

# INITIATION AU LOGICIEL MATLAB

MATLAB est un environnement interactif particulièrement adapté à tous les traitements comportant du calcul matriciel. Il permet de travailler interactivement en passant des commandes au clavier, ou de réaliser des programmes en plaçant ces commandes dans des fichiers textes, qui sont directement exécutables, le nom du fichier devenant alors une nouvelle commande MATLAB.

Ce document n'est pas destiné à remplacer le 'tutorial' très bien fait de la documentation, mais seulement à permettre une prise en main immédiate dans le cadre des TP. Il est composé de trois parties :

- la première présente les commandes de base. Elles permettent de se familiariser progressivement avec l'environnement.
- la seconde montre comment programmer sous MATLAB.
- la dernière donne des commandes complémentaires que l'on pourra expérimenter dans un deuxième temps.

## I - Commandes de base

### 1 . Lancement du logiciel

Après le lancement, MATLAB affiche un incitateur de commande ("*prompt*") : `>>`


Une session MATLAB consiste en une suite de *commandes* lancées au clavier et exécutées immédiatement par MATLAB (interpréteur).

MATLAB maintient une *pile des commandes* passées. On peut les rappeler par la touche `↑` et les éditer sur ligne à l'aide des touches flèches `←`, `→` et de la souris ou faire du "coupé-copier-coller".

La frappe des premiers caractères d'une commande passée suivie de `↑` permet de rappeler celle-ci. Si une commande ne tient pas sur une seule ligne, on peut utiliser `"..."` en fin de ligne comme caractère "*suite*".

### 2 . Manipulations élémentaires

MATLAB ne connaît qu'un seul type de variable : les *matrices de complexes*. Un scalaire réel (ou complexe) est une matrice  $1 \times 1$ . Un vecteur est une matrice  $1 \times N$  (*ligne*) ou  $N \times 1$  (*colonne*) (MATLAB distingue les vecteurs lignes et les vecteurs colonnes). Les nombres sont uniquement codés en double précision. Il existe des chaînes de caractères, mais ce sont aussi des vecteurs (chaque code ASCII est un réel double précision).

 **MATLAB distingue les majuscules des minuscules dans les noms de variables !**



Les quelques exemples suivants montrent des manipulations élémentaires. Essayez :

$x=[1\ 3\ 5]$  crée un vecteur 1 ligne x 3 colonnes contenant les valeurs 1, 3 et 5  
 $x1=2*x$  crée un vecteur x1 en multipliant par 2 tous les éléments de x  
 $y=x'$  crée un vecteur colonne transposé de x (opérateur: ')  
 $p=x*y$  calcule un produit scalaire  
 $q=y*x$  calcule une matrice 3x3  
 $q(2,3)$  est le terme situé à l'intersection de la ligne 2 et de la colonne 3 de q

Attention, la numérotation des lignes et des colonnes commence à 1 et non à zéro ! En traitement du signal, où les indices débutent souvent à 0, ajouter alors 1 pour passer en indice MATLAB.

☞ Attention : - les crochets [ ] sont utilisés pour créer des matrices ou des vecteurs  
 - les parenthèses ( ) sont utilisées pour désigner des indices.

Pour éviter l'affichage des résultats, utiliser un ";" en fin de ligne : essayez q et q;.

Le ";" permet également de séparer des plusieurs commandes sur une même ligne.

Pour poursuivre l'écriture d'une commande sur la ligne suivante, utiliser "..." en fin de ligne.

☞ Dès que vous manipulez des vecteurs ou matrices volumineux, n'oubliez pas de terminer toute commande par le ";" pour éviter un affichage à l'écran inutile et fastidieux !

### 3 . Aide en ligne

MATLAB dispose d'un très grand nombre de fonctions regroupées en sous-répertoires.

help affiche les différents sous-répertoires de fonctions disponibles.  
 help nom\_fonction donne la syntaxe d'appel et indique ce que fait la fonction nom\_fonction.  
 doc nom\_sujet documentation en HTML

☞ Utilisez la commande "help" pour en savoir plus sur les commandes !

