1, la commande pie normalise. Dans le cas contraire pie trace un camembert incomplet.

```
» s2 = s(1:2)/sum(s);
» pie(s2)
```

5.3.3 Histogrammes

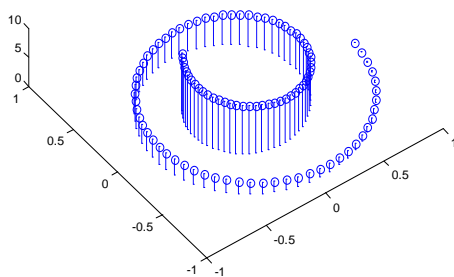
```
» Y = randn(1000,3);
» hist(Y,-4:4)
» hist(Y,-4:0.5:4)
```



5.3.4. Affichage de donnees discrettes

stem	affichage d'une séquence de valeurs discrètes
stem3	affichage 3D
stairs	affichage dechelons

```
» t = 0:0.1:10;
» s = 0.1+i;
» y = exp(-s*t);
» stem3(real(y),imag(y),t)
» view(-37,75)
» grid off
```



5.3.5 Affichage de vecteurs

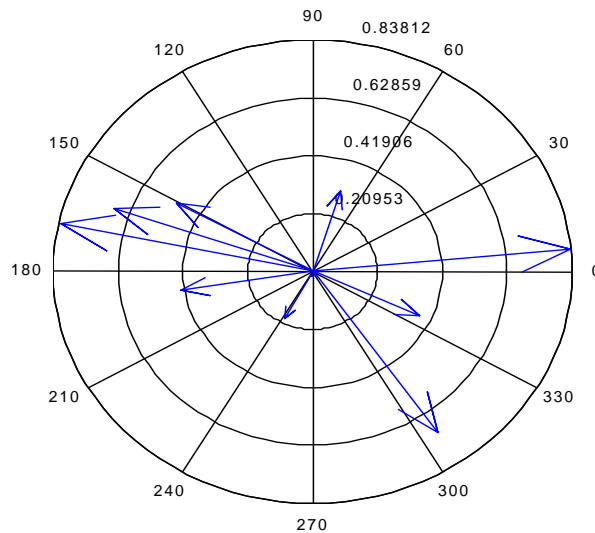
compass	affichage de vecteurs centrés sur 0
feather	vecteurs le long d'un axe
quiver	affichage dans le plan
quiver3	affichage dans l'espace

COMPASS Compass plot.

COMPASS(U,V) draws a graph that displays the vectors with components (U,V) as arrows emanating from the origin.

COMPASS(Z) is equivalent to COMPASS(REAL(Z),IMAG(Z)).

```
» a = rand(10,1)*2*pi;  
» r = rand(10,1);  
» compass(r.*cos(a),r.*sin(a))
```



5.4 Images

MATLAB lit et écrit des fichiers contenant différents formats d'image : TIFF, JPEG, BMP, PCX, XWD et HDH.

image	affichage d'une image
imagesc	affichage d'une image en niveaux de gris
imread	lecture d'un fichier contenant une image
imwrite	
imfinfo	lecture des informations d'un fichier d'image
capture	capture d'une image(figure)
print	sauvegarde ou impression d'image

La syntaxe exacte varie selon le type de format d'image

- image indexée
- image en niveaux de gris
- image en vraies couleurs

La commande `imfinfo` permet de connaître le format

```
» info= imfinfo('surprise.jpg')
info =
    Filename: 'surprise.jpg'
    FileModDate: '7-Jan-1999 20:22:34'
    FileSize: 43041
    Format: 'jpg'
    FormatVersion: ''
    Width: 500
    Height: 700
    BitDepth: 24
    ColorType: 'truecolor'
```

```
» RGB = imread('labo.jpg');
» image(RGB)
```

Le noyau MATLAB contient un certain nombre d'images. Ces images ne sont pas sauvegardées sous un des formats de fichiers standard. Elles sont stockées dans des fichiers binaires MATLAB et sont accessibles en utilisant la commande **load**. Ce sont toutes des images indexées.

```
» close all
» load earth
» image(X), colormap(map),axis square
```

L'éditeur graphique de Matlab a beaucoup évolué dans les dernières versions et la modification 'à la main' des graphiques est devenue facile. Les commandes répertoriées ci-dessus sont toujours nécessaires pour pouvoir 'programmer' l'allure des figures et des autres objets graphiques.

