Introduction à MATLAB

1. Généralités

1.1) Qu'est-ce que MATLAB?

Logiciel interactif de calcul scientifique:

- fonctions mathématiques usuelles
- calcul matriciel
- racines d'un polynôme
- équations différentielles
- intégration numérique
- graphiques 2D & 3D
- etc.

1ère génération (Fortran): MATrix LABoratory
(LINPACK, EISPACK)

2 ième génération (C):
PC-MATLAB (MS-Dos)
MacMATLAB (Macintosh)
PRO-MATLAB (Sun, Vax)

+ extensions:
Signal Processing Toolbox
Control System Toolbox
Optimization Toolbox...
SIMULINK

1.2) Quelques atouts de MATLAB

ordinateur = supercalculatrice

apprentissage facile, commandes intuitives

moins de C, Pascal, Fortran, Basic, etc.

inclusion des graphiques dans les documents

interaction avec des programmes externes

1.3) Entrer et sortir de MATLAB

matlab

< M A T L A B (R)>
(c) Copyright 1984-94 The MathWorks, Inc.
All Rights Reserved
Version 4.2c
Dec 31 1994

Commands to get started: intro, demo, help help
Commands for more information: help, whatsnew, info, subscribe

>>

•••
(SESSION MATLAB)
•••

>> quit *(exit)*

0 flop(s).

1.4) Mode interactif

```
>> 1900/81

ans =

        23.4568

>> cos(pi/4)

ans =

         0.7071
```

2. Variables MATLAB

VARIABLE: matrice $m \times n$, $m, n \geq 1$

- vecteur: $1 \times n$, $n \geq 1$
- scalaire: 1×1

Nombres OU chaînes de caractères

nom_de_variable: aBz23k...

- 1er caractère = lettre
- 19 premiers caractères reconnus
- A ≠ a

2.1) Initialisation explicite

```
>> a=[1 2 3; 4 5 6; 7 8 9]

    a =

        1    2    3
        4    5    6
        7    8    9

>> x=['ab' 'c']

    x =

    abc

>> y= 'matlab'

        y=matlab
```

- inhiber la rétroaction visuelle: `>> r=[10 11 12];`
- concaténation: `>> a=[a;r]`

```
  a =

      1    2    3
      4    5    6
      7    8    9
     10   11   12
```

2.2) Initialisation avec énoncés MATLAB

```
>> x=[-1.3 sqrt(3) (1+2+3)]

      x =

        -1.3000    1.7321    6.0000
```

- continuer l'énoncé sur la ligne suivante

```
>> s=1-2+3-4+5...
   -6+7-8+9+10

      s =

          15
```

2.3) Initialisation par fichier exécutable ".m"

- éditer un fichier ASCII "*nom_de_fichier*.m"

```
A=1;
B=2;
  ...
Z=26;
```

- dans la session MATLAB, exécuter "*nom_de_fichier*":

```
>> nom_de_fichier
>>
```

2.4) Initialisation par fichier binaire ".mat"

- constituer un fichier binaire "*nom_de_fichier*.mat"

- lire le fichier "*nom_de_fichier*.mat" dans la session MATLAB

```
>>load nom_de_fichier
>>
```

- toutes les variables énumérées dans "*nom_de_fichier*.mat" sont maintenant présentes dans l'espace de travail.

2.5) Formes particulières d'initialisation

Incrément = 1

```
>> a=1:5

      a =

          1    2    3    4    5
```

Incrément ≠ 1

```
>> x=0:pi/4:pi
>> x=5:-1:1

x =x =

    0    0.7854    1.5708    2.3562    3.1416     5     4     3
2     1
```

Nombre de points spécifique

```
>> x=linspace(-pi,pi,3)

x =

  -3.1416         0    3.1416
```

Matrices particulières

```
>> x=ones(2,3)

x =

     1     1     1
     1     1     1

>> y=zeros(2)

y =

     0     0
     0     0

>> z=eye(2)

z =

     1     0
     0     1
```