On dispose aussi des commandes suivantes :

type nom\_fichier affiche le contenu d'un fichier (.m)  
 pwd permet de connaître le chemin du répertoire courant  
 dir affiche le contenu du répertoire courant

### 4 . Création de vecteurs et de matrices

$n=5:15$

$u=0:\pi/4:2*\pi$

$v=5:-0.5:1$

$A=[1\ 2\ 3;4\ 5\ 6;7\ 8\ 0]$  A matrice 3x3

$B=[1\ 2\ 3\ 4$  B matrice 3x4 avec ...

$5\ 6\ 7\ 8$  entrée ligne ...

$9\ 10\ 11\ 12]$  par ligne.

Vecteurs et matrices particulières :

ones(3) ones(1,3) ones(3,1)

zeros(3) zeros(1,3) zeros(3,1)

eye(3) → noter le jeu de mot, uniquement compréhensible en anglais ...

MATLAB gère également les nombres complexes :  $z=(2+3i)*(1+j)$   $L = \log_{10}(-2.2)$

Les variables i et j sont prédéfinies. Essayez help i.

Remarquer les "NaN" (Not a Number) dans L. Toute opération avec un "NaN" produit un "NaN".



**Récupération des dimensions des variables :**

`[L M]=size(A)` dimensions (ligne, colonnes) d'un vecteur ou d'une matrice  
`L=length(x)` longueur d'un vecteur ou plus grande dimension d'une matrice.

**5. Opérations sur les matrices et sur les éléments****Concaténation de matrices compatibles en nombre de lignes:**

`[q q]`      `[q x]`      Essayer `[q x]`

**Indexation :**

<code>A(1,1)</code>	<code>A(2,3)</code>	<code>A(3,1)</code>	éléments <code>A(ligne, colonne)</code>
<code>A(1,:)</code>	<code>A(:,3)</code>	<code>A(2:3,2)</code>	lignes et colonnes
<code>A(1:2,2:3)</code>	<code>A(:,1:2)</code>	<code>A(2:3,:)</code>	extraction de sous-matrices
<code>A(:)</code>			vecteur suite des éléments de la matrice
<code>u=1:2;</code>	<code>v=2:3;</code>	<code>A(u,v)</code>	extraction d'une sous-matrice par vecteurs
<code>A(:,3:-1:1)</code>			renversement de l'ordre des colonnes
<code>A(:,2) = []</code>			suppression de la 2 <sup>e</sup> colonne de A
<code>n([2 4 6 8 10])</code>			indexation par vecteur

**Addition et soustraction :**

`x + x1`      `x - x1`

Les dimensions doivent correspondre. Essayer `x + y`. Une exception existe lorsqu'un des opérandes est un scalaire. Essayer `n + 1` (évite l'emploi de `ones()`).

**Multiplication, division, élévation à une puissance :**

`"*"` est l'opérateur de multiplication *au sens matriciel*. Il nécessite une compatibilité des dimensions internes des matrices A et B (nbre de colonnes de A égal au nbre de lignes de B); la seule exception est la multiplication scalaire d'une matrice `s*A` (s scalaire, A matrice quelconque).

`".*"` (utilisation du point "." en préfixe) est l'opérateur de multiplication des *éléments de même position* des matrices (multiplication terme à terme); il en est de même pour la division (`./`) et l'élévation à une puissance (`.^`). Tous ces opérateurs nécessitent l'égalité des dimensions des matrices.

Essayez      `A*B`    `A*A`    `2.*A`    `A.*A`    `A./A`    `A./2`    `A^2`    `A.^2`  
 Comparer    `q*(2*ones(3))` et `q.*(2*ones(3))`

**6. Espace de travail**

Les variables créées sont conservées en mémoire. On peut en avoir la liste par:

<code>who</code>	affichage des variables en mémoire
<code>whos</code>	affichage des variables en mémoire avec leurs dimensions
<code>save</code>	sauvegarde des variables sur disque
<code>load</code>	chargement de variables depuis le disque

On peut effacer les variables par `clear` et libérer la place mémoire correspondante.



