

Initiation au logiciel

" Matlab "

1.	L'AIDE EN LIGNE.	4
2.	TABLEAUX ET MATRICES.	4
2.1	Définir une matrice.	4
2.2	Addition et multiplication matricielles.	4
2.3	Inversion d'une matrice.	4
2.4	Opérations élément par élément.	5
2.5	Fonctions utiles pour la manipulation des matrices.	5
3.	PROGRAMMATION ET M-FILE FUNCTIONS.	5
3.1	Boucles for.	5
3.2	Boucles while.	5
3.3	Structure if else end.	5
3.4	Les fonctions.	6
4.	ANALYSE DE DONNEES.	6
5.	LES POLYNOMES.	6
5.1	Racines d'un polynôme.	6
5.2	Addition.	6
5.3	Multiplication.	6
5.4	Division.	6
5.5	Dérivation.	6
5.6	Evaluation.	6
6.	ANALYSE NUMERIQUE.	7
6.1	Tracé graphique.	7
6.2	Minimum d'une fonction.	7
6.3	Zéro d'une fonction.	7
6.4	Intégration.	7
6.5	Dérivation.	8
6.6	Equations différentielles.	8
7.	LE GRAPHISME A DEUX DIMENSIONS.	9
7.1	La fonction plot.	9
7.2	Les commandes grid, xlabel, ylabel et text.	9
7.3	La commande axis.	9
7.4	Les commandes figure, hold, close.	9
8.	EXERCICES D'APPLICATION.	10
9.	AIDE MATLAB 4	12
9.1	OPERATORS AND SPECIAL CHARACTERS.	12
9.2	LOGICAL CHARACTERISTICS.	12
9.3	ELEMENTARY MATH FUNCTIONS.	12
9.3.1	Trigonometric.	12
9.3.2	Exponential.	13
9.3.3	Complex.	13
9.3.4	Numeric.	13
9.4	NONLINEAR NUMERICAL METHODS.	13
9.5	ELEMENTARY MATRICES AND MATRIX MANIPULATION.	13
9.5.1	Elementary matrices.	13
9.5.2	Special variables and constants.	13
9.5.3	Time and dates.	14
9.5.4	Matrix manipulation.	14

Aide Matlab

9.6	SPECIALIZED MATRICES.....	14
9.7	GENERAL PURPOSE GRAPHICS FUNCTIONS.....	14
9.7.1	Figure window creation and control.....	14
9.7.2	Axis creation and control.....	14
9.7.3	Handle Graphics objects.....	15
9.7.4	Handle Graphics operations.....	15
9.7.5	Hardcopy and storage.....	15
9.7.6	Movies and animation.....	15
9.7.7	Miscellaneous.....	15
9.8	TWO DIMENSIONAL GRAPHICS.....	15
9.8.1	Elementary X-Y graphs.....	15
9.8.2	Specialized X-Y graphs.....	15
9.8.3	Graph annotation.....	16
9.9	THREE DIMENSIONAL GRAPHICS.....	16
9.9.1	Line and area fill commands.....	16
9.9.2	Contour and other 2-D plots of 3-D data.....	16
9.9.3	Surface and mesh plots.....	16
9.9.4	Volume visualization.....	16
9.9.5	Graph appearance.....	16
9.9.6	Graph annotation.....	16
9.9.7	3-D objects.....	17
9.10	MATRIX FUNCTIONS NUMERICAL LINEAR ALGEBRA.....	17
9.10.1	Matrix analysis.....	17
9.10.2	Linear equations.....	17
9.10.3	Eigenvalues and singular values.....	17
9.10.4	Matrix functions.....	17
9.11	POLYNOMIAL AND INTERPOLATION FUNCTIONS.....	17
9.11.1	Polynomials.....	17
9.11.2	Data interpolation.....	18
9.11.3	Spline interpolation.....	18
9.12	DATA ANALYSIS AND FOURIER TRANSFORM FUNCTIONS.....	18
9.12.1	Basic operations.....	18
9.12.2	Finite differences.....	18
9.12.3	Vector operations.....	18
9.12.4	Correlation.....	18
9.12.5	Filtering and convolution.....	18
9.12.6	Fourier transforms.....	18
9.13	SOUND PROCESSING FUNCTIONS.....	19
9.14	CHARACTER STRING FUNCTIONS.....	19
9.14.1	General.....	19
9.14.2	String comparison.....	19
9.14.3	String to number conversion.....	19
9.14.4	Hexadecimal to number conversion.....	19
9.15	SPECIALIZED MATH FUNCTIONS.....	19
9.16	SPARSE MATRIX FUNCTIONS.....	20
9.16.1	Elementary sparse matrices.....	20
9.16.2	Full to sparse conversion.....	20
9.16.3	Working with nonzero entries of sparse matrices.....	20
9.16.4	Visualizing sparse matrices.....	20
9.16.5	Reordering algorithms.....	20
9.16.6	Norm, condition number, and rank.....	20
9.16.7	Operations on trees.....	20
9.16.8	Micellanous.....	21
9.17	LANGUAGE CONSTRUCTS AND DEBUGGING.....	21
9.17.1	Matlab as a programming language.....	21
9.17.2	Control flow.....	21
9.17.3	Interactive input.....	21
9.17.4	Debugging commands.....	21
9.18	COLOR CONTROL AND LIGHTING MODEL FUNCTIONS.....	21
9.18.1	Color controls.....	21
9.18.2	Color maps.....	21
9.18.3	Color map related functions.....	22

Aide Matlab

9.18.4	Lighting models.	22
9.19	LOW-LEVEL FILE I/O FUNCTIONS.	22
9.19.1	File opening and closing.	22
9.19.2	Unformatted I/O.	22
9.19.3	Formatted I/O.	22
9.19.4	File positioning.	22
9.19.5	String conversion.	22
9.19.6	File Import/Export Routines.	22
9.19.7	WK1 Format.	22
9.19.8	CSV Format.	22
9.19.9	ASCII Delimited Format.	23
9.20	GENERAL PURPOSE COMMANDS.	23
9.20.1	Managing commands and functions.	23
9.20.2	Managing variables and the workspace.	23
9.20.3	Working with files and the operating system.	23
9.20.4	Controlling the command window.	23
9.20.5	Starting and quitting from Matlab.	23
9.20.6	General information.	23

Aide Matlab

Le logiciel Matlab est ouvert dans la fenêtre de commande Matlab **Mcw** (Matlab command window).