2.6) Indiçage matriciel

```
a =

     1     2     3
     4     5     6
     7     8     9

>> a(3,3)=a(1,3)+a(3,1)

a =

     1     2     3
     4     5     6
     7     8    10

>> a(1:2,3)

ans =

     3
     6

>> a(2,:)

ans =

     4     5     6
```

2.7) Espace de travail

```
>> who

Your variables are:

a       b       s       w       z
ans     r       theta   x
```

```
>> whos
```

Name	Size	Total	Complex
a	1 by 2	2	No
ans	1 by 1	1	No
b	1 by 1	1	No
r	1 by 1	1	No
s	1 by 1	1	No
theta	1 by 1	1	No
w	1 by 1	2	Yes
x	2 by 3	6	No
z	1 by 1	1	No

Grand total is (16 * 8) = 128 bytes,

Éliminer des variables de l'espace de travail

```
>>clear x
>>
```

Sauver l'espace de travail

```
>>save
Saving to: matlab.mat
>>
```

Sauver des variables dans "*nom_de_fichier*.mat"

```
>>save nom_de_fichier X Y Z
>>
```

Lire l'espace de travail

```
>>load
Loading from: matlab.mat
>>
```

Ajouter/modifier des variables

```
>>load nom_de_fichier
>>
```

2.8) Nombres

Notations: 3 -99 0.001 -1.6E-20 1.4e-3

Format IEEE: 16 chiffres signés,

$$10^{-308} \leq |x| \leq 10^{308}$$

Résultat infini/indéterminé

```
>> s=1/0
Warning: Divide by zero
     s =
       Inf
>> s=0/0
Warning: Divide by zero
     s =
       NaN
```

2.9) Nombres complexes

$i = \sqrt{-1}$ ou $j = \sqrt{-1}$ pré-initialisées

```
>> a=3+4*i
     a =
       3.0000 + 4.0000i
>> a=3+4*j;
>> w=r*exp(j*theta);
>>a=[1 2 3 4]+i*[5 6 7 8];
>> a=[1+5*j 2+6*i 3+7*j 4+8*i];
```

Mais i et j non-réservées

```
>> i=sqrt(-1)

      i =

            0 + 1.0000i
```

Nb. complexe $\leftrightarrow$ nombre réel dans $+$, $\times$, $/$, etc.

2.10) Affichage des résultats

```
>> a=[4/3 1.2345e-6]

      a =

         1.3333    0.0000

>> format short
>> a

      a =

         1.3333    0.0000

>> format long
>> a

      a =

         1.33333333333333   0.00000123450000

>> format short e
>> a

      a =

         1.3333e+00   1.2345e-06


>> format long e
>> a

      a =
```

```
        1.333333333333333e+00     1.234500000000000e-06

>> format hex
>> a

      a =

        3ff5555555555555   3eb4b6231abfd271

>> format +
>> a

      a =

      ++

>> format bank
>> a

      a =

            1.33          0.00
```

3. Fonctions MATLAB

Plus de 500 fonctions MATLAB prédéfinies

+ fonctions “usager” (fichiers exécutables “.m”)

Composition de fonctions:

```
>>x=sqrt(log(z));
```

Fonctions multiarguments:

```
>>theta=atan2(y,x);
```

Fonctions multirésultats:

```
>>[y i]=max(x);
```

La variable à droite n'est jamais altérée

3.1) Aide dans MATLAB

Liste des sujets "help":

```
>>help
```

"help" spécifique:

```
>> help cos
```

```
COS   COS(X) is the cosine of the elements of X.
```

Manuel "on-line" avec Netscape:

file:/a/natashquan/gel/natashquan/solaris2/matlab4.2/toolbox/matlab/doc/ReferenceTOC.html#Main

3.2) Caractères et variables réservés

Caractères spéciaux	
=	assignment statement
[	used to form vectors and matrices
]	see [
(	arithmetic expression precedence
)	see (
.	decimal point
...	continue statement to the next line
,	separate subscripts and function arguments
;	end rows, suppress printing
%	comments
:	subscripting, vector generation
!	execute operating system command