5.5 Animations

Une façon d'animer une image est de sauvegarder les images et les rejouer (cinéma)

moviein	initialisation de la mémoire (pre V.11)
getframe	capture de chaque image
movie	Exécution de l'animation

animsurf.m

```
% script pour générer une animation

Z = peaks;
surf(Z)

% on recoupe les limites des axes pour les imposer plus tard
lim = axis;

%création d'une matrice de N lignes et 10 colonnes N= nb pixels
% 10 = nb frames
M = moviein(10);
for j = 1:10
    surf(sin(2*pi*j/10)*Z,Z)
    axis(lim)
    %sauvegarde dans chaque colonne de la matrice M chaque pixel
    %dans la fenetre
    M(:,j) = getframe;
end
```

» colormap(winter)
» animsurf execution du script animsurf
» movie(M,10) animation

5.6 Handle Graphics

Le **Handle Graphics** est le système graphique de MATLAB qui permet de créer et de manipuler des lignes, des surfaces, des axes..... sous forme d'objets. Il offre aussi la possibilité de créer sa propre interface graphique à l'aide des boutons et du menu.

voir >>helpdesk /Graphic Handle Objects

Chacun des objets est identifié par un nombre unique qu'on appelle id ou handle

Exemple de creation d'objets

» fh = figure('position',[100 100 500 200],...
 'Color',[.2 .5 .6])


```
fh =  
1
```

```
» ah = axes('Color','none','Position',[0 0.2 1 0.6],'Box','on','XGrid','on')  
ah =  
6.0006
```

Les commandes **get** et **set** permettent la récupération et la définition des propriétés d'un objet.

Syntaxe : **get**(id-objet) liste toute les propriétés
 valeur = get(id-objet,'propriété') donne la valeur de la propriété choisie

 set(id-objet,'propriété',valeur) définit la valeur de la propriété choisie

Pour la liste des propriétés de l'objet tapez
 get(fh) pour figure ou **get**(gcf)
 get(ah) pour axes **get**(gca)
 get(h) pour texte

Exemple de modification des propriétés de la figure

```
set (fh,'Color','w','Name','EXEMPLE','MenuBar','none','Position',[200 200 500 200])
```

Exemple de modification des propriétés de l'axe

```
» bar(rand(1,5)*100)           - données  
» set(ah,'xlim',[0 7],'xtick',1:6,'xticklabel',{'jan';'fev';'mar';'avr';'mai';'juin'},'ygrid','on')  
  - définition des axes  
» h = text(1.5,90,'\leftarrow MAX VALUE') - ajouter du texte
```

5.7 Les Interfaces Homme-Machine

Pour générer les interfaces homme-machines l'ensemble des commandes suivantes est amplement suffisant... mais il y en a d'autres !

figure	création d'une figure
axes	création d'un système d'axes
uicontrol	création d'un bouton de control
uimenu	création d'un menu
uicontextmenu	création d'un menu attaché à un objet

Cependant, très rapidement la génération à l'aide de ces commandes deviennent complexes parce qu'il faut modifier un grand nombre de propriétés. Un utilitaire est disponible pour générer graphiquement les interfaces. Cet utilitaire s'appelle **guide** (Graphic User Interface Design Environment.)

Le **guide** regroupe les outils graphiques suivants :

- Control Panel interface principale
- Property Editor utilise pour modifier les objets graphiques
- Callback Editor éditeur de commandes Matlab à associer avec un bouton
- Alignment Tool outil d'alignement des objets graphiques
- Menu Editor éditeur de menus

Syntaxe >>guide