1. L'AIDE EN LIGNE.

On obtient les informations sur une fonction (contenue dans Matlab ou dans un des toolbox) en tapant **help** suivi du nom de cette fonction.

On peut aussi utiliser un mot clé avec la fonction **lookfor**, le mot clé n'étant pas nécessairement le nom d'une fonction de Matlab.

exemples : help diag, help clear, lookfor complex...

2. TABLEAUX ET MATRICES.

2.1 Définir une matrice.

$A = [1\ 2\ 3\ 4]$ définit la **matrice ligne** comportant les 4 éléments 1, 2, 3 et 4.

$A = [1,2,3,4]$ donne le même résultat.

$C = ['a' 'b' 'c']$ définit la **matrice ligne** comportant les caractères a, b et c.

$B = [1;2;3;4]$ définit la **matrice colonne** comportant les mêmes valeurs que A. Si on remarque que B est la transposée de A, on peut aussi écrire $B = A'$ l'apostrophe à droite signifiant la **transposition**.

$C = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]$ définit la **matrice carrée** 3x3, dont la première ligne est 1 2 3, la seconde ligne 4 5 6, etc... On obtient sa **transposée** en écrivant C' .

$C(i,:)$ est sa **i^{ème} ligne** et $C(:,j)$ est sa **j^{ème} colonne**.

$x = 0:0.5:10$ définit un vecteur ligne en partant du premier élément 0 et en incrémentant de 0.5 en 0.5 jusqu'au dernier élément 10. x' est alors un vecteur colonne.

linspace(1^{ère} valeur , dernière valeur , nb d'éléments) permet d'obtenir le même résultat en écrivant `linspace(0, 10, 21)`.

L'utilisation de **logspace()** permet d'avoir un espacement logarithmique des valeurs.

Matlab permet de créer des matrices particulières. Voir **zeros, ones, rand, eye**.

2.2 Addition et multiplication matricielles.

On peut additionner ou multiplier des matrices de même type comme on le ferait avec des nombres.

$B = C*D$ multiplie C par D, et le résultat est évidemment différent de $D*C$.

$B = C+D$ additionne les deux matrices.

Le résultat du produit $A*B$ (ligne*colonne) est un scalaire, alors que le produit $B*A$ (colonne*ligne) donne une matrice.

2.3 Inversion d'une matrice.

Soit la matrice carrée A dont on veut calculer l'inverse A_i .

$A = [10\ 10\ 0; 0\ 10\ 10; 10\ 19.9\ 10]$

$A_i = \text{inv}(A)$ calcule A_i , inverse de A si celle-ci n'est pas singulière.

On le vérifiera en calculant $D = A*A_i$ qui donne la matrice unité.

Si le déterminant **det(A)** est nul (matrice singulière), Matlab envoie un message d'erreur.

Remplacer dans A ci-dessus 19.9 par 20 et essayer de calculer l'inverse.

Il est évident que A est singulière dans ce cas, pourquoi?

$X = A \setminus B$ calcule $X = A^{-1}.B$, solution de $AX=B$.

$X = A/B$ calcule $A.B^{-1}$.

2.4 Opérations élément par élément.

Il est possible de faire le produit de deux tableaux a et b élément par élément, c'est à dire d'obtenir le tableau c tel que $c(i) = a(i) * b(i)$ en écrivant **c = a .* b**

$y = a.^n$ donnera y qui contiendra les éléments de a à la puissance n.

Pour la division élémentaire, essayer a./b et a.\b.

Pour la somme, quelles sont vos conclusions?

2.5 Fonctions utiles pour la manipulation des matrices.

Utiliser help pour voir **size**, **length**, **poly**, **rank**.

3. PROGRAMMATION ET M-FILE FUNCTIONS.

Matlab est aussi un langage de programmation qui dispose de structures de boucles **for** ou **while** ainsi que de la **structure if elseif else end**. Un exemple est donné ci-dessous pour chaque cas.

Noter que le point-virgule permet de gérer l'affichage ou le non affichage des valeurs manipulées.

3.1 Boucles for.

```
for i=1:2:10
    x(i)=i*i
end
```

3.2 Boucles while.

```
i=0;x=1;
while(1+x)>1
    x=x/2;
    i=i+1;
end
i
x=2*x
x-eps
```

Essayez de comprendre ce que représente **eps**.

3.3 Structure if else end.

```
x=1;
for i= 1:1000 x=x/2;
    if (1+x)<=1 x=2*x
        break
    end
end
i
x-eps
```

Conclusion?

3.4 Les fonctions.

Matlab fournit de nombreuses fonctions et permet à l'utilisateur d'écrire les fonctions spécifiques dont il a besoin, à condition de respecter la syntaxe. Par exemple la fonction **truc**(t,a) contenue dans le fichier **truc.m** ci-dessous

```
function y = truc(t,a); %function doit toujours être écrit en première ligne
y = 2*t.^2 + a;
est utilisée par le programme xf.m ci-dessous.
t = linspace(0,10,6);
a = 1.5;
z = feval('truc',t,a);
plot(t,z,'r');pause;plot(t,z,'y+');pause;plot(t,z,'r',t,z,'b+');
```

Certaines fonctions de Matlab ne sont pas prévues pour le passage de paramètres. On utilise dans ce cas une variable globale. Voir **global**.

4. ANALYSE DE DONNEES.

Les **données** seront par convention stockées sous forme de **matrice colonne**.

max(x) renvoie le maximum de chaque colonne du tableau x.

min(x) renvoie le minimum de chaque colonne du tableau x.

mean(x) donne la valeur moyenne de chaque colonne du tableau x.