Valeurs spéciales	
ans	answer when expression is not assigned
eps	floating point precision
pi	π
i,j	$\sqrt{-1}$
Inf	∞
NaN	Not-a-Number
clock	wall clock
date	date
flops	floating point operation count
nargin	number of function input arguments
nargout	number of function output arguments

3.3) Fonctions d'usage général

Fonctions d'usage général	
help	help facility
demo	demo
who	list variables in memory
what	list M-files on disk
size	row and column dimensions
length	vector length
clear	clear workspace
computer	type of computer
^C	local abort
quit	quit
exit	exit

3.4) Opérateurs arithmétiques “Matrix” et “Array”

Opérateurs “Matrix”	
+	addition
-	subtraction
*	multiplication
/	right division
\	left division
^	power
‘	conjugate transpose

Arithmétique matricielle usuelle

addition: A(m x n) + B(m x n)

multiplication: A(m x n) x B(n x m)division à gauche/droite:

X=A\B est solution de A*X=B

X=B/A est solution de X*A=B

- un scalaire s’applique à tous les éléments

Opérateurs “Array”	
+	addition
-	subtraction
.*	multiplication
./	right division
.\	left division
.^	power
.‘	transpose

Arithmétique élément-par-élément

addition: A(m x n) + B(m x n)

multiplication: A(m x n) x B(m x n)

division à gauche/droite:

X=A.\B: X(i,j)=B(i,j)/A(i,j)

X=A./B: X(i,j)=A(i,j)/B(i,j)

- un scalaire s'applique à tous les éléments

3.4.1 Opérateurs "Matrix"

```
a =                     b =

   1    2                  5    6    7
   3    4                  8    9   10

>> a*b

ans =

   21   24   27
   47   54   61a =

  1.0000 + 1.0000i   2.0000 + 2.0000i
  3.0000 + 3.0000i   4.0000 + 4.0000i

>> a'

ans =

  1.0000 - 1.0000i   3.0000 - 3.0000i
  2.0000 - 2.0000i   4.0000 - 4.0000i
```

3.4.2 Opérateurs "Array"

```
x =                     y =

   1    2                  5    6
   3    4                  7    8


>> x.*y

ans =

    5   12
   21   32

>> a.'
```

```
ans =

   1.0000 + 1.0000i   3.0000 + 3.0000i
   2.0000 + 2.0000i   4.0000 + 4.0000i
```

3.5) Fonctions mathématiques et trigonométriques

Appliquées à chacun des éléments d'une variable

Fonctions mathématiques élémentaires	
abs	absolute value or complex magnitude
angle	phase angle
sqrt	square root
real	real part
imag	imaginary part
conj	complex conjugate
round	round to nearest integer
fix	round towards zero
floor	round towards $-\infty$
ceil	round towards ∞
sign	signum function
rem	remainder and modulus
exp	exponential base e
log	natural logarithm
log10	log base 10

Fonctions trigonométriques	
sin	sine
cos	cosine
tan	tangent
asin	arcsine
acos	arccosine
atan	arctangent
atan2	four quadrant arctangent
sinh	hyperbolic sine
cosh	hyperbolic cosine
tanh	hyperbolic tangent
asinh	hyperbolic arcsine
acosh	hyperbolic arccosine
atanh	hyperbolic arctangent

```
>> x=linspace(0,2*pi,4);
>> cos(x)

      ans =

         1.0000   -0.5000   -0.5000    1.0000

>> x=[10 10 10 10];
>> y=[1 2 3 4];
>> z=x.^y

      z =

            10        100       1000      10000

>> log10(z)

      ans =

         1.0000    2.0000    3.0000    4.0000
```

3.6) Opérations matricielles

Manipulations matricielles	
rot90	rotation
fliplr	flip matrix left-to-right
flipud	flip matrix up-and-down
diag	extract or create diagonal
tril	lower triangular part
triu	upper triangular part
reshape	reshape
.’	transposition
:	general rearrangement