## 7 . Tracés graphiques

MATLAB a une fenêtre de commandes et une fenêtre graphique (passage de l'une à l'autre en cliquant sur la fenêtre désirée ou par la barre des tâches de "Windows").

### Graphismes 2D :

<code>N=20; nu0=0.02;</code>	<code>N</code> nombre de points ; <code>nu0</code> fréquence réduite
<code>n=0:N-1;</code>	<code>n</code> vecteur "abscisses"
<code>x=10*sin(2*pi*nu0*n);</code>	<code>y=10*cos(2*pi*nu0*n);</code> crée deux séquences sinusoïdales
<code>plot(x)</code>	tracé avec l'indice MATLAB en abscisse débutant implicitement à 1
<code>figure</code>	ouverture d'une fenêtre graphique supplémentaire
<code>plot(n,x)</code>	tracé avec abscisses correctes (avec un indice séquence débutant à 0)
<code>plot(n,x,'r-',n,x,'b+')</code>	tracé de 2 courbes avec le choix de la <u>couleur</u> et du <u>trait</u>
<code>zoom</code>	zoom avec la souris (bouton gauche: <i>zoom in</i> , bouton droit: <i>zoom out</i> )
<code>plot(n(1:10),x(1:10))</code>	tracé des 10 premiers points
<code>stem(n(1:10),x(1:10))</code>	tracé type séquence (impulsions de Kronecker)

On peut figer un graphisme pour que les tracés se superposent, avec la commande `hold` :

<code>plot(n,x)</code>	
<code>hold on</code>	maintient du graphisme courant activé
<code>plot(5*rand(1,N),'b')</code>	
<code>hold off</code>	maintient du graphisme courant désactivé

On peut utiliser plusieurs rectangles de tracé sur l'écran divisé en tableau :

<code>subplot(2,1,1); plot(n,x)</code>	division en 2 lignes, 1 colonne et choix rectangle n°1
<code>subplot(2,1,2); plot(n,y)</code>	division en 2 lignes, 1 colonne et choix rectangle n°2

### Annotations sur les graphiques

<code>title('titre du graphe')</code>	affiche 'titre du graphe' en titre du graphique courant
<code>xlabel('temps')</code>	affiche 'temps' sous l'axe des abscisses
<code>ylabel('tension')</code>	affiche 'tension' le long de l'axe des ordonnées
<code>text(10,3,'commentaire')</code>	affiche 'commentaire' au point d'abscisse 10 et d'ordonnée 3

### Fonctions de conversion de nombre en chaînes

<code>num2str(nu0)</code>	renvoie une chaîne contenant la valeur du réel <code>nu0</code> → noter le jeu de mots anglo-saxon ( <i>number to string</i> ) ...
<code>int2str(N)</code>	renvoie une chaîne contenant la valeur de l'entier <code>N</code> → noter le jeu de mots anglo-saxon ( <i>integer to string</i> ) ...



## II - Programmation sous MATLAB

La programmation sous MATLAB s'effectue à l'aide de *fichiers texte* ayant l'extension ".m" appelés "*fichiers .m*". Il existe deux types de fichiers .m :

- les **fichiers de commandes** (encore appelés *scripts*)
- les **fichiers de fonctions**.

Tous les fichiers ayant un nom avec le suffixe .m situés dans le répertoire courant ou dans les répertoires connus de MATLAB (chemins donnés par la commande 'path') seront exécutables.

Depuis MATLAB, on peut lancer l'éditeur intégré :

<File ...New ...M-File> ou <File...Open M-File...>

### 8. Fichier de commandes (script)

C'est un fichier texte, contenant une séquence quelconque de commandes MATLAB (telles qu'on les écrirait dans le mode interactif) → voir un exemple dans la suite : programme "ESSA1.M"

Une fois que le fichier a été créé par l'éditeur, on peut en lancer l'exécution à partir de la fenêtre de commande de MATLAB en entrant simplement son nom (l'extension .m est facultative).

### 9. Fichier de fonctions

Le deuxième type de fichiers '.m' de MATLAB est celui des **fonctions** auxquelles on peut passer des *paramètres d'entrée* et qui renvoient des *résultats* (comme les fonctions du langage C).

Un fichier fonction débute par le mot '*function*', suivi d'une déclaration de fonction :

```
function resultat = Nom_fonction(paramètres)
... suite quelconque de commandes Matlab.
```

☛ Le nom de la fonction doit obligatoirement être le même que celui du fichier associé !

La fonction s'utilise en plaçant un membre de gauche recevant le (ou les) résultats :

```
t = nom_fonction(x, y, z);
```

Le membre de gauche n'est pas obligatoire: on peut se contenter d'exploiter uniquement les effets produits par la fonction (exemple : affichage d'une courbe) sans récupérer les valeurs renvoyées.

**Exemple:** la fonction (standard) '*mean*' de Matlab est implémentée dans le fichier texte '*mean.m*' dont le contenu est le suivant (récupérable par '*type mean*') :

```
function y = mean(x)
% MEAN      Average or mean value. For vectors, MEAN(X) is the mean
%           value of the elements in vector X. For matrices, MEAN(x)
%           is a row vector containing the mean value of each column.

[m,n] = size(x);
if m == 1
    m = n;
end
y = sum(x) / m;
```

Ici, comme l'expliquent les lignes de commentaire %, qui seront affichées par le '*help mean*', x est le paramètre d'entrée (matriciel), '*mean*' le nom d'appel de la fonction, y le résultat (vectoriel) rendu représentant ici la moyenne par colonnes de la matrice x.

Notez la structure de contrôle du '*if*'.

- **Exercice:** - entrer le programme suivant (pouvant servir de *modèle de programme*)  
- sauvegarder sous "essai.m" .

```
% ESSAI.M
% My first program in MatLab
%
```

} La commande "help essai" affichera cet en-tête

```
clear all
close all
```

← efface toutes les variables en mémoire

← referme toutes les fenêtres

```
% Constantes
% -----
Npts=256;
nu0=0.05;
n=0:Npts-1;
```

} définition des constantes pour usage symbolique

```
% Parametres interactifs
% -----
m=input('m = ?');
```

← entrée d'un paramètre au clavier (essayer 0.1)

```
% Signaux
% -----
x1=exp(-m*2*pi*nu0*n).*sin(2*pi*nu0*n);
x2=exp(-m*2*pi*nu0*n).*cos(2*pi*nu0*n);
```

← noter l'utilisation du point '.' pour le produit terme à terme

```
% Trace des courbes
% -----
figure
subplot(2,1,1)
plot(n,x1,n,x2,'r');
title(['x1[n] ; Npts=', int2str(Npts),...
      ' ; m = ', num2str(m)]);
xlabel('n');
```

← ouverture d'une nouvelle fenêtre de tracé

← "subplot()" : plusieurs tracés dans une même fenêtre

← tracé de 2 courbes dans un même subplot

← valeurs entières ou réelles placées dans le titre (noter le caractère "suite" constitué par "...")

← zone de texte disponible sous l'axe des abscisses

```
subplot(2,1,2)
plot(x1,x2,'b')
title ('x2=f(x1)');
xlabel('x1');
ylabel('x2');
pause
```

← maintien du graphique à l'écran

```
% Pointage sur les courbes
% -----
[xx,yy]=ginput;
disp('Abscisses :'); xx
disp('Ordonnées:'); yy
```

← récupération de coordonnées avec la souris (valeurs mises dans un vecteur d'abscisses et un vecteur d'ordonnées); fin d'entrée : <return>.



**SOMMAIRE THEMATIQUE DES FONCTIONS  
STANDARD MATLAB**

## Operators and Special Characters

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

Controlling the Command Window	
clc	Clear command window.
echo	Echo commands inside script files.
format	Set output format.
home	Send cursor home.
more	Control paged output in command window.

Starting and Quitting from MATLAB	
matlabrc	Master startup M-file.
quit	Terminate MATLAB.
startup	M-file executed when MATLAB is invoked.

## General Purpose Commands

Managing Commands and Functions	
demo	Run demos.
help	Online documentation.
info	Information about MATLAB and The MathWorks.
lookfor	Keyword search through the help entries.
path	Control MATLAB's search path.
type	List M-file.
what	Directory listing of M-, MAT- and MEX-files.
which	Locate functions and files.

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

Working with Files and the Operating System	
cd	Change current working directory.
delete	Delete file.
diary	Save text of MATLAB session.
dir	Directory listing.
getenv	Get environment value.
unix	Execute operating system command; return result.
!	Execute operating system command.

- 1 -

## Elementary Matrices and Matrix Manipulation

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

Special Variables and Constants	
ans	Most recent answer.
computer	Computer type.
eps	Floating point relative accuracy.
flops	Count of floating point operations.
i, j	Imaginary unit.
inf	Infinity.
NaN	Not-a-Number.
nargin	Number of function input arguments.
nargout	Number of function output arguments.
pi	3.1415926535897....
realmax	Largest floating point number.
realmin	Smallest floating point number.

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

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.

## Elementary Functions

Elementary Math Functions	
abs	Absolute value.
acos	Inverse cosine.
acosh	Inverse hyperbolic cosine.
angle	Phase angle.
asin	Inverse sine.
asinh	Inverse hyperbolic sine.
atan	Inverse tangent.
atan2	Four quadrant inverse tangent.
atanh	Inverse hyperbolic tangent.
ceil	Round towards plus infinity.
conj	Complex conjugate.
cos	Cosine.
cosh	Hyperbolic cosine.
exp	Exponential.
fix	Round towards zero.
floor	Round towards minus infinity.
imag	Complex imaginary part.
log	Natural logarithm.
log10	Common logarithm.
real	Complex real part.
rem	Remainder after division.
round	Round towards nearest integer.
sign	Signum function.
sin	Sine.
sinh	Hyperbolic sine.
sqrt	Square root.
tan	Tangent.
tanh	Hyperbolic tangent.

## Specialized Matrices

Specialized Matrices	
compan	Companion matrix.
hadamard	Hadamard matrix.
hankel	Hankel matrix.
hilb	Hilbert matrix.
invhilb	Inverse Hilbert matrix.
magic	Magic square.
pascal	Pascal matrix (see online help).
rosser	Classic symmetric eigenvalue test problem.
toeplitz	Toeplitz matrix.
vander	Vandermonde matrix.
wilkinson	Wilkinson's eigenvalue test matrix.

- 2 -



## Matrix Functions - Numerical Linear Algebra

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

Linear Equations	
chol	Cholesky factorization.
inv	Matrix inverse.
lsqcov	Least squares in the presence of known covariance.
lu	Factors from Gaussian elimination.
nnls	Non-negative least-squares.
pinv	Pseudoinverse.
qr	Orthogonal-triangular decomposition.
\ and /	Linear equation solution.

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

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

## Specialized Math Functions

Specialized Math Functions	
bessel	Bessel function.
besselh	Hankel function.
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.
gamma	Gamma function.
gammainc	Incomplete gamma function.
gammainv	Logarithm of gamma function.
log2	Dissect floating point numbers.
pow2	Scale floating point numbers.
rat	Rational approximation.
rats	Rational output.

Logical Functions	
all	True if all elements of vector are true.
any	True if any element of vector is true.
exist	Check if variables or functions exist.
find	Find indices of non-zero elements.
finite	True for finite elements.
isempty	True for empty matrix.
isieee	True for IEEE floating point arithmetic.
isinf	True for infinite elements.
isnan	True for Not-A-Number.
issparse	True for sparse matrix.
isstr	True for text string.

-3-

## Language Constructs and Debugging

MATLAB as a Programming Language	
eval	Execute string with MATLAB expression.
feval	Execute function specified by string.
function	Add new function.
global	Define global variable.
nargchk	Validate number of input arguments.

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

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.

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

## Data Analysis and Fourier Transform Functions

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

Finite Differences	
del2	Five-point discrete Laplacian.
diff	Difference function and approximate derivative.
gradient	Approximate gradient (see online help).

Correlation	
corrcoef	Correlation coefficients.
cov	Covariance matrix.

Filtering and Convolution	
conv	Convolution and polynomial multiplication.
conv2	Two-dimensional convolution (see online help).
deconv	Deconvolution and polynomial division.
filter	One-dimensional digital filter (see online help).
filter2	Two-dimensional digital filter (see online help).

Fourier Transforms	
abs	Magnitude.
angle	Phase angle.
cp1xpair	Sort numbers into complex conjugate pairs.
fft	Discrete Fourier transform.
fft2	Two-dimensional discrete Fourier transform.
fftshift	Move zeroth lag to center of spectrum.
ifft	Inverse discrete Fourier transform.
ifft2	Two-dimensional inverse discrete Fourier transform.
nextpow2	Next higher power of 2.
unwrap	Remove phase angle jumps across 360° boundaries.

-4-



## General Purpose Graphics Functions

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

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

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

Handle Graphics Operations	
delete	Delete object.
drawnow	Flush pending graphics events.
get	Get object properties.
reset	Reset object properties.
set	Set object properties.

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

Hardcopy and Storage	
orient	Set paper orientation.
print	Print graph or save graph to file.
printopt	Configure local printer defaults.

Movies and Animation	
getframe	Get movie frame.
movie	Play recorded movie frames.
moviein	Initialize movie frame memory.

Miscellaneous	
ginput	Graphical input from mouse.
ishold	Return hold state.

## Two Dimensional Graphics

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

Specialized X-Y Graphs	
bar	Bar graph.
compass	Compass plot.
errorbar	Error bar plot.
feather	Feather plot.
fplot	Plot function.
hist	Histogram plot.
polar	Polar coordinate plot.
rose	Angle histogram plot.
stairs	Stairstep plot.

-5-

## Three Dimensional Graphics

Line and Area Fill Commands	
fill3	Draw filled 3-D polygons in 3-D space.
plot3	Plot lines and points in 3-D space.

Contour and Other 2-D Plots of 3-D Data	
clabel	Contour plot elevation labels.
contour	Contour plot.
contour3	3-D contour plot.
contourc	Contour plot computation (used by contour).
image	Display image.
pcolor	Pseudocolor (checkerboard) plot.
quiver	Quiver plot.

Surface and Mesh Plots	
mesh	3-D mesh surface.
meshc	Combination mesh/contour plot.
meshz	3-D Mesh with zero plane.
slice	Volumetric visualization plot (see online help).
surf	3-D shaded surface.
surfc	Combination surf/contour plot.
surf1	3-D shaded surface with lighting.
waterfall	Waterfall plot (see online help).

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

## Color Control and Lighting Model Functions

Color Controls	
caxis	Pseudocolor axis scaling.
colormap	Color lookup table.
shading	Color shading mode.

Color Maps (see online help)	
bone	Gray-scale with a tinge of blue color map.
cool	Shades of cyan and magenta color map.
copper	Linear copper-tone color map.
flag	Alternating red, white, blue, and black color map.
gray	Linear gray-scale color map.
hsv	Hue-saturation-value color map.
hot	Black-red-yellow-white color map.
pink	Pastel shades of pink color map.

Color Map Related Functions	
brighten	Brighten or darken color map.
hsv2rgb	Hue-saturation-value to red-green-blue conversion.
rgb2hsv	Red-green-blue to hue-saturation-value conversion.
rgbplot	Plot color map.
spinmap	Spin color map.

Lighting Models	
diffuse	Diffuse reflectance.
specular	Specular reflectance.
surf1	3-D shaded surface with lighting.
surfnorm	Surface normals.



## Polynomial and Interpolation Functions

Polynomials	
conv	Multiply polynomials.
deconv	Divide polynomials.
poly	Construct polynomial with specified roots.
polyder	Differentiate polynomial (see online help).
polyfit	Fit polynomial to data.
polyval	Evaluate polynomial.
polyvalm	Evaluate polynomial with matrix argument.
residue	Partial-fraction expansion (residues).
roots	Find polynomial roots.

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

## Function Functions

Function Functions - Nonlinear Numerical Methods	
fmin	Minimize function of one variable.
fmins	Minimize function of several variables.
fplot	Plot function.
fzero	Find zero of function of one variable.
ode23	Solve differential equations, low order method.
ode45	Solve differential equations, high order method.
quad	Numerically evaluate integral, low order method.
quad8	Numerically evaluate integral, high order method.

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

Norm, Condition Number, and Rank	
condst	Estimate 1-norm condition.
normest	Estimate 2-norm.
sprank	Structural rank.

Miscellaneous	
spaugment	Form least squares augmented system.
sparsm	Set parameters for sparse matrix routines.
symsfact	Symbolic factorization analysis.

## Sparse Matrix Functions

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

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

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

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

-7-

## Sound Processing Functions

General Sound Functions	
saxis	Sound axis scaling.
sound	Convert vector into sound.

SPARCstation-specific Sound Functions	
auread	Read Sun audio file (see online help).
auwrite	Write Sun audio file (see online help).
lin2mu	Linear to mu-law conversion (see online help).
mu2lin	Mu-law to linear conversion (see online help).

## Character String Functions

General	
abs	Convert string to numeric values.
eval	Execute string with MATLAB expression.
isstr	True for string.
setstr	Convert numeric values to string.
str2mat	Form text matrix from individual strings.
string	About character strings in MATLAB.

String Comparison	
lower	Convert string to lowercase.
strcmp	Compare strings.
upper	Convert string to uppercase.

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

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

## Low-level File I/O Functions

File Opening and Closing	
fclose	Close file.
fopen	Open file.

Unformatted I/O	
fread	Read binary data from file.
fwrite	Write binary data to file.

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

File Positioning	
ferror	Inquire file I/O error status.
frewind	Rewind file.
fseek	Set file position indicator.
ftell	Get file position indicator.

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



This section contains detailed descriptions of all **Signal Processing Toolbox functions**. It begins with a list of functions grouped by subject area and continues with the reference entries in alphabetical order. Information is also available through the online Help facility.

### Waveform Generation

diric	Dirichlet (periodic sinc) function.
sawtooth	Sawtooth and triangle wave function.
sinc	Sinc or $\sin(\pi x)/\pi x$ function.
square	Square wave function.

### Filter Analysis/Implementation

abs	Magnitude.
angle	Phase angle.
conv	Convolution.
fftfilt	Overlap-add filter implementation.
filter	Direct filter implementation.
filtfilt	Zero-phase version of filter.
filtic	Filter initial conditions.
freqs	Laplace transform frequency response.
freqspace	Frequency spacing for frequency response.
freqz	z-transform frequency response.
grpdelay	Group delay.
impz	Impulse response of digital filter.
unwrap	Unwrap phase.
zplane	Discrete pole-zero plot.

### IIR Filter Design - Classical and Direct

besself	Bessel filter design.
butter	Butterworth filter design.
cheby1	Chebyshev type I filter design.
cheby2	Chebyshev type II filter design.
ellip	Elliptic filter design.
yulewalk	Yule-Walker filter design.

### IIR Filter Order Selection

buttord	Butterworth filter order selection.
cheb1ord	Chebyshev type I filter order selection.
cheb2ord	Chebyshev type II filter order selection.
ellipord	Elliptic filter order selection.

### FIR Filter Design

fir1	Window-based FIR filter design - standard response.
fir2	Window-based FIR filter design - arbitrary response.
firls	Least-squares FIR filter design.
intfilt	Interpolation FIR filter design.
remez	Parks-McClellan optimal FIR filter design.
remezord	Parks-McClellan filter order estimation.

### Transforms

czt	Chirp z-transform.
dct	Discrete cosine transform.
dftmtx	Discrete Fourier transform matrix.
fft	Fast Fourier transform.
fftshift	Swap halves of transformed vectors.
hilbert	Hilbert transform.
idct	Inverse discrete cosine transform.
ifft	Inverse fast Fourier transform.

### Windows

bartlett	Bartlett window.
blackman	Blackman window.
boxcar	Rectangular window.
chebwin	Chebyshev window.
hamming	Hamming window.
hanning	Hanning window.
kaiser	Kaiser window.
triang	Triangular window.

- 9 -

### Statistical Signal Processing

cohere	Coherence function estimation.
corrcoef	Correlation coefficients.
cov	Covariance matrix.
csd	Cross spectrum estimation.
psd	Power spectrum estimation.
tfe	Transfer function estimation.
xcorr	Cross-correlation function.
xcov	Covariance function.

### Parametric Modeling

invfreqs	Analog filter fit to frequency response.
invfreqz	Discrete filter fit to frequency response.
levinson	Levinson's recursion.
lpc	Linear prediction coefficients.
prony	Prony's discrete filter fit to time response.
stmc	Linear model using Steiglitz-McBride iteration.

### Other

conv2	2-D convolution.
cxlpair	Order vector into complex conjugate pairs.
detrend	Linear trend removal.
fft2	2-D fast Fourier transform.
fftshift	Swap quadrants of arrays.
filter2	2-D filtering.
ifft2	Inverse 2-D fast Fourier transform.
polystab	Polynomial stabilization.
strips	Strip plot.
xcorr2	2-D cross-correlation.

### Frequency Translation

lp2bp	Lowpass to bandpass transformation.
lp2bs	Lowpass to bandstop transformation.
lp2hp	Lowpass to highpass transformation.
lp2lp	Lowpass to lowpass transformation.

### Filter Discretization

bilinear	Bilinear transformation.
----------	--------------------------

### Specialized Operations

ccaps	Complex cepstrum.
decimate	Lowpass FIR decimation.
deconv	Deconvolution.
demod	Demodulation.
interp	Lowpass interpolation.
medfilt1	One-dimensional median filtering.
modulate	Modulation.
rceps	Real cepstrum.
resample	Resample sequence with new sampling rate.
spectgram	Spectrogram.
spline	Cubic spline interpolation.
vco	Voltage controlled oscillator.

### Analog Prototype Design

besselap	Bessel filter prototype.
buttap	Butterworth filter prototype.
cheb1ap	Chebyshev type I filter (passband ripple) prototype.
cheb2ap	Chebyshev type II filter (stopband ripple) prototype.
ellipap	Elliptic filter prototype.

### Linear System Transformations

convmtx	Convolution matrix.
poly2rc	Polynomial to reflection coefficients conversion.
rc2poly	Reflection coefficients to polynomial conversion.
residuez	Partial fraction expansion.
sos2ss	Second-order sections to state-space conversion.
sos2tf	Second-order sections to transfer function conversion.
sos2zp	Second-order sections to zero-pole conversion.
ss2sos	State-space to second-order sections conversion.
ss2tf	State-space to transfer function conversion.
ss2zp	State-space to zero-pole conversion.
tf2ss	Transfer function to state-space conversion.
tf2zp	Transfer function to zero-pole conversion.
zp2sos	Zero-pole to second-order sections conversion.
zp2ss	Zero-pole to state-space conversion.
zp2tf	Zero-pole to transfer function conversion.

- 10 -