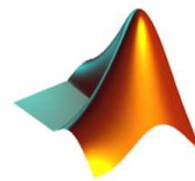


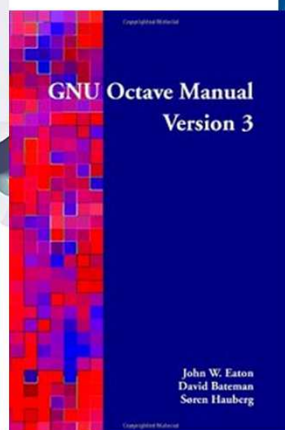
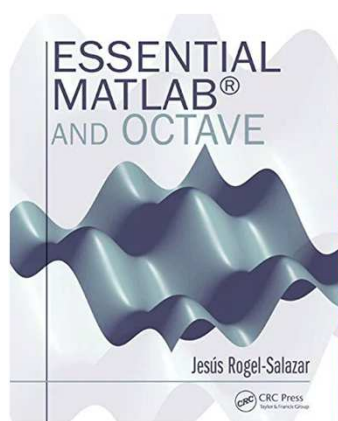


## Chapitre 1 : Introduction à Matlab / Octave

Dr Efstratios Rappos  
25 septembre 2017 – 02 octobre 2017

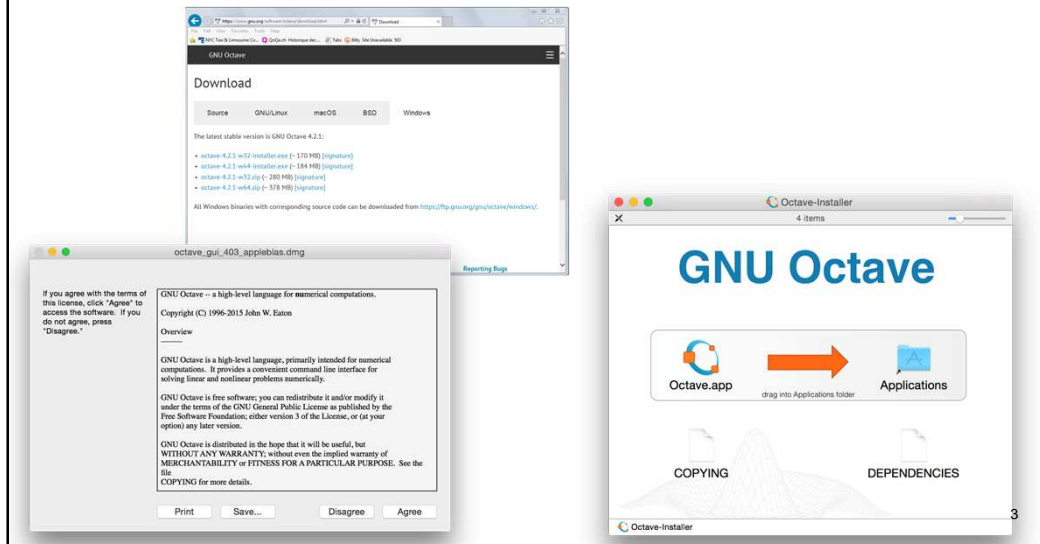


## Matlab & Octave

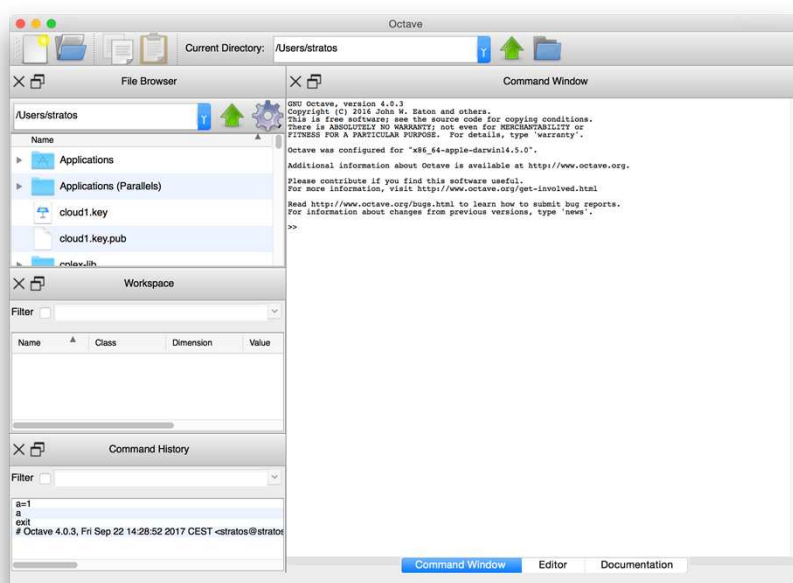


## Téléchargement et installation

- <https://www.gnu.org/software/octave/download.html>



## Téléchargement et installation



4

## Introduction à Matlab & Octave

---

- Vidéo : "Getting started with Matlab"
- <https://ch.mathworks.com/videos/getting-started-with-matlab-68985.html>
- E-book : "Experiments with Matlab"
- <https://ch.mathworks.com/moler/exm.html>

5

## Calculateur

---

EDU>> x=3	EDU>> x=sqrt(1+x)
x =	x =
3	1.6213
EDU>> x=sqrt(1+x)	EDU>> x=sqrt(1+x)
x =	x =
2	1.6191
EDU>> x=sqrt(1+x)	EDU>> x=sqrt(1+x)
x =	x =
1.7321	1.6184
EDU>> x=sqrt(1+x)	EDU>> x=sqrt(1+x)
x =	x =
1.6529	1.6181
EDU>> x=sqrt(1+x)	EDU>> x=sqrt(1+x)
x =	x =
1.6288	1.6181
EDU>> x=sqrt(1+x)	EDU>> x=sqrt(1+x)
x =	x =
1.6213	<b>1.6180 (Golden ratio)</b>

6

## Format de sortie

---

- Exemple:

```
EDU>> x= pi
```

(pour l'espacement des lignes, plus serré)

```
EDU>> format compact
```

```
EDU>> x
```

```
x =
    3.1416
```

```
EDU>> format long
```

```
EDU>> x
```

```
x =
    3.141592653589793
```

```
EDU>> format short
```

```
EDU>> x
```

```
x =
    3.1416
```

7

## Calculateur (2)

---

```
EDU>> format long
```

```
EDU>> x=3
```

```
x =
```

```
    3
```

```
EDU>> x=sqrt(1+x)
```

```
x =
```

```
    2
```

```
EDU>> x=sqrt(1+x)
```

```
x =
```

```
    1.732050807568877
```

```
EDU>> x=sqrt(1+x)
```

```
x =
```

```
    1.652891650281070
```

```
EDU>> x=sqrt(1+x)
```

```
x =
```

```
    1.628769980777233
```

```
EDU>> x=sqrt(1+x)
```

```
x =
```

```
    1.621348198499395
```

```
...
```

```
EDU>> x=sqrt(1+x)
```

```
x =
```

```
    1.618033988749901
```

```
EDU>> x=sqrt(1+x)
```

```
x =
```

```
    1.618033988749897
```

```
EDU>> x=sqrt(1+x)
```

```
x =
```

```
    1.618033988749895
```

```
EDU>> x=sqrt(1+x)
```

```
x =
```

```
    1.618033988749895
```

Note: répétition de l'opération avec ▲

8

## Calculateur (3)

---

```
EDU>> help sqrt
```

SQRT Square root.  
 SQRT(X) is the square root of the elements of X. Complex results are produced if X is not positive.