Manipulations sur les colonnes	
max	maximum value
min	minimum value
mean	mean value
median	median value
std	standard deviation
sort	sorting
sum	sum of elements
prod	product of elements
cumsum	cumulative sum of elements
cumprod	cumulative product of elements
diff	approximate derivatives
hist	histogram
corrcoef	correlation coefficients
cov	covariance matrix
cplxpair	reorder into complex pairs

Matrices spéciales	
compan	companion
diag	diagonal
eye	identity
gallery	esoteric
hadamard	Hadamard
hankel	Hankel
hilb	Hilbert
invhilb	inverse Hilbert
linspace	linearly spaced vectors
logspace	logarithmically spaced vectors
magic	magic square
meshdom	domain for mesh plots
ones	constant
rand	random elements
toeplitz	Toeplitz
vander	Vandermonde
zeros	zero

Décompositions et factorisations	
balance	balanced form
backsub	backsubstitution
cdf2rdf	convert complex-diagonal to real-diagonal
chol	Cholesky factorization
eig	eigenvalues and eigenvectors
hess	Hessenberg form
inv	inverse
lu	factors from Gaussian elimination
nnls	nonnegative least-squares
null	null space
orth	orthogonalization
pinv	pseudoinverse
qr	orthogonal-triangular decomposition
qz	QZ algorithm
rref	reduced row echelon form
rsf2csf	convert real-Schur to complex-Schur
schur	Schur decomposition
svd	singular value decomposition

Fonctions matricielles élémentaires	
expm	matrix exponential
logm	matrix logarithm
sqrtm	matrix square root
funm	arbitrary matrix functions
poly	characteristic polynomial
det	determinant
trace	trace
kron	Kronecker tensor product

Condition d'une matrice	
cond	condition number in 2-norm
norm	1-norm, 2-norm, F-norm, ∞ norm
rank	rank
rcond	condition estimate

3.7) Fonctions texte et chaînes de caractères

Texte et chaînes de caractères	
abs	convert string to ASCII values
eval	evaluate text macro
num2str	convert number to string
int2str	convert integer to string
setstr	set flag indicating matrix is a string
sprintf	convert number to string
isstr	detect string variables
strcmp	compare string variables
hex2num	convert hexadecimal string to number

3.8) Autres fonctions

Traitement du signal	
abs	complex magnitude
angle	phase angle
conv	convolution
corrcoef	correlation coefficients
cov	covariance
deconv	deconvolution
fft	fast Fourier transform
fft2	2-D fast Fourier transform
ifft	inverse fast Fourier transform
ifft2	inverse 2-D fast Fourier transform
fftshift	swapt quadrants of matrices

Polynômes	
poly	characteristic polynomial
roots	polynomial roots - comp. matrix
roots1	polynomial roots - Laguerre
polyval	polynomial evaluation
polyvalm	matrix polynomial evaluation
conv	multiplication
deconv	division
residue	partial-fraction expansion
polyfit	polyonmial curve fitting

Solution d'équations différentielles	
ode23	2nd/3rd order Runge-Kutta method
ode45	4th/5th order Runge-Kutta method

Équations non linéaires et optimisation	
fmin	minimum of a function of one variable
fmins	minimum of a multivariable function (unconstrained nonlinear optim.)
fsolve	solution to a system of nonlinear eqns. (zeros of a multivariable function)
fzero	zero of a function of one variable

Interpolation	
spline	cubic spline
table1	1-D table look-up
table2	2-D table look-up

Intégration numérique	
quad	numerical function integration
quad8	numerical function integration

3.9) Fonctions “Usager”

Suite d’énoncés MATLAB écrits dans fichier ASCII “*nom_de_fichier*.m”

```
    ...
x=y*sin(z);
a=b+c;
x=x/a;
    ...
```

Appelée dans MATLAB comme toute autre fonction MATLAB

```
>> nom_de_fichier
```

3.9.1 Fonctions “usager” de type “script”

- fonction sans argument

- variables globales

```
%*****************************%
%*      pivot_x.m              *%
%*****************************%
echo off
x=[1 2 3 4]
x=fliplr(x)


>> pivot_x

x =

    1    2    3    4

x =

    4    3    2    1
```

3.9.2 Fonctions "usager" de type "function"

- fonction avec arguments i/o
- variables locales

```
%*******************************%
%*      plptnb.m                 *%
%*******************************%
% Retourne le plus petit de*%
% n1 ou n2                      *%
%*******************************%
function y= plptnb(n1,n2)
y= n2;
if n1<n2
       y=n1;
end;

>> a=-2; b=3;
>> plptnb(a,b)

ans =

   -2
```

Programmation et fichiers M	
input	get numbers from keyboard
keyboard	call keyboard as M-file
error	display error message
function	define function
eval	interpret text in variables
feval	evaluate function given by string
echo	enable command echoing
exist	check if variables exist
casesen	set case sensitivity
global	define global variables
startup	startup M-files
getenv	get environment string
menu	select item from menu
etime	elapsed time

Énoncés de contrôle	
if	conditionally execute statements
elseif	used with if
else	used with if
end	terminante if, for, while
for	repeat statements a number of times
while	do while
break	break out of for and while loops
return	return from functions
pause	pause until key press

Opérateurs relationnels et logiques			
<	less than	&	AND
<=	less than or equal	|	OR
>	greater than	~	NOT
>=	greater than or equal		
==	equal		
-=	not equal		

Fonctions relationnelles et logiques	
any	logical conditions
all	logical conditions
find	find array indices of logical values
exist	check if variables exist
isnan	detect NaN's
finite	detect infinities
isempty	detect empty matrices
isstr	detect string variables
strcmp	compare string variables

Fenêtre de commandes	
clc	clear command screen
home	home cursor
format	set output display format
disp	display matrix or text
fprintf	print formatted number
echo	enable command echoing

"Vectoriser" les calculs: >10 fois + rapide

```
angles=0:pi/9:pi;
for k=1:10
    y(k)=sin(angle(k));
end;
```

```
angles=0:pi/9:pi;
y=sin(angles)
```

Pré-allocation de la mémoire

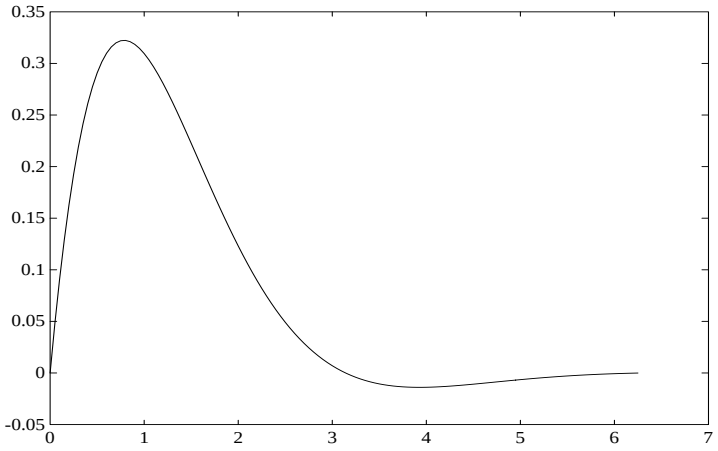
```
y=zeros(1,10);

angles=0:pi/9:pi;
for k=1:10
    y(k)=sin(angle(k));
end;
```

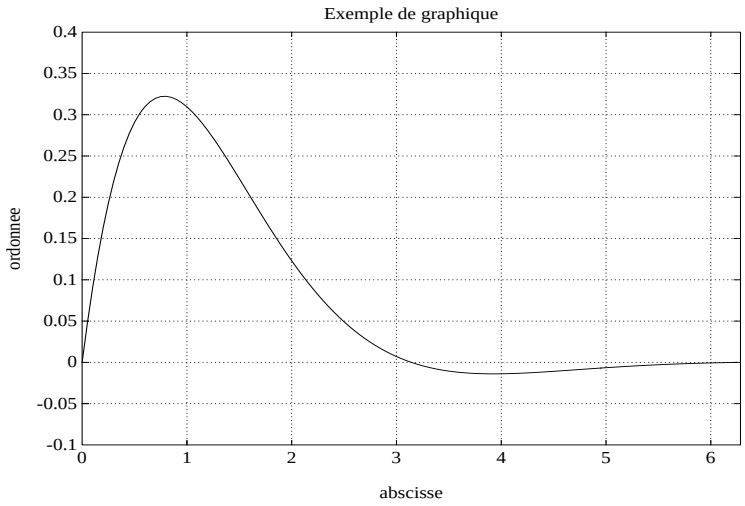
4. Graphiques

4.1) Exemples

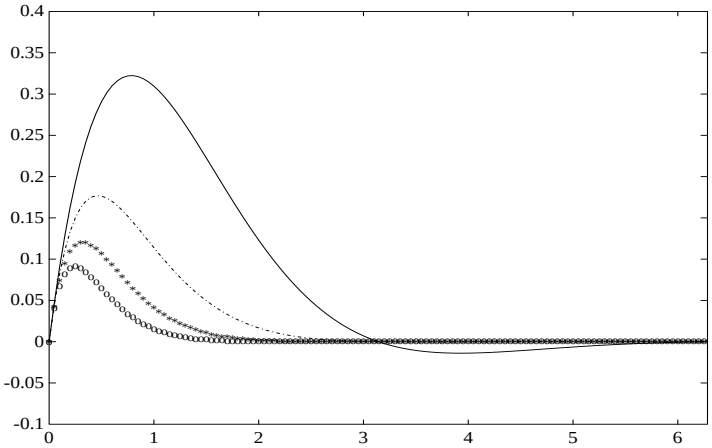
```
>> x=0:0.05:(2*pi);
>> y=exp(-x).*sin(x);
>>
>> plot(x,y);
```



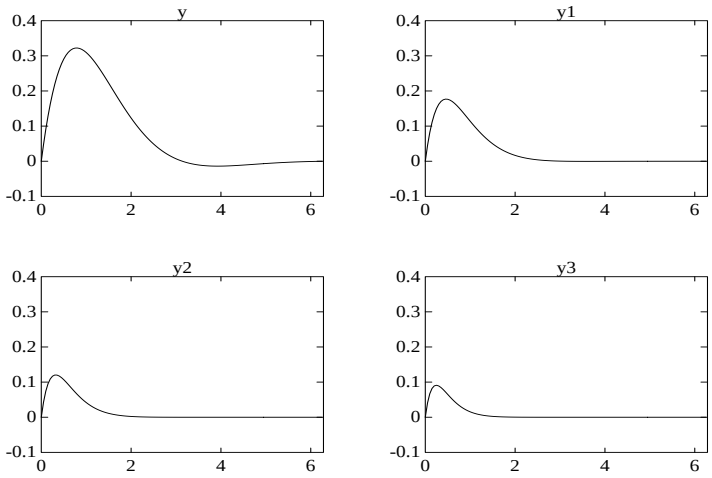
```
>> axis([0 (2*pi) -0.1 0.4]);
>> plot(x,y);
>>
>> title ('Exemple de graphique');
>> xlabel('abscisse');
>> ylabel('ordonnee');
>> grid;
```



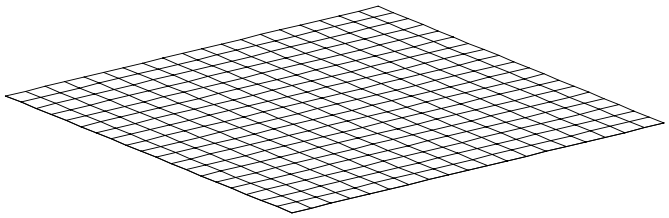

```
>> y1=exp(-2*x).*sin(x);
>> y2=exp(-3*x).*sin(x);
>> y3=exp(-4*x).*sin(x);
>>
>> plot(x,y,'-',x,y1,'-.',x,y2,'*',x,y3,'o');
```



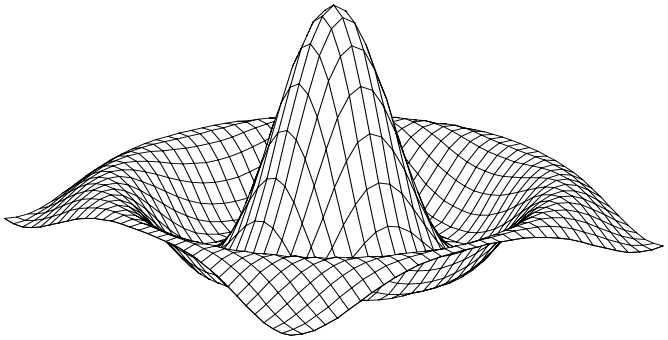
```
>> subplot(2,2,1), plot(x,y), title('y');
>> subplot(2,2,2), plot(x,y1), title('y1');
>> subplot(2,2,3), plot(x,y2), title('y2');
>> subplot(2,2,4), plot(x,y3), title('y3');
```




```
>> x=zeros(20);
>> mesh(x);
```



```
>> x=(-8:0.5:8);
>> y=x';
>> X=ones(y)*x;
>> Y=y*ones(x);
>> R=sqrt(X.^2+Y.^2)+eps;
>> Z=sin(R)./R;
>> mesh(Z);
```



4.2) Fonctions graphiques

Fonctions graphiques	
plot	linear X-Y plot
loglog	loglog X-Y plot
semilogx	semi-log X-Y plot
semilogy	semi-log X-Y plot
polar	polar plot
mesh	3-dimensional mesh surface
contour	contour plot
meshdom	domain for mesh plots
bar	bar charts
stairs	stairstep graphs
errorbar	add errorbars

Annotation des graphiques	
title	plot title
xlabel	x-axis label
ylabel	y-axis label
grid	draw grid lines
text	arbitrary positioned text
gtext	mouse-positioned text
ginput	graphics input

Contrôle de la fenêtre graphique	
axis	manual axis scaling
hold	hold plot on screen
shg	show graph on screen
clg	clear graph screen
subplot	split graph window

4.3) Impression et sauvegarde d'un graphique

4.3.1 Impression d'un graphique

Exemple (réseau SUN, salle 2117)

```
>> plot(x,y)
>> print -dps -Plj2117
```

4.3.2 Sauvegarde d'un graphique

```
>> plot(x,y)
>> print -dps nom_du_fichier.ps
>> print -deps nom_du_fichier.eps
```

5. Fichiers et programmes externes

Fichiers sur disque	
chdir	change current directory
delete	delete file
diary	diary of the session
dir	directory of files on disk
load	load variables from file
save	save variables on file
type	list function or file
what	show M-files on disk
fprintf	write to a file
pack	compact memory via save

MATLAB	MS-DOS	UNIX	VAX/VMS
chdir	chdir	cd	set default
delete	del	rm	delete
dir	dir	ls	dir
type	type	cat	type

5.1) Exécution d'un programme externe à MATLAB

- Commande du système d'exploitation

- Tout autre programme

...

>> !*nom_du_programme*

...

6. À retenir

- Variable MATLAB $\leftrightarrow$ matrice

- Opérateurs “matrix” et “array”

- Fichiers exécutables “.m” type “script” et “function”; vectoriser les énoncés

- Fichiers binaires “.mat”pour échange de données

- Extensions (toolboxes)

6.1) Accès à MATLAB

Réseau PC:

```
E:\>cd matlab
E:\MATLAB>matlab
```

Réseau Sun:

```
rlogin nat
setenv DISPLAY nom_de_ma_machine:0
matlab
```