Voir aussi **hist**, **sort**....

5. LES POLYNOMES.

Matlab représente un polynôme comme un **vecteur ligne** qui contient les coefficients des termes rangés selon les puissances décroissantes.

$p = [1 \ -12 \ 0 \ 5 \ 0]$ représente $x^4 - 12x^3 + 5x$.

5.1 Racines d'un polynôme.

r=roots(p) fournit les racines du polynôme donné ci-dessus.

p=poly(r) est l'opération inverse qui redonne les coefficients rangés selon les puissances décroissantes.

5.2 Addition.

On additionne les polynômes a et b en écrivant **d = a+b** s'ils sont de même degré. Sinon il faut compléter celui qui a le degré le plus faible avec des zéros.

5.3 Multiplication.

La fonction **conv(a,b)** réalise le produit des polynômes a et b. On renvoie le résultat dans c en écrivant **c = conv(a,b)**. Ce produit est obtenu par **convolution** des vecteurs a et b.

5.4 Division.

La fonction **deconv()** permet la division.

[q,r] = deconv(c,b) divise c par b. q est le quotient et r le reste.

5.5 Dérivation.

La fonction **polyder()** fournit la dérivée d'un polynôme.

h = polyder(g) donne h dérivée de g.

5.6 Evaluation.

La valeur prise par le polynôme pour une valeur de la variable est fournie par la fonction **polyval()**

Aide Matlab

```
exemple : x = linspace(-1,3,5);
          p = [1 4 -7 -10];
          v=polyval(p,x), pause, plot(x,v)
```

Que fait ce programme?

6. ANALYSE NUMERIQUE.

Il n'est pas toujours possible de déterminer analytiquement les extréma d'une fonction, de l'intégrer ou de la dériver, etc... Matlab propose un certain nombre de fonctions qui peuvent résoudre numériquement ces problèmes.

6.1 *Tracé graphique.*

On utilise la fonction **fplot** qui assure une représentation plus fine que **plot** qui sera vue plus loin.

```
ex: g = '2*exp(-x).*sin(x)';
     fplot(g,[0 8])
```

6.2 *Minimum d'une fonction.*

On utilise la fonction **fmin**. Si on cherche un maximum, on utilise aussi **fmin** sur la fonction changée de signe.

```
ex: g = '2*exp(-x).*sin(x)';
     xmin = fmin(g,2,5) % on cherche sur 2 < x < 5
     x = xmin;
     ymin = eval(g) % calcule g pour x = xmin
```

6.3 *Zéro d'une fonction.*

On utilise la fonction **fzero** qui renvoie la valeur de x qui annule la fonction g que l'on aura préalablement définie dans un fichier g.m contenant :

```
function y=g(x); % fonction doit toujours être écrit en première ligne
y=2*exp(-x).*sin(x)
```

Ensuite, taper dans la fenêtre MCW:

```
xz = fzero('g',3.2) % cherche au voisinage de 3.2 et renvoie dans xz
```

6.4 *Intégration.*

Les fonctions **trapz**, **quad** et **quad8** calculent l'aire comprise entre la courbe et l'axe horizontal.

Définir le fichier g1.m :

```
function y=g1(x);
y=x.*x;
```

puis écrire le programme:

```
clear all
x=0:0.1:1;
y=g1(x);
aire1=trapz(x,y)
aire2=quad('g1',0,1)
```

6.5 Dérivation.

On évitera autant que possible la fonction **diff** qui dérive en faisant la différence entre éléments voisins dans le tableau des valeurs expérimentales, toujours faussées par le bruit.

Il est plus astucieux de remplacer la courbe par une bonne approximation polynomiale avec **polyfit** et de dériver celle-ci avec **polyder**.

6.6 Equations différentielles.

De nombreux systèmes physiques sont décrits par des équations différentielles, linéaires ou non. Pour intégrer une équation différentielle d'ordre n, on se ramène à un système de n équations du premier ordre, ce qui permettra d'utiliser les fonctions **ode23** ou **ode45** (ordinary differential equations) qui fourniront la solution cherchée.

Exemple : on veut résoudre l'équation de Van der Pol $d^2x/dt^2 - m(1-x^2)dx/dt + x = 0$.

Elle représente un système du second degré dont l'amortissement est non linéaire.

On pose $y_1 = x$ et $y_2 = dx/dt$, d'où le système équivalent: $dy_1/dt = y_2$
 $dy_2/dt = m(1-y_1^2)y_2 - y_1$

Pour $m = 2$, on programme :

```
function yprime = vdpol(t,y);  
m = 2;  
yprime = [y(2); m*(1-y(1).^2)*y(2) - y(1)];
```

On remarque que yprime est un vecteur colonne à deux éléments qui décrit complètement l'équation différentielle.

Le programme suivant calcule la solution :

```
[t,y] = ode23('vdpol',[0,30],[0;0.01]); %utiliser help pour les arguments  
y1 = y(:,1) % y(:,n) récupère la nème colonne de y, donc x  
plot(t,y1) % trace x en fonction du temps
```


7. LE GRAPHISME A DEUX DIMENSIONS.

Matlab propose de nombreuses fonctions graphiques. Nous donnons ici les plus utilisées avec un minimum d'explications. Utilisez **help** pour en savoir plus.

7.1 La fonction plot.

plot permet de tracer une ou plusieurs courbes sur le même graphe, en choisissant le style du trait et sa couleur et en marquant les points du tableau des valeurs si on le désire.

```
ex :      x = linspace(0,2*pi,30);
          y = sin(x); z = cos(x);
          plot(x,y,'g:',x,z,'r--',x,y,'mo',x,z,'c+');
```

7.2 Les commandes grid, xlabel, ylabel et text.

grid on met la grille sur le graphe tracé par plot. **grid off** efface la grille.

xlabel et **ylabel** écrivent du texte le long de l'axe correspondant.

text(x,y,'string') écrit la chaîne de caractères string aux coordonnées x,y du graphe.

gtext('string') donne un curseur qu'on amène à l'endroit où on désire placer le texte. Celui ci s'écrit quand on clique la souris.

7.3 La commande axis.

axis([xmin,xmax,ymin,ymax]) impose les échelles en x et y.

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

7.4 Les commandes figure, hold, close....

La commande **figure** est très riche en possibilités. Elle permet de créer une figure. La commande **set** permet d'agir sur tous les paramètres. Voici un exemple d'utilisation :

```
t=linspace(0,10,100)
close
figure(1);
set(1,'position',[0 0 300 300]);
plot(t,sin(t),'y');grid;
title('commande');
figure(2);
set(2,'position',[320 0 300 300]);
plot(t,(sin(t)).^2,'c');grid;
title('sortie processus');
```

hold on permet de tracer plusieurs courbes sur la figure courante, jusqu'à la commande **hold off** qui permettra de changer.

close(n) permet de fermer la fenêtre graphique n.

8. EXERCICES D'APPLICATION.

Pour chaque exercice, on écrira un programme **Matlab** intitulé x0_i.m pour l'exercice i et on utilisera **help** autant que nécessaire.

Exercice 1 (fichier x0_1.m)

Résoudre avec **Matlab** le système :

$$\begin{aligned}x + y + z &= 2 \\ 2x + y + z &= 0 \\ 7x - 5y - z &= -6\end{aligned}$$

Exercice 2 (fichier x0_2.m)

Résoudre graphiquement le système

$$\begin{aligned}y &= 2x - 4 \\ y &= -x + 5\end{aligned}$$

On utilisera **ginput(...)**.

Exercice 3 (fichier x0_3.m)

Résoudre $x \exp(-x^2) = 0.1/x^2$ en utilisant la méthode de l'exercice 2. Combien y a-t-il de solutions?

Exercice 4 (fichiers x0_4.m et f0_4.m)

Ecrire le fichier f0_4.m qui crée la **fonction** $y = f0_4(x) = x \exp(-x^2) - 0.1/x^2$
Ecrire x0_4.m qui trace le graphe de f0_4(x) et donne les valeurs de x pour lesquelles f0_4(x) = 0.
voir **eval** (ou **feval**) et **fzero**.

Exercice 5 (fichier x0_5.m)

On donne les polynômes

$$\begin{aligned}p1 &= x^3 - 2x^2 - x + 2 \\ p2 &= -x^2 + 4x - 2 \\ p3 &= x^2 - 1\end{aligned}$$

Utiliser Matlab pour :

- Calculer leurs dérivées respectives dp1, dp2, dp3.
- Calculer le produit p12 = p1*p2.
- Calculer dp = p1 - p2 et en déduire les solutions de p1 = p2.
- Diviser p1 par p3.
- Calculer la valeur de p1 pour x = 0, 1, 2 et 3.
- Vérifier graphiquement les solutions trouvées au c/ et e/
voir **polyder**, **conv**, **roots**, **deconv**, **polyval**.

Aide Matlab

Exercice 6 (fichiers x0_6.m et f0_6.m)

On considère la fonction $y = 1 - \exp(-z.w.t) \cdot ((z/m) \cdot \sin(w_p.t) + \cos(w_p.t))$

avec $m = (1-z^2)^{1/2}$ et $w_p = m.w$ pour $w = 6.28$ rad/s.

Ecrire le fichier f0_6.m qui contiendra la fonction f0_6(t,z). Ecrire ensuite x0_6.m utilisant f0_6 pour représenter **sur le même graphique** les courbes représentatives de cette fonction pour les valeurs de z suivantes : 0.1, 0.3, 0.45 et 0.8 pour t variant de 0 à 5s. Indiquer la valeur de z à côté de chaque courbe, prendre **une couleur différente pour chaque courbe** et mettre comme titre « Réponse indicielle d'un second ordre ».

voir **figure, hold on, hold off, feval, num2str, xlabel, ylabel, title, gtext, grid**.

Exercice 7 (fichier x0_7.m)

Modifier le programme précédent afin de pouvoir rentrer au clavier les valeurs de z. voir **input**.

Exercice 8 (fichier x0_8.m)

Modifier le programme de l'exercice 6 pour mesurer le temps de réponse à 5% de chaque courbe, temps à partir duquel la valeur de y sera toujours comprise entre 0.95 et 1.05.

voir **ginput**.

Exercice 9 (fichiers x0_9.m et f0_9.m)

On considère le système décrit par l'équation différentielle suivante, x étant la sortie et u l'entrée du système

$$d^2x/dt^2 + 2z w dx/dt + w^2 x = w^2 u(t)$$

On considère le cas où u(t) est constante et égale à 1 et où w=6.28 rad/s.

Ecrire la fonction yprime = f0_9(t,y) qui décrit l'équation ci-dessus puis écrire x0_9.m qui intègre l'équation avec ode23 et donne une représentation graphique de la solution.

Modifier ensuite ce programme pour tracer sur le même graphe les solutions correspondant à z = 0.25, 0.50, 0.75 et 1 avec une couleur différente pour chaque courbe.

voir **global, ode23**.

Exercice 10 (fichier x0_10.m)

On considère l'équation différentielle du 3^{ème} ordre suivante, qui décrit la sortie x d'un système attaqué en entrée par le signal sin wt.

$$d^3x/dt^3 + 4d^2x/dt^2 + 5 dx/dt + 2 x = \sin wt$$

Résoudre cette équation avec ode23 et représenter sur le même graphe la sortie du système pour

w = 3.14, 6.28 et 12.57 rad/s (une couleur différente pour chaque courbe), avec des conditions initiales nulles, puis avec x(0)=0, x'(0)=-1 et x''(0)=0.

9. AIDE MATLAB 5.3

9.1 OPERATORS AND SPECIAL CHARACTERS.

		<u>Help</u>
+	Plus	arith
-	Minus	arith
*	Matrix multiplication	arith
.*	Array multiplication	arith
^	Matrix power	arith
.^	Array power	arith
\	Backslash or left division	slash
/	Slash or right division	slash
./	Array division	slash
kron	Kronecker tensor product	kron
:	Colon	colon
()	Parentheses	paren
[]	Brackets	paren
.	Decimal point	punct
..	Parent directory	punct
...	Continuation	punct
,	Comma	punct
;	Semicolon	punct
%	Comment	punct
!	Exclamation point	punct
'	Transpose and quote	punct
=	Assignment	punct
==	Equality	relop
<>	Relational operators	relop
&	Logical AND	relop
	Logical OR	relop
~	Logical NOT	relop
xor	Logical EXCLUSIVE OR	xor

9.2 LOGICAL CHARACTERISTICS.

exist	Check if variables or functions are defined.
any	True if any element of vector is true.
all	True if all elements of vector are true.
find	Find indices of non-zero elements.
isnan	True for Not-A-Number.
isinf	True for infinite elements.
finite	True for finite elements.
isempty	True for empty matrix.
issparse	True for sparse matrix.
isstr	True for text string.
isglobal	True for global variables.

9.3 ELEMENTARY MATH FUNCTIONS.

9.3.1 Trigonometric.

sin	Sine.
sinh	Hyperbolic sine.
asin	Inverse sine.
asinh	Inverse hyperbolic sine.
cos	Cosine.
cosh	Hyperbolic cosine.
acos	Inverse cosine.
acosh	Inverse hyperbolic cosine.
tan	Tangent.
tanh	Hyperbolic tangent.
atan	Inverse tangent.

Aide Matlab

atan2	Four quadrant inverse tangent.
atanh	Inverse hyperbolic tangent.
sec	Secant.
sech	Hyperbolic secant.
asec	Inverse secant.
asech	Inverse hyperbolic secant.
csc	Cosecant.
csch	Hyperbolic cosecant.
acsc	Inverse cosecant.
acsch	Inverse hyperbolic cosecant.
cot	Cotangent.
coth	Hyperbolic cotangent.
acot	Inverse cotangent.
acoth	Inverse hyperbolic cotangent.

9.3.2 Exponential.

exp	Exponential.
log	Natural logarithm.
log10	Common logarithm.
sqrt	Square root.

9.3.3 Complex.

abs	Absolute value.
angle	Phase angle.
conj	Complex conjugate.
imag	Complex imaginary part.
real	Complex real part.

9.3.4 Numeric.

fix	Round towards zero.
floor	Round towards minus infinity.
ceil	Round towards plus infinity.
round	Round towards nearest integer.
rem	Remainder after division.
sign	Signum function.

9.4 NONLINEAR NUMERICAL METHODS.

ode23	Solve differential equations, low order method.
ode23p	Solve and plot solutions.
ode45	Solve differential equations, high order method.
quad	Numerically evaluate integral, low order method.
quad8	Numerically evaluate integral, high order method.
fmin	Minimize function of one variable.
fmins	Minimize function of several variables.
fzero	Find zero of function of one variable.
fplot	Plot function.

See also The Optimization Toolbox, which has a comprehensive and function functions for optimizing and minimizing functions.

9.5 ELEMENTARY MATRICES AND MATRIX MANIPULATION.

9.5.1 Elementary matrices.

zeros	Zeros matrix.
ones	Ones matrix.
eye	Identity matrix.
rand	Uniformly distributed random numbers.
randn	Normally distributed random numbers.
linspace	Linearly spaced vector.
logspace	Logarithmically spaced vector.
meshgrid	X and Y arrays for 3-D plots.
:	Regularly spaced vector.

9.5.2 Special variables and constants.

ans	Most recent answer.
-----	---------------------

Aide Matlab

eps	Floating point relative accuracy.
realmax	Largest floating point number.
realmin	Smallest positive floating point number.
pi	3.1415926535897....
i, j	Imaginary unit.
inf	Infinity.
NaN	Not-a-Number.
flops	Count of floating point operations.
nargin	Number of function input arguments.
nargout	Number of function output arguments.
computer	Computer type.
isieee	True for computers with IEEE arithmetic.
isstudent	True for the Student Edition.
why	Succinct answer.
version	MATLAB version number.

9.5.3 Time and dates.

clock	Wall clock.
cputime	Elapsed CPU time.
date	Calendar.
etime	Elapsed time function.
tic, toc	Stopwatch timer functions.

9.5.4 Matrix manipulation.

diag	Create or extract diagonals.
fliplr	Flip matrix in the left/right direction.
flipud	Flip matrix in the up/down direction.
reshape	Change size.
rot90	Rotate matrix 90 degrees.
tril	Extract lower triangular part.
triu	Extract upper triangular part.
:	Index into matrix, rearrange matrix.

9.6 SPECIALIZED MATRICES.

compan	Companion matrix.
gallery	Several small test matrices.
hadamard	Hadamard matrix.
hankel	Hankel matrix.
hilb	Hilbert matrix.
invhilb	Inverse Hilbert matrix.
kron	Kronecker tensor product.
magic	Magic square.
pascal	Pascal matrix.
rosser	Classic symmetric eigenvalue test problem.
toeplitz	Toeplitz matrix.
vander	Vandermonde matrix.
wilkinson	Wilkinson's eigenvalue test matrix.

9.7 GENERAL PURPOSE GRAPHICS FUNCTIONS.

9.7.1 Figure window creation and control.

figure	Create Figure (graph window).
gcf	Get handle to current figure.
clf	Clear current figure.
close	Close figure.

9.7.2 Axis creation and control.

subplot	Create axes in tiled positions.
axes	Create axes in arbitrary positions.
gca	Get handle to current axes.
cla	Clear current axes.
axis	Control axis scaling and appearance.
caxis	Control pseudocolor axis scaling.
hold	Hold current graph.

9.7.3 Handle Graphics objects.

figure	Create figure window.
axes	Create axes.
line	Create line.
text	Create text.
patch	Create patch.
surface	Create surface.
image	Create image.
uicontrol	Create user interface control.
uimenu	Create user interface menu.

9.7.4 Handle Graphics operations.

set	Set object properties.
get	Get object properties.
reset	Reset object properties.
delete	Delete object.
gco	Get handle to current object.
drawnow	Flush pending graphics events.
newplot	M-file preamble for NextPlot property.
findobj	Find objects with specified property values.

9.7.5 Hardcopy and storage.

print	Print graph or save graph to file.
printopt	Configure local printer defaults.
orient	Set paper orientation.
capture	Screen capture of current figure.

9.7.6 Movies and animation.

moviein	Initialize movie frame memory.
getframe	Get movie frame.
movie	Play recorded movie frames.

9.7.7 Miscellaneous.

ginput	Graphical input from mouse.
ishold	Return hold state.
graymon	Set graphics window defaults for gray-scale monitors.
rbbox	Rubberband box.
rotate	Rotate an object about a specified direction.
terminal	Set graphics terminal type.
uiputfile	Put up dialog box for saving files.
uigetfile	Put up dialog box which queries for file names.
whitebg	Set graphics window defaults for white background.
zoom	Zoom in and out on a 2-D plot.
waitforbuttonpress	Wait for key/buttonpress over figure.

See also PLOTXY, PLOTXYZ, COLOR.

9.8 TWO DIMENSIONAL GRAPHICS.

9.8.1 Elementary X-Y graphs.

plot	Linear plot.
loglog	Log-log scale plot.
semilogx	Semi-log scale plot.
semilogy	Semi-log scale plot.
fill	Draw filled 2-D polygons.

9.8.2 Specialized X-Y graphs.

polar	Polar coordinate plot.
bar	Bar graph.
stem	Discrete sequence or "stem" plot.
stairs	Stairstep plot.

errorbar	Error bar plot.
hist	Histogram plot.
rose	Angle histogram plot.
compass	Compass plot.
feather	Feather plot.
fplot	Plot function.
comet	Comet-like trajectory.

9.8.3 Graph annotation.

title	Graph title.
xlabel	X-axis label.
ylabel	Y-axis label.
text	Text annotation.
gtext	Mouse placement of text.
grid	Grid lines.

See also PLOTXYZ, GRAPHICS.

9.9 **THREE DIMENSIONAL GRAPHICS.**

9.9.1 Line and area fill commands.

plot3	Plot lines and points in 3-D space.
fill3	Draw filled 3-D polygons in 3-D space.
comet3	3-D comet-like trajectories.

9.9.2 Contour and other 2-D plots of 3-D data.

contour	Contour plot.
contour3	3-D contour plot.
clabel	Contour plot elevation labels.
contourc	Contour plot computation (used by contour).
pcolor	Pseudocolor (checkerboard) plot.
quiver	Quiver plot.

9.9.3 Surface and mesh plots.

mesh	3-D mesh surface.
meshc	Combination mesh/contour plot.
meshz	3-D Mesh with zero plane.
surf	3-D shaded surface.
surfc	Combination surf/contour plot.
surfl	3-D shaded surface with lighting.
waterfall	Waterfall plot.

9.9.4 Volume visualization.

slice	Volumetric visualization plots.
-------	---------------------------------

9.9.5 Graph appearance.

view	3-D graph viewpoint specification.
viewmtx	View transformation matrices.
hidden	Mesh hidden line removal mode.
shading	Color shading mode.
axis	Axis scaling and appearance.
caxis	Pseudocolor axis scaling.
colormap	Color look-up table.

9.9.6 Graph annotation.

title	Graph title.
xlabel	X-axis label.
ylabel	Y-axis label.
zlabel	Z-axis label for 3-D plots.
text	Text annotation.
gtext	Mouse placement of text.
grid	Grid lines.

9.9.7 3-D objects.

cylinder Generate cylinder.
sphere Generate sphere.

See also COLOR, PLOTXY, GRAPHICS.

9.10 MATRIX FUNCTIONS NUMERICAL LINEAR ALGEBRA.

9.10.1 Matrix analysis.

cond Matrix condition number.
norm Matrix or vector norm.
rcond LINPACK reciprocal condition estimator.
rank Number of linearly independent rows or columns.
det Determinant.
trace Sum of diagonal elements.
null Null space.
orth Orthogonalization.
rref Reduced row echelon form.

9.10.2 Linear equations.

\and / Linear equation solution; use "help slash".
chol Cholesky factorization.
lu Factors from Gaussian elimination.
inv Matrix inverse.
qr Orthogonal-triangular decomposition.
qrdelete Delete a column from the QR factorization.
qrinsert Insert a column in the QR factorization.
nnls Non-negative least-squares.
pinv Pseudoinverse.
lsconv Least squares in the presence of known covariance.

9.10.3 Eigenvalues and singular values.

eig Eigenvalues and eigenvectors.
poly Characteristic polynomial.
polyeig Polynomial eigenvalue problem.
hess Hessenberg form.
qz Generalized eigenvalues.
rsf2csf Real block diagonal form to complex diagonal form.
cdf2rdf Complex diagonal form to real block diagonal form.
schur Schur decomposition.
balance Diagonal scaling to improve eigenvalue accuracy.
svd Singular value decomposition.

9.10.4 Matrix functions.

expm Matrix exponential.
expm1 M-file implementation of expm.
expm2 Matrix exponential via Taylor series.
expm3 Matrix exponential via eigenvalues and eigenvectors.
logm Matrix logarithm.
sqrtm Matrix square root.
funm Evaluate general matrix function.

9.11 POLYNOMIAL AND INTERPOLATION FUNCTIONS.

9.11.1 Polynomials.

roots Find polynomial roots.
poly Construct polynomial with specified roots.
polyval Evaluate polynomial.
polyvalm Evaluate polynomial with matrix argument.

residue	Partial-fraction expansion (residues).
polyfit	Fit polynomial to data.
polyder	Differentiate polynomial.
conv	Multiply polynomials.
deconv	Divide polynomials.

9.11.2 Data interpolation.

interp1	1-D interpolation (1-D table lookup).
interp2	2-D interpolation (2-D table lookup).
interpft	1-D interpolation using FFT method.
griddata	Data gridding.

9.11.3 Spline interpolation.

spline	Cubic spline data interpolation.
ppval	Evaluate piecewise polynomial.

9.12 DATA ANALYSIS AND FOURIER TRANSFORM FUNCTIONS.

9.12.1 Basic operations.

max	Largest component.
min	Smallest component.
mean	Average or mean value.
median	Median value.
std	Standard deviation.
sort	Sort in ascending order.
sum	Sum of elements.
prod	Product of elements.
cumsum	Cumulative sum of elements.
cumprod	Cumulative product of elements.
trapez	Numerical integration using trapezoidal method.

9.12.2 Finite differences.

diff	Difference function and approximate derivative.
gradient	Approximate gradient.
del2	Five-point discrete Laplacian.

9.12.3 Vector operations.

cross	Vector cross product.
dot	Vector dot product.

9.12.4 Correlation.

corrcoef	Correlation coefficients.
cov	Covariance matrix.
subspace	Angle between subspaces.

9.12.5 Filtering and convolution.

filter	One-dimensional digital filter.
filter2	Two-dimensional digital filter.
conv	Convolution and polynomial multiplication.
conv2	Two-dimensional convolution.
deconv	Deconvolution and polynomial division.

9.12.6 Fourier transforms.

fft	Discrete Fourier transform.
fft2	Two-dimensional discrete Fourier transform.
ifft	Inverse discrete Fourier transform.
ifft2	Two-dimensional inverse discrete Fourier transform.
abs	Magnitude.
angle	Phase angle.
unwrap	Remove phase angle jumps across 360 degree boundaries.

fftshift	Move zeroth lag to center of spectrum.
cplxpair	Sort numbers into complex conjugate pairs.
nextpow2	Next higher power of 2.

9.13 SOUND PROCESSING FUNCTIONS.

sound	Convert vector into sound.
saxis	Sound axis scaling.
Computer-specific sound functions.	
auwrite	Write mu-law encoded audio file.
auread	Read mu-law encoded audio file.
wavwrite	Write MS Windows .WAV audio file.
wavread	Read MS Windows .WAV audio file.
mu2lin	Mu-law to linear conversion.
lin2mu	Linear to mu-law conversion.

9.14 CHARACTER STRING FUNCTIONS.

9.14.1 General.

strings	About character strings in Matlab.
abs	Convert string to numeric values.
setstr	Convert numeric values to string.
isstr	True for string.
blanks	String of blanks.
deblank	Remove trailing blanks.
str2mat	Form text matrix from individual strings.
eval	Execute string with Matlab expression.

9.14.2 String comparison.

strcmp	Compare strings.
findstr	Find one string within another.
upper	Convert string to uppercase.
lower	Convert string to lowercase.
isletter	True for letters of the alphabet.
isspace	True for white space characters.
strrep	Replace a string with another.
strtok	Find a token in a string.

9.14.3 String to number conversion.

num2str	Convert number to string.
int2str	Convert integer to string.
str2num	Convert string to number.
sprintf	Convert number to string under format control.
sscanf	Convert string to number under format control.

9.14.4 Hexadecimal to number conversion.

hex2num	Convert hex string to IEEE floating point number.
hex2dec	Convert hex string to decimal integer.
dec2hex	Convert decimal integer to hex string.

9.15 SPECIALIZED MATH FUNCTIONS.

besselj	Bessel function of the first kind.
bessely	Bessel function of the second kind.
besseli	Modified Bessel function of the first kind.
besselk	Modified Bessel function of the second kind.
beta	Beta function.
betainc	Incomplete beta function.
betaln	Logarithm of beta function.
ellipj	Jacobi elliptic functions.
ellipke	Complete elliptic integral.
erf	Error function.
erfc	Complementary error function.
erfcx	Scaled complementary error function.
erfinv	Inverse error function.
expint	Exponential integral function.

gamma	Gamma function.
gcd	Greatest common divisor.
gammainc	Incomplete gamma function.
lcm	Least common multiple.
legendre	Associated Legendre function.
gammaln	Logarithm of gamma function.
log2	Dissect floating point numbers.
pow2	Scale floating point numbers.
rat	Rational approximation.
rats	Rational output.
cart2sph	Transform from Cartesian to spherical
cart2pol	Transform from Cartesian to polar
pol2cart	Transform from polar to Cartesian
sph2cart	Transform from spherical to Cartesian

9.16 SPARSE MATRIX FUNCTIONS.

9.16.1 Elementary sparse matrices.

speye	Sparse identity matrix.
sprandn	Sparse random matrix.
sprandsym	Sparse symmetric random matrix.
spdiags	Sparse matrix formed from diagonals.

9.16.2 Full to sparse conversion.

sparse	Create sparse matrix from nonzeros and indices.
full	Convert sparse matrix to full matrix.
find	Find indices of nonzero entries.
spconvert	Convert from sparse matrix external format.

9.16.3 Working with nonzero entries of sparse matrices.

nnz	Number of nonzero entries.
nonzeros	Nonzero entries.
nzmax	Amount of storage allocated for nonzero entries.
spones	Replace nonzero entries with ones.
spalloc	Allocate memory for nonzero entries.
issparse	True if matrix is sparse.
spfun	Apply function to nonzero entries.

9.16.4 Visualizing sparse matrices.

spy	Visualize sparsity structure.
gplot	Plot graph, as in "graph theory".

9.16.5 Reordering algorithms.

colmmd	Column minimum degree.
symmmd	Symmetric minimum degree.
symrcm	Reverse Cuthill-McKee ordering.
colperm	Order columns based on nonzero count.
randperm	Random permutation vector.
dmperm	Dulmage-Mendelsohn decomposition.

9.16.6 Norm, condition number, and rank.

normest	Estimate 2-norm.
condest	Estimate 1-norm condition.
sprank	Structural rank.

9.16.7 Operations on trees.

treelayout	Lay out a tree or forest.
treeplot	Plot a picture of a tree.
etree	Elimination tree of a matrix.
etreeplot	Plot the elimination tree.

9.16.8 Miscellaneous

symsfact	Symbolic factorization analysis.
spparms	Set parameters for sparse matrix routines.
spaument	Form least squares augmented system.

9.17 LANGUAGE CONSTRUCTS AND DEBUGGING.

9.17.1 Matlab as a programming language.

script	About Matlab scripts and M-files.
function	Add new function.
eval	Execute string with Matlab expression.
feval	Execute function specified by string.
global	Define global variable.
nargchk	Validate number of input arguments.
lasterr	Last error message.

9.17.2 Control flow.

if	Conditionally execute statements.
else	Used with IF.
elseif	Used with IF.
end	Terminate the scope of FOR, WHILE and IF statements.
for	Repeat statements a specific number of times.
while	Repeat statements an indefinite number of times.
break	Terminate execution of loop.
return	Return to invoking function.
error	Display message and abort function.

9.17.3 Interactive input.

input	Prompt for user input.
keyboard	Invoke keyboard as if it were a Script-file.
menu	Generate menu of choices for user input.
pause	Wait for user response.
uimenu	Create user interface menu.
uicontrol	Create user interface control.

9.17.4 Debugging commands.

dbstop	Set breakpoint.
dbclear	Remove breakpoint.
dbcont	Resume execution.
dbdown	Change local workspace context.
dbstack	List who called whom.
dbstatus	List all breakpoints.
dbstep	Execute one or more lines.
dbtype	List M-file with line numbers.
dbup	Change local workspace context.
dbquit	Quit debug mode.
mexdebug	Debug MEX-files.

9.18 COLOR CONTROL AND LIGHTING MODEL FUNCTIONS.

9.18.1 Color controls.

colormap	Color look-up table.
caxis	Pseudocolor axis scaling.
shading	Color shading mode.

9.18.2 Color maps.

hsv	Hue-saturation-value color map.
gray	Linear gray-scale color map.
hot	Black-red-yellow-white color map.
cool	Shades of cyan and magenta color map.
bone	Gray-scale with a tinge of blue color map.
copper	Linear copper-tone color map.

pink	Pastel shades of pink color map.
prism	Prism color map.
jet	A variant of HSV.
flag	Alternating red, white, blue, and black color map.

9.18.3 Color map related functions.

colorbar	Display color bar (color scale).
hsv2rgb	Hue-saturation-value to red-green-blue conversion.
rgb2hsv	Red-green-blue to hue-saturation-value conversion.
contrast	Gray scale color map to enhance image contrast.
brighten	Brighten or darken color map.
spinmap	Spin color map.
rgbplot	Plot color map.

9.18.4 Lighting models.

surfl	3-D shaded surface with lighting.
specular	Specular reflectance.
diffuse	Diffuse reflectance.
surfnorm	Surface normals.

9.19 LOW-LEVEL FILE I/O FUNCTIONS.

9.19.1 File opening and closing.

fopen	Open file.
fclose	Close file.

9.19.2 Unformatted I/O.

fread	Read binary data from file.
fwrite	Write binary data to file.

9.19.3 Formatted I/O.

fscanf	Read formatted data from file.
fprintf	Write formatted data to file.
fgetl	Read line from file, discard newline character.
fgets	Read line from file, keep newline character.

9.19.4 File positioning.

ferror	Inquire file I/O error status.
feof	Test for end-of-file.
fseek	Set file position indicator.
ftell	Get file position indicator.
frewind	Rewind file.

9.19.5 String conversion.

sprintf	Write formatted data to string.
sscanf	Read string under format control.

9.19.6 File Import/Export Routines.

9.19.7 WK1 Format.

wk1const	WK1 record definitions.
wk1read	Read WK1 file/range.
wk1write	Write out matrix in a WK1 formatted file.
wk1wrec	Write a WK1 record HEADER.

9.19.8 CSV Format.

csvread	Read Comma Separated Value formatted file into a matrix.
csvwrite	Write out matrix in a CSV formatted file.

9.19.9 ASCII Delimited Format.

dlmread	Read ASCII delimited file into a matrix.
dlmwrite	Write out matrix in ASCII delimited file format.

9.20 GENERAL PURPOSE COMMANDS.

9.20.1 Managing commands and functions.

help	On-line documentation.
doc	Load hypertext documentation.
what	Directory listing of M-, MAT- and MEX-files.
type	List M-file.
lookfor	Keyword search through the HELP entries.
which	Locate functions and files.
demo	Run demos.
path	Control Matlab's search path.

9.20.2 Managing variables and the workspace.

who	List current variables.
whos	List current variables, long form.
load	Retrieve variables from disk.
save	Save workspace variables to disk.
clear	Clear variables and functions from memory.
pack	Consolidate workspace memory.
size	Size of matrix.
length	Length of vector.
disp	Display matrix or text.

9.20.3 Working with files and the operating system.

cd	Change current working directory.
dir	Directory listing.
delete	Delete file.
getenv	Get environment value.
!	Execute operating system command.
unix	Execute operating system command & return result.
diary	Save text of Matlab session.

9.20.4 Controlling the command window.

cedit	Set command line edit/recall facility parameters.
clc	Clear command window.
home	Send cursor home.
format	Set output format.
echo	Echo commands inside script files.
more	Control paged output in command window.

9.20.5 Starting and quitting from Matlab.

quit	Terminate Matlab.
startup	M-file executed when Matlab is invoked.
matlabrc	Master startup M-file.

9.20.6 General information.

info	Information about Matlab and The MathWorks, Inc.
subscribe	Become subscribing user of Matlab.
hostid	Matlab server host identification number.
whatsnew	Information about new features not yet documented.
ver	Matlab, Simulink, and TOOLBOX version