See also sqrtm, realsqrt, hypot.

Overloaded methods:  
 sym/sqrt

Reference page in Help browser  
 doc sqrt

```
EDU>> doc sqrt -> Browser
```

9

## Nombres et opérations

---

### Nombres

Peuvent être assignés à des variables.

Nombre réel:  $x = 42$

Nombre complexe:  $y = -3 + 2i$

### Opérations

^

\*, /, \

+, -

### Expressions

```
EDU>> 3*4^2+4/2
```

```
ans =
```

```
50
```

'ans' est le nom du résultat si aucun = ne figure dans l'expression

10

## Nombres et opérations

---

### Comparaisons

< , > , >= , <= , == , ~=  
Résultat est 0 ou 1

### Logique

|| or (ou)  
&& and (et)  
~ not (pas)

### Expressions

```
EDU>> 2<4 || 'a' == 'b'
ans =
     1
```

11

## Exercices

---

- Évaluez les expressions suivantes:  $43^2$  ,  $\sqrt[4]{32}$  ,  $\sin \frac{\pi}{3}$
  - Convertir une température de degrés Celsius en degrés Fahrenheit
  - Autre point fixe à calculer à la main:  $x = \frac{1}{\sqrt{1+x^2}}$
- Après combien d'itérations trouvez-vous le «golden ratio»?
- Même question que la précédente pour le point fixe:  $x = \cos(x)$

12

## Vecteurs et matrices

---

Matlab = «Matrix Laboratory»

### Vecteur

```
EDU>> x=[2; 4]
```

```
x =
```

```
2
```

```
4
```

### Matrice

```
EDU>> A=[2 3 ; -1 4]
```

```
A =
```

```
2    3
```

```
-1   4
```

### Multiplication A\*x

```
EDU>> A*x
```

```
ans =
```

```
16
```

```
14
```

13

## Vecteurs et matrices (2)

---

- Ajout d'éléments à un vecteur:

```
vec =
```

```
1    3    5    7
```

```
EDU>> vec(end+1)=4
```

```
vec =
```

```
1    3    5    7    4
```

14

## Vecteurs et matrices (3)

---

### Matrice A transposée

```
EDU>> A'
```

```
ans =
```

```
 2   -1
 3    4
```

### Vecteur x transposé

```
EDU>> x'
```

```
ans =
```

```
 2    4
```

### Éléments de la matrice

```
EDU>> A(1,2)
```

```
ans =
```

```
 3
```

```
EDU>> A(2,2)
```

```
ans =
```

```
 4
```

```
x =
```

```
 2
 4
```

```
A =
```

```
 2    3
-1    4
```

15

## Vecteurs et matrices (4)

---

### Vecteurs ligne

```
EDU>> y=0:2:10
```

```
y =
```

```
 0    2    4    6    8   10
```

```
EDU>> n=10
```

```
EDU>> y=0:n
```

```
y =
```

```
 0    1    2    3    4    5    6    7    8    9   10
```

### Concaténation de matrices

```
B =
```

```
 0    1
 1    2
```

```
EDU>> cat(1,A,B)
```

```
ans =
```

```
 2    3
-1    4
 0    1
 1    2
```

```
A =
```

```
 2    3
-1    4
```

16



## Vecteurs et matrices (5)

### Concaténation de matrices (suite)

```
EDU>> cat(2,A,B)
```

```
ans =
```

```
     2     3     0     1
    -1     4     1     2
```

```
EDU>> cat(3,A,B)
```

```
ans(:, :, 1) =
```

```
     2     3
    -1     4
```

```
ans(:, :, 2) =
```

```
     0     1
     1     2
```

```
A =
```

```
     2     3
    -1     4
```

```
B =
```

```
     0     1
     1     2
```

17

## Vecteurs et matrices (6)

### Génération de matrices

```
EDU>> Z=zeros(3,4)
```

```
Z =
```

```
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

```
EDU>> E=ones(4,3)
```

```
E =
```

```
     1     1     1
     1     1     1
     1     1     1
     1     1     1
```

```
EDU>> I=eye(3,3)
```

```
I =
```

```
     1     0     0
     0     1     0
     0     0     1
```

```
EDU>> x=[2 4 6]
```

```
x =
```

```
     2     4     6
```

```
EDU>> ones(4,1)*x
```

```
ans =
```

```
     2     4     6
     2     4     6
     2     4     6
     2     4     6
```

```
EDU>> x(ones(4,1), :)
```

```
ans =
```

```
     2     4     6
     2     4     6
     2     4     6
     2     4     6
```

18

## Vecteurs et matrices (7)

### Génération de matrices (suite)

```
EDU>> R=rand(2,3)
```

```
R =
    0.8147    0.1270    0.6324
    0.9058    0.9134    0.0975
```

```
A =
     2     3
    -1     4
```

### Arithmétique avec les matrices

```
EDU>> A+B
```

```
ans =
     2     4
     0     6
```

```
B =
     0     1
     1     2
```

```
EDU>> A-B
```

```
ans =
     2     2
    -2     2
```

```
EDU>> A*B
```

```
ans =
     3     8
     4     7
```

19

## Vecteurs et matrices (8)

### Arithmétique avec les matrices

```
EDU>> A/B
```

```
ans =
    -1     2
     6    -1
```

```
A =
     2     3
    -1     4
```

```
EDU>> A^2
```

```
ans =
     1    18
    -6    13
```

```
B =
     0     1
     1     2
```

### *multiplication élément par élément*

```
EDU>> A.*B
```

```
ans =
     0     3
    -1     8
```

### *produit scalaire*

```
EDU>> x'*x
```

```
ans =
    20
```

```
x =
     2
     4
```

20

## Vecteurs et matrices (9)

### Manipulation de matrices

C =

```

1     2     3     4
5     6     7     8
9     10    11    12
13    14    15    16
```

EDU>> C(:,2)

ans =

```

2
6
10
14
```

EDU>> C(:,2)=[ ]

C =

```

1     3     4
5     7     8
9     11    12
13    15    16
```

EDU>> C(2,:)=[ ]

C =

```

1     2     3     4
9     10    11    12
13    14    15    16
```

EDU>> C(2:3,:)

ans =

```

5     6     7     8
9     10    11    12
```

EDU>> C(2:3,:)=[50 60 70 80  
; 90 100 110 120]

C =

```

1     2     3     4
50    60    70    80
90    100   110   120
13    14    15    16
```

21

## Vecteurs et matrices (10)

### Manipulation de matrices (suite)

C =

```

1     2     3     4
5     6     7     8
9     10    11    12
13    14    15    16
```

### Suppression d'une colonne

EDU>> C(:,2)=[ ]

C =

```

1     3     4
50    70    80
90    110   120
13    15    16
```

### Suppression d'une ligne

EDU>> C(2,:)=[ ]

C =

```

1     2     3     4
9     10    11    12
13    14    15    16
```

EDU>> C2=C(2,:)

C2 =

```

5     6     7     8
```

EDU>> C2(C2>=6)

ans =

```

6     7     8
```

EDU>> length(C2(C2>=6))

ans =

```

3
```

22

## Vecteurs et matrices (11)

### Manipulation de matrices (suite)

```
C =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
```

```
EDU>> [i,j]=find(C>11)
```

```
i =
```

```
4
```

```
4
```

```
4
```

```
3
```

```
4
```

```
j =
```

```
1
```

```
2
```

```
3
```

```
4
```

```
4
```

### Suppression de valeurs manquantes (NaN=Not a Number)

```
EDU>>vec_C=[1 2 3 NaN 5 6]
```

```
EDU>>vec_C(~isnan(vec_C))
```

```
ans =
```

```
1     2     3     5     6
```

### Fonctions utiles, spécialisées, pour ignorer les valeurs qui manquent:

```
isfinite, isinf, nanmax,  
nanmin, nansum, nanmean,...
```

23

## Vecteurs et matrices (12)

### Calcul du déterminant

```
EDU>> D=rand(4,4)
```

```
D =
```

```
0.2785    0.1576    0.8003    0.7922
```

```
0.5469    0.9706    0.1419    0.9595
```

```
0.9575    0.9572    0.4218    0.6557
```

```
0.9649    0.4854    0.9157    0.0357
```

```
EDU>> det(D)
```

```
ans =
```

```
0.1086
```

### Calcul de l'inverse d'une matrice

```
EDU>> D_inv=inv(D)
```

```
D_inv =
```

```
1.0746   -6.1455    7.8911   -3.6214
```

```
-1.9856    6.3378   -7.0938    4.0204
```

```
-0.1351    3.2073   -4.6861    2.8714
```

```
1.4160   -2.3405    3.3711   -2.4275
```

24

## Vecteurs et matrices (13)

---

### Vérification

```
EDU>> D*D_inv
```

```
ans =
    1.0000    0.0000    0.0000    0.0000
    0.0000    1.0000    0.0000    0.0000
    0.0000    0.0000    1.0000    0.0000
    0.0000    0.0000    0.0000    1.0000
```

### Valeurs propres de la matrice

```
EDU>> e=eig(D)
```

```
e =
    2.4921
    0.2113
   -0.7039
   -0.2929
```

### Coefficients du polynôme caractéristique

```
EDU>> poly(B)
```

```
ans =
    1.0000   -2.0000   -1.0000
```

25

## Vectorisation

---

- La plupart des fonctions qui opèrent sur les vecteurs ou sur les matrices retournent un objet de même grandeur. Prenons l'exemple de la racine carrée:

```
EDU>> x=[9 15 16 19 21 25]
```

```
x =
    9    15    16    19    21    25
```

```
EDU>> sx=sqrt(x)
```

```
sx =
    3.0000    3.8730    4.0000    4.3589    4.5826    5.0000
```

26

## Vectorisation (2)

---

- Certaines fonctions (`mean`, `max`, `sum`, `median`, `prod`, `sort`, `std`, ...) opèrent automatiquement sur chaque colonne:

```
EDU>> xmat=[1 3 5; 2 5 4; 5 8 7; 2 4 5]
```

```
xmat =
     1     3     5
     2     5     4
     5     8     7
     2     4     5
```

```
EDU>> mean(xmat)
```

```
ans =
     2.5000     5.0000     5.2500
```

```
EDU>> mean(xmat(:)) %sur tous les éléments de la matrice
```

```
ans =
     4.2500
```

27

## Résumé: opérateurs

---

OP	Fonction
+	Addition
-	Soustraction
*	Multiplication
.*	Multiplication élément par élément
/	Division
./	Division élément par élément
\	Division à gauche
.\	Division à gauche élément par élément
^	Matrice élevée à la puissance
.^	Élément élevé à la puissance
'	Transposition de matrice

28

## Exercices

- Trouvez la matrice inverse ( $\mathbf{A\_inv}$ ) de  $\mathbf{A} = \begin{pmatrix} 2 & 3 \\ -1 & 4 \end{pmatrix}$   
tel que  $\mathbf{A} \cdot \mathbf{A\_inv} = \mathbf{I}$
- Trouvez  $\mathbf{A}^n$  avec  $\mathbf{A} = \begin{pmatrix} 0.99 & 0.01 \\ 0.02 & 0.98 \end{pmatrix}$ ,  $n=2, 4, 8$
- Trouvez  $\mathbf{A}^n$  avec  $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ ,  $n=2, 4, 8, 10$

29

## Tableaux de cellules (Cell Arrays)

- Avec Matlab 5: les éléments des tableaux de cellules peuvent être des éléments de différents types. Ils sont indexés comme des matrices.

Exemple:

```
EDU>> xc={vec ['one ' ' two'] [1 0;0 1]}
xc =
    [1x4 double]    'one two '    [2x2 double]
```

```
EDU>> xc{3}
ans =
     1     0
     0     1
```

```
EDU>> xc([1 3])
ans =
    [1x4 double]    [2x2 double]
```

30

## Tableaux de cellules (2)

---

```
EDU>> vec
vec =
     1     3     5     7

EDU>> xc={vec(1) vec(2)}; ['one ' 'two ']; [1 0;0 1]}
xc =
 [1x2 double]
 'one two '
 [2x2 double]
```

Création d'une matrice à partir d'un tableau de cellules

```
EDU>> cell2mat(xc([1 3]))
ans =
     1     3
     1     0
     0     1
```

31

## Chaînes de caractères (strings)

---

```
EDU>> string=['deux'; 'cinq']
string =
deux
cinq
```

Transposition

```
EDU>> string'
ans =
dc
ei
un
xq
```

32



## Chaînes de caractères (2)

---

Fonctions sur les chaînes de caractères

- `strcmp`, `strcmpi`, `strncmp` - compare deux chaînes de caractères
- `strcat` - concatène deux chaînes de caractères sur une ligne
- `strvcat` - concatène deux chaînes de caractères en colonne
- `sprintf` - texte formaté -> chaîne de caractère
- `blanks` - crée une chaîne de caractères remplie de blancs
- `upper`, `lower` - convertit les chaînes de caractères en majuscules/minuscules
- `deblank`, `strtrim` - enlève les blancs des chaînes de caractères
- `regexp`, `regexp`, `regexprep` - traitement d'expressions régulières

33

## Structures

---

Les structure Matlab permettent la mise en mémoire de données différentes dans un seul objet, et d'adresser les éléments par des noms, par exemple:

```
EDU>> firms(1).name='Nestle Nespresso'
firms =
    name: 'Nestle Nespresso'
EDU>> firms(1).profits=[8000 10000 11500]
EDU>> firms(2).name='Migros Delizioso';
EDU>> firms(2).profits=[4000 6000 7000]
```

Ajout d'une nouvelle entreprise:

```
EDU>> firms(end + 1) = struct('name','Coop
Expresso','profits',...
[9000 9500 9750]);
```

34

## Structures (2)

---

- Il est possible de faire des extractions depuis la structure et de créer des vecteurs, matrices ou des tableaux de cellules:

```
EDU>> thenames = {firms.name}
thenames =
'Nestle Nespresso'      'Migros Delizioso'      'Coop Espresso'

EDU>> theprofits=[firms.profits]
theprofits =
Columns 1 through 7
8000 10000 11500 4000 6000 7000 9000
Columns 8 through 9
9500 9750

EDU>> size(theprofits)
ans =
     1     9
```

35

## Structures (3)

---

- Calcul de la moyenne des profits:

```
EDU>> cellfun(@mean,{firms.profits})

ans =
     1.0e+03 *
     9.8333     5.6667     9.4167
```

36

## Tri

---

- Premier exemple simple

```
EDU>> x=[7 5 6 9 3 4 5 2]
x =
     7     5     6     9     3     4     5     2

EDU>> sort(x)
ans =
     2     3     4     5     5     6     7     9

EDU>> sort(x, 'descend')
ans =
     9     7     6     5     5     4     3     2
```

37

## Tri (2)

---

- Deuxième exemple avec des matrices

```
EDU>> x=[12 17 15; 3 6 4; 2 7 9; 11 10 9]
x =
    12    17    15
     3     6     4
     2     7     9
    11    10     9

EDU>> sort(x)
ans =
     2     6     4
     3     7     9
    11    10     9
    12    17    15
```

38

## Tabulation

- La fonction «`tabulate`» permet de lister les fréquences des différents chiffres apparaissant dans un vecteur. Le résultat est une matrice 3 x 3:

```
EDU>> x=[8 2 3 8 5 5 6 6 6 7 8 3 4 6]
```

```
x =
```

```
8 2 3 8 5 5 6 6 6 7 8 3 4 6
```

```
EDU>> tabulate(x)
```

Value	Count	Percent
1	0	0.00%
2	1	7.14%
3	2	14.29%
4	1	7.14%
5	2	14.29%
6	4	28.57%
7	1	7.14%
8	3	21.43%

39

## Tabulation (2)

- Il y a des valeurs qui n'apparaissent pas («1» ici). Elles peuvent être éliminées ainsi:

```
EDU>> res(res(:,2)>0,:)
```

```
ans =
```

2.0000	1.0000	7.1429
3.0000	2.0000	14.2857
4.0000	1.0000	7.1429
5.0000	2.0000	14.2857
6.0000	4.0000	28.5714
7.0000	1.0000	7.1429
8.0000	3.0000	21.4286

40

## Lecture de données externes

---

fichier data1.m:

```
6 2 Stephan 65
20 1 Rebecca 98
15 12 Oliver 94
29 8 Steve 04
```

```
EDU>> f=fopen('data1.m','r')
EDU>> x=textscan(f,'%n %n %s %n');
EDU>> x
x =
[4x1 double]    [4x1 double]    {4x1 cell}    [4x1 double]

EDU>> xmat=[x{1} x{2} x{4}]
xmat =
     6     2    65
    20     1    98
    15    12    94
    29     8     4
```

41

## Lecture de données utilisateur

---

- Entrée de données

```
EDU>> x=input('Entrez un vecteur de nombres entiers: ')
Entrez un vecteur de nombres entiers: [1 2 4 6 8]
x =
     1     2     4     6     8
```

- Entrée d'un nom de fichier

```
EDU>> fname=input('Nom du fichier des données: ', 's')
Nom du fichier des données: data1.m
fname =
data1.m

EDU>> f=fopen(fname,'r')
EDU>> x = textscan(f,'%n %n %s %n')
x =
[4x1 double]    [4x1 double]    {4x1 cell}    [4x1 double]
```

42

## Graphiques

---

Fonctions graphiques les plus utilisées:

- `plot`, `plot3`: graphiques 2D et 3D à partir de données
- `fplot`: graphique 2D
- `loglog`, `semilogx`, `semilogy`: graphiques 2D avec des échelles log
- `hist`, `bar`, `barh`, `rose`: graphiques 2D, histogrammes
- `mesh`, `meshz`, `meshc`, `waterfall`: graphiques 3D, surfaces à partir de données
- `surf`, `surfc`, `surfl`: graphiques 3D

Les graphiques créés par Matlab peuvent être modifiés quand ils sont créés ou interactivement.

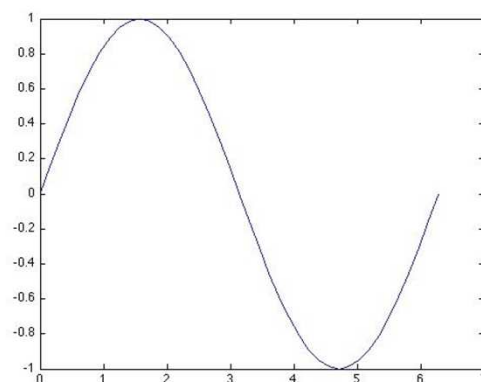
43

## Graphiques 2D

---

Le plus simple des graphiques:

```
EDU>> x=0:pi/20:2*pi
EDU>> y=sin(x)
EDU>> plot(x,y)
```



44

## Couleurs, styles

Les graphiques peuvent être modifiés à l'aide des symboles suivants (tiré de Phil Spector, *An Introduction to Matlab*):

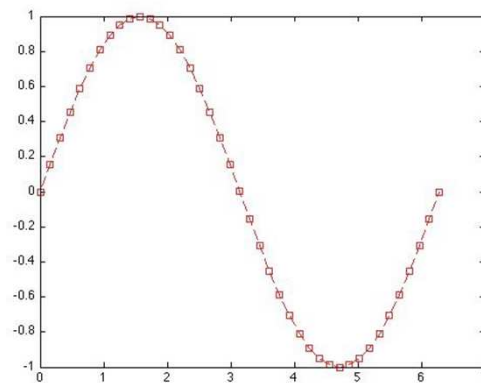
Symbol	Color	Symbol	Marker	Symbol	Linestyle
b	blue	.	point	-	solid line
g	green	o	circle	:	dotted line
r	red	x	cross	-.	dash-dot line
c	cyan	+	plus sign	--	dashed line
m	magenta	*	asterisk		
y	yellow	s	square		
k	black	v	triangle (down)		
w	white	^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

45

## Couleurs, styles (2)

Exemple précédent:

```
EDU>> x=0:pi/20:2*pi
EDU>> y=sin(x)
EDU>> plot(x,y,'-rs')
```



46

## Informations supplémentaires sur les graphiques

Les fonctions-commandes suivantes sont disponibles:

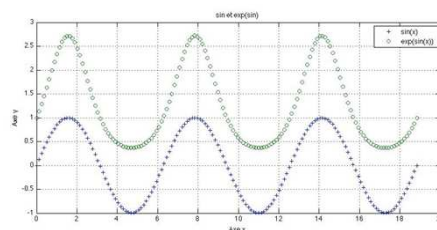
- `xlabel`, `ylabel` - ajoute des labels sur les axes
- `axis` - permet de changer les propriétés des axes
- `grid` (`grid on` ou `grid off`) - permet d'ajouter une grille sur le graphique
- `box` (`box on` ou `box off`)
- `title` - permet d'ajouter un titre au graphe
- `text` - permet d'ajouter du texte sur le graphique
- `gtext` - permet d'ajouter du texte sur le graphique avec un «pointeur»
- `annotation` - permet d'ajouter des objets sur le graphique (flèches, rectangles, ellipses,...)
- `legend` - ajoute une légende sur un graphique

47

## Graphiques

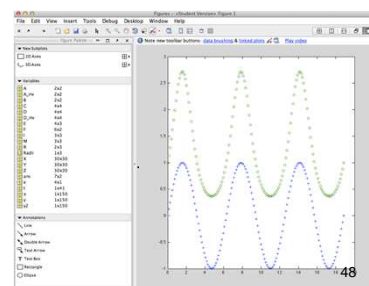
Un petit peu plus élaboré

```
EDU>> x=linspace(0,6*pi,150)
EDU>> y=sin(x)
EDU>> y2=exp(sin(x))
EDU>> plot(x,y,'+',x,y2,'o')
EDU>> title('sin et exp(sin)')
EDU>> xlabel('Axe x')
EDU>> ylabel('Axe y')
EDU>> grid
EDU>> legend('sin(x)', 'exp(sin(x))')
```



Remarques:

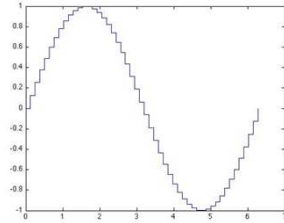
1. Un sous-ensemble du langage LaTeX est disponible sous Matlab (ex: `\alpha`, `\sin`,...).
2. Il est possible de sauver la figure en format Adobe Illustrator (\*.ai) et de la retravailler avec cet outil.
3. Il est possible d'éditer et d'insérer du texte-labels à la figure à l'aide de la palette ci contre:



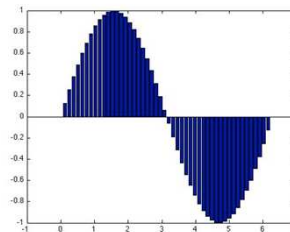


## Graphiques: variations

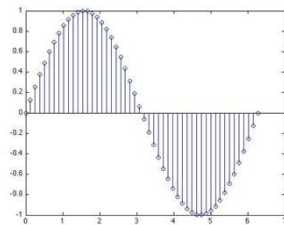
EDU>> `stairs(x,y)`



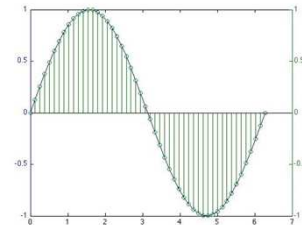
EDU>> `bar(x,y)`



EDU>> `stem(x,y)`



EDU>> `ploty(x,y,x,y, plot , stem )`

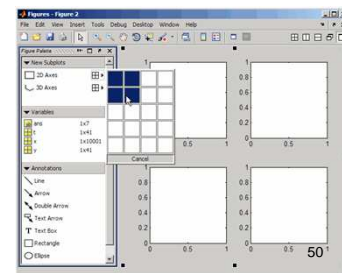
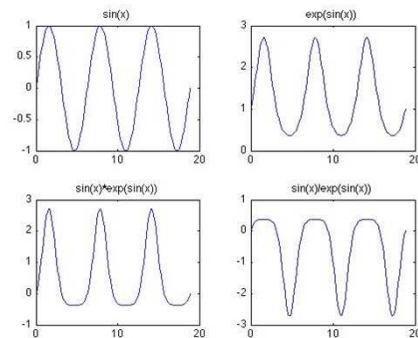


49

## Graphiques: subplots

```
EDU>> x=linspace(0,6*pi,150)
EDU>> y=sin(x)
EDU>> y2=exp(sin(x))
EDU>> a=y.*y2
EDU>> b=y./y2
EDU>> subplot(2,2,1)
EDU>> plot(x,y),title('sin(x)')
EDU>> subplot(2,2,2)
EDU>> plot(x,y2),title('exp(sin(x))')
EDU>> subplot(2,2,3)
EDU>> plot(x,a),title('sin(x)*exp(sin(x))')
EDU>> subplot(2,2,4)
EDU>> plot(x,b),title('sin(x)/exp(sin(x))')
```

Autre option: création de graphes avec la palette ->



## Langage LaTeX reconnu par Matlab

Character Sequence	Symbol	Character Sequence	Symbol	Character Sequence	Symbol
\alpha	$\alpha$	\varepsilon	$\varepsilon$	\aleph	$\aleph$
\angie	$\sphericalangle$	\phi	$\phi$	\leq	$\leq$
\ast	$*$	\chi	$\chi$	\infty	$\infty$
\beta	$\beta$	\psi	$\psi$	\subseteq	$\subseteq$
\gamma	$\gamma$	\omega	$\omega$	\diamondsuit	$\blacklozenge$
\delta	$\delta$	\Gamma	$\Gamma$	\heartsuit	$\blackheartsuit$
\epsilon	$\epsilon$	\Delta	$\Delta$	\spadesuit	$\blackspadesuit$
\zeta	$\zeta$	\Upsilon	$\Upsilon$	\leqrightarrow	$\rightarrowtail$
\eta	$\eta$	\Lambda	$\Lambda$	\leftarrow	$\leftarrow$
\theta	$\theta$	\Xi	$\Xi$	\leftarrowtail	$\leftarrowtail$
\vartheta	$\vartheta$	\Pi	$\Pi$	\uparrow	$\uparrow$
\iota	$\iota$	\Sigma	$\Sigma$	\rightarrow	$\rightarrow$
\kappa	$\kappa$	\Upsilon	$\Upsilon$	\rightarrowtail	$\rightarrowtail$
\lambda	$\lambda$	\Phi	$\Phi$	\downarrow	$\downarrow$
\mu	$\mu$	\Psi	$\Psi$	\circ	$\circ$
\nu	$\nu$	\Omega	$\Omega$	\pm	$\pm$
\xi	$\xi$	\forall	$\forall$	\geq	$\geq$
\pi	$\pi$	\exists	$\exists$	\propto	$\propto$
\rho	$\rho$	\partial	$\partial$	\partial	$\partial$
\sigma	$\sigma$	\cong	$\cong$	\bullet	$\bullet$

51

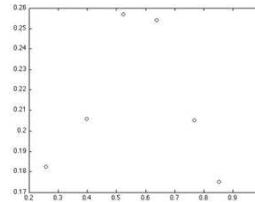
## Langage LaTeX reconnu par Matlab (2)

\varsigma	$\varsigma$	\approx	$\approx$	\div	$\div$
\tau	$\tau$	\simeq	$\simeq$	\neq	$\neq$
\equiv	$\equiv$	\oplus	$\oplus$	\aleph	$\aleph$
\Im	$\Im$	\cup	$\cup$	\wp	$\wp$
\otimes	$\otimes$	\subseteq	$\subseteq$	\lesseqgtr	$\lesseqgtr$
\cap	$\cap$	\in	$\in$	\asymp	$\asymp$
\supseteq	$\supseteq$	\llcorner	$\llcorner$	\subseteq	$\subseteq$
\int	$\int$	\cdot	$\cdot$	\lo	$\lo$
\rfloor	$\rfloor$	\odot	$\odot$	\nabla	$\nabla$
\lfloor	$\lfloor$	\times	$\times$	\dots	$\dots$
\perp	$\perp$	\surd	$\surd$	\prime	$\prime$
\wedge	$\wedge$	\wp	$\wp$	\O	$\O$
\lrcorner	$\lrcorner$	\rangle	$\rangle$	\mid	$\mid$
\vee	$\vee$			\copyright	$\copyright$
\langle	$\langle$				

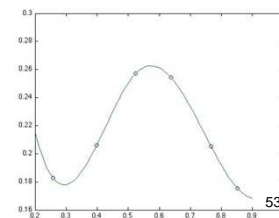
52

## Graphiques: saisie et spline

```
EDU>> G=ginput
  Saisir les points et faire <return>
  quand c'est terminé
EDU>> x=G(:,1) '
EDU>> y=G(:,2) '
EDU>> plot(x,y,'o',)
```



```
EDU>> cs = spline(x,y);
EDU>> xx = linspace(0.2,0.9,101);
EDU>> plot(x,y,'o',xx,ppval(cs,xx),'-');
```



## Graphiques 2D

Autres fonctions utiles:

- `hold on/hold off` (ajout d'un graphe sur une figure)
- `axis([XMIN XMAX YMIN YMAX])`
- `loglog`
- `semilogx`
- `semilogy`
- `polar(t,r)`: t en radians, r: rayon
- `comet(x,y)` (approche dynamique de l'affichage)

## Exercices

- Représenter les deux fonctions en utilisant deux échelles différentes pour chacune d'elles:

$$y_1(t) = 200e^{-0.05t} \sin(t)$$

$$y_2(t) = 0.8 \sin(10t)e^{-0.5t}$$

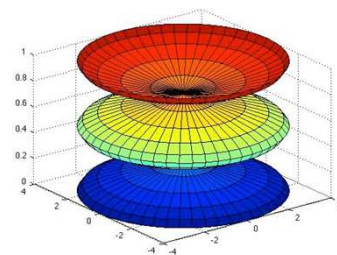
- Représenter  $y_1 = x$  et  $y_2 = \sqrt{1+x}$  sur le même graphique et indiquez le point fixe avec une flèche (écrire «point fixe»)
- Représenter les points fixes de  $x = \cos(x)$  et de  $x = \tan(x)$ . Mettre les labels aux axes, une légende, un titre et indiquez le point fixe.

55

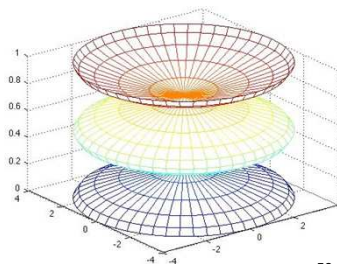
## Graphiques 3D

Fonctions utilisées: surf, mesh, contour

```
EDU>> t = 0:pi/10:2*pi;
EDU>> [X,Y,Z] = cylinder(4*cos(t),50);
EDU>> surf(X,Y,Z)
```



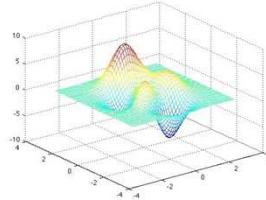
```
EDU>> mesh(X,Y,Z)
```



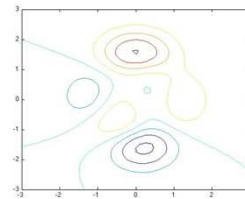
56

## Graphiques 3D

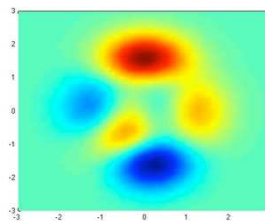
```
EDU>> [X,Y,Z] = peaks
EDU>> mesh(X,Y,Z)
```



```
EDU>> contour(X,Y,Z)
```



```
EDU>> pcolor(x,y,z)
EDU>> shading interp
```

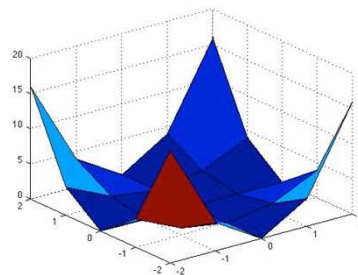


57

## Graphiques 3D (2)

Exemple simple avec **surf**

```
EDU>> X=[-2 -1 0 1 2]
EDU>> Y=X
EDU>> Z(:,1)=(X.^2)*(-2)^2
EDU>> Z(:,2)=(X.^2)*(-1)^2
EDU>> Z(:,3)=(X.^2)*(0)^2
EDU>> Z(:,4)=(X.^2)*(1)^2
EDU>> Z(:,5)=(X.^2)*(2)^2
EDU>> surf(X,Y,Z)
```



Remarques:

- Spécification des limites des axes: `axis([xmin xmax, ymin ymax, zmin zmax])`
- Mode automatique (limites choisies automatiquement): `axis auto`
- Point de vue de l'utilisateur: `view(AZ,EL)`, AZ: rotation horizontale en degrés, EL: élévation en degrés.  
Valeurs par défaut: -37,30.
- Nouvelles versions de Matlab: possibilité de faire des animations avec `VideoWriter`.

58

## Sauver/imprimer son graphique

---

Pour sauver son graphique:

```
print -dpng plot.png
print -dpng plot
```

Autres formats: `-dps`, `-dtiff`, `-dpsc` (ps couleur), `-depsc` (encapsulated ps)

Pour imprimer son graphique sur l'imprimante par défaut:

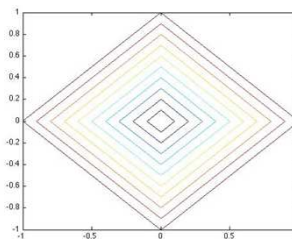
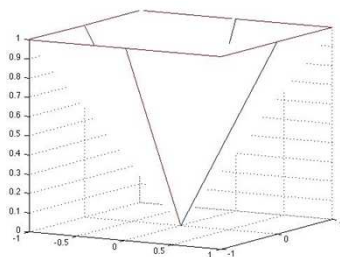
```
print
```

59

## Exercice

---

⇒ Dessinez la surface (3D, mesh, contour) qui relie les points suivants:  $(-1,-1,1)$ ,  $(-1,0,1)$ ,  $(-1,1,1)$ ,  $(0,-1,1)$ ,  $(0,1,1)$ ,  $(1,-1,1)$ ,  $(1,0,1)$ ,  $(1,1,1)$  et  $(0,0,0)$ . Changez le point de vue (30,10).



60

## Calcul symbolique (notions)

Premier exemple:

```
EDU>> diff('cos(x)')
ans =
-sin(x)
```

Deuxième exemple:

```
EDU>> M=sym('[a,b;c,d]')
M =
[ a, b]
[ c, d]
EDU>> det(M)
ans =
a*d - b*c
```

Troisième exemple:

```
EDU>> int('cos(x)')
ans =
sin(x)
```

Quatrième exemple:

```
EDU>> f='x^2+x'
f =
x^2+x
EDU>> diff(f)
ans =
2*x + 1
```

Cinquième exemple:

```
EDU>> finverse('x^2',x)
Warning: finverse(x^2) is not
unique.
> In sym.finverse at 46
ans =
x^(1/2)
```

```
Matlab R2014b:
>>syms x t
>>diff(sin(t))
```

61

## Calcul symbolique (2)

Sixième exemple:

```
EDU>> f='cos(a*x) '
f =
cos(a*x)
EDU>> diff(f,'x',2)
ans =
-a^2*cos(a*x)
```

Septième exemple:

```
EDU>> f='cos(a*x) '
f =
cos(a*x)
EDU>> int(f)
ans =
sin(a*x)/a
EDU>> int(f,'a')
ans =
sin(a*x)/x
```

Huitième exemple:

```
EDU>>
pretty(diff('sqrt(3*x^2+2*x+5)'))
          6 x + 2
-----
          2          1/2
2 (3 x  + 2 x + 5)
```

```
EDU>>
pretty(simplify(diff('sqrt(3*x^2+2*
x+5)'))))
          3 x + 1
-----
          2          1/2
(3 x  + 2 x + 5)
```

62

## Calcul symbolique (3)

---

Neuvième exemple:

```
EDU>> f=taylor(sym('sin(x)'))
f =
x^5/120 - x^3/6 + x
```

Dixième exemple (Variable Precision Arithmetics):

```
EDU>> vpa('pi',50)
ans =
3.1415926535897932384626433832795028841971693993751
```

Onzième exemple:

```
EDU>> f=taylor(sym('sin(x)'))
EDU>> ezplot(f,[0 5])
```

63

## Calcul symbolique (4)

---

### Déclaration de variables

Exemple:

```
syms a b c x
f(x,y)=a*x^2+b*x+c
```

### Déclaration de variables avec contraintes

Exemple:

```
syms x beta real
est équivalent à:
x = sym('x','real');
beta = sym('beta','real');
```

64



## Calcul symbolique: calcul différentiel

### Première dérivée

Exemple:

```
EDU>> syms x fx a b
EDU>> fx=1/(a+b*cos(x))
EDU>> fDiff=diff(fx,'x',1)
fDiff =
(b*sin(x))/(a + b*cos(x))^2
EDU>> simplify(fDiff)
ans =
(b*sin(x))/(a + b*cos(x))^2
EDU>> latex(ans)
```

$$\frac{b \sin(x)}{(a + b \cos(x))^2}$$

65

## Calcul symbolique: calcul différentiel (2)

### Deuxième dérivée

Exemple:

```
EDU>> syms x fx a b
EDU>> fx=1/(a+b*cos(x))
EDU>> fDiff2=diff(fx,'x',2)
fDiff2 =
(2*b^2*sin(x)^2)/(a + b*cos(x))^3 + (b*cos(x))/(a +
b*cos(x))^2
EDU>> simplify(fDiff2)
ans =
-(b^2*(cos(x)^2 - 2) - a*b*cos(x))/(a + b*cos(x))^3
EDU>> latex(ans)
```

$$-\frac{b^2 (\cos(x)^2 - 2) - a b \cos(x)}{(a + b \cos(x))^3}$$

66

## Calcul symbolique: calcul différentiel (3)

### Fonction à 2 variables

Exemple:

```
EDU>> fxy=x^2+y^2+x*y
EDU>> fDiffx=diff(fxy,'x',1)
fDiffx =
2*x + y
EDU>> fDiffy=diff(fxy,'y',1)
fDiffy =
x + 2*y
EDU>> fDiffxy=diff(diff(fxy,'x'),'y')
fDiffxy =
1
EDU>> fDiffyx=diff(diff(fxy,'y'),'x')
fDiffyx =
1
```

67

## Calcul symbolique: calcul intégral

### Intégration indéfinie

Exemple:

```
EDU>> fx
fx =
1/(a + b*cos(x))
EDU>> simplify(int(fx,x))
ans =
(2*atan((tan(x/2)*(a^2 - b^2)^(1/2))/(a + b)))/(a^2 -
b^2)^(1/2)
```

$$2 \arctan\left(\frac{\tan\left(\frac{x}{2}\right) \sqrt{a^2 - b^2}}{a + b}\right)$$


---


$$\sqrt{a^2 - b^2}$$


---


$$\sqrt{a^2 - b^2}$$

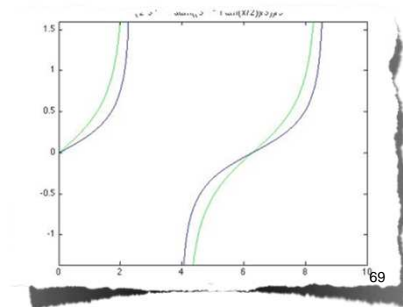
68

## Calcul symbolique: calcul intégral (2)

### Intégration définie

Exemple de l'intégrale définie entre 0 et 1 (a=1, b=2):

```
EDU>> fx
fx =
1/(a + b*cos(x))
EDU>> Res=int(subs(fx,{a,b},{1,2}),x,0,1)
Res =
(2*3^(1/2)*atanh((3^(1/2)*sin(1/2))/(3*cos(1/2))))/3
EDU>> vpa(Res)
ans =
0.3770552654488305643164964
EDU>> h=ezplot(int(subs(fx,{a,b},
{1,2})),[0,10])
EDU>> set(h,'color','g')
EDU>> hold on
EDU>> g=ezplot(int(subs(fx,{a,b},
{2,3})),[0,10])
EDU>> hold off
```



## Intégrale définie simple

```
EDU>> f='x^2'
EDU>> int(f,0,2)
ans =
8/3
EDU>> int(sym(f))
ans =
x^3/3
```

## Calcul symbolique: Séries

### Série géométrique finie

```
EDU>> simple(symsum(a^k,k,0,N-1))
ans =
piecewise([a = 1, N], [a <> 1, (a^N - 1)/(a - 1)])
```

$$\begin{cases} N & \text{if } a = 1 \\ \frac{a^N - 1}{a - 1} & \text{if } a \neq 1 \end{cases}$$

### Série géométrique infinie

```
EDU>> simple(symsum(a^k,k,0,Inf))
ans =
piecewise([1 <= a, Inf], [abs(a) < 1, -1/(a - 1)])
```

$$\begin{cases} \infty & \text{if } 1 \leq a \\ -\frac{1}{a-1} & \text{if } |a| < 1 \end{cases}$$

71

## Calcul symbolique: Séries (2)

### Intercorrélation

```
EDU>> simplify(symsum(a^k*b^(k+1),k,0,N-1))
ans =
piecewise([a*b = 1, N*b^1], [a*b <> 1, -(b^1 -
a^N*b^N*b^1)/(a*b - 1)sin()]
```

$$\begin{cases} N b^1 & \text{if } ab = 1 \\ -\frac{b^1 - a^N b^N b^1}{a b - 1} & \text{if } ab \neq 1 \end{cases}$$

### Transformée de Fourier discrète d'un signal

```
EDU>> simplify(symsum(a^k*exp(-j*2*pi*k*n/N),k,0,N-1))
ans =
piecewise([a*exp(-2/N*i*n*pi) = 1, N], [a*exp(-2/N*i*n*pi)
<> 1, (a^N/exp(2*pi*i*n) - 1)/(a/exp((2*pi*i*n)/N) - 1)])
```

$$\begin{cases} N & \text{if } a e^{-\frac{\pi n 2i}{N}} = 1 \\ \frac{a^N e^{-\frac{\pi n 2i}{N}} - 1}{a e^{-\frac{\pi n 2i}{N}} - 1} & \text{if } a e^{-\frac{\pi n 2i}{N}} \neq 1 \end{cases}$$

72

## Exercices

---

- Trouvez la dérivée par rapport à  $x$  de l'expression suivante  $ax^4 + bx^2 + 3$
- Trouvez l'intégrale non définie de l'expression suivante  $ax^4 + bx^2 + 3$
- Trouvez la dérivée par rapport à  $m$  de l'expression suivante  $ax^m + bx^n + 3$
- Trouvez le résultat de l'intégrale définie suivante:  $\int_0^1 \frac{\sin(x) \cos(x)}{x} dx$
- Calculez la somme suivante:  $\sum_0^{10} 0.5^k$
- Calculez la somme suivante:  $\sum_{k=1}^{\infty} \frac{1}{(2k-1)^2}$
- Calculez l'intercorrélation (4 périodes) entre  $\sin\left(\frac{2\pi f n}{N}\right)$  et  $\sin\left(\frac{2\pi f(n-l)}{N}\right)$  avec  $f=1$  Hz,  $N=100$  et  $l=0, 50, 100, 150$

73

## Fonctions

---

⇒Format:

```
function[output_args] = untitled(input_args)
```

⇒Création d'une fonction sous MatLab:

- File/New/Function M-File

```
function [output_args] = untitled(input_args)
% UNTITLED2 Summary of this function goes here
% Detailed explanation goes here
end
```

74

## Fonction simple

---

### **myfunction.m:**

```
function b = myfunction(a)
    b = a.^2 + a.*2;
end
```

→ Appel de la fonction:

```
EDU>> myfunction(3)
ans =
    15
EDU>> myfunction(pi)
ans =
    16.1528
```

75

## Fonctions

---

→ Exemple 1:

```
function [mean,stdev] = stat(x)
    n = length(x);
    mean = sum(x)/n;
    stdev = sqrt(sum((x-mean).^2)/n);
end
```

→ Appel de la fonction:

```
EDU>> x=rand(100,1)
EDU>> [a,b]=stat(x)
a =
    0.4997
b =
    0.3034
```

76

## Fonctions imbriquées

→ Exemple 2 (deux fichiers séparés):

```
function [mean,stdev] = stat2(x)
    n = length(x);
    mean = avg(x,n);
    stdev = sqrt(sum((x-avg(x,n)).^2)/n);
end
function mean = avg(x,n)
    mean = sum(x)/n;
end
```

→ Appel de la fonction:

```
EDU>> x=rand(100,1)
EDU>> [a,b]=stat2(x)
a =
    0.4997
b =
    0.3034
```

77

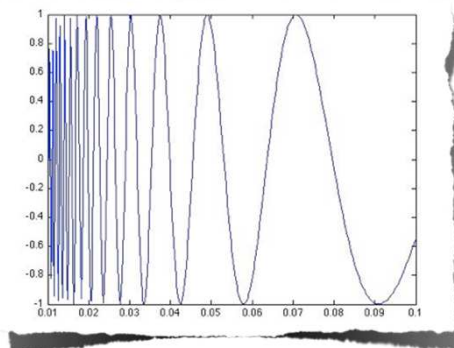
## Fonctions anonymes

Fonctions sans avoir à créer de fichier

Exemple 1:

```
EDU>> s = @(x) sin(1./x);
EDU>> y = s(pi)
y =
    0.3130
```

```
EDU>> range = [0.01,0.1];
EDU>> fplot(s,range)
```



78

## Fonctions anonymes (2)

Exemple 2:

```
EDU>> a = 1.3;
EDU>> b = .2;
EDU>> c = 30;
EDU>> parabola = @(x) a*x.^2 + b*x + c;
EDU>> x=2
x =
    2
EDU>> parabola(x)
ans =
    35.6000
```

Exemple 3 (sans entrée):

```
EDU>> t = @() datestr(now);
EDU>> d = t()
d =
06-Aug-2013 22:03:22
```

79

## Fonctions anonymes (3)

Exemple 4 (multiples entrées):

```
EDU>> myfunction = @(x,y) (x.^2 + y.^2 + x*y);
EDU>> x = 1;
EDU>> y = 10;
EDU>> z = myfunction(x,y)
z =
    111
```

Exemple 5 (tableau de fonctions anonymes):

```
EDU>> f = {@(x) (x.^2);
           @(y) (y + 10);
           @(x,y) (x.^2 + y + 10)};
EDU>> x=1;
EDU>> y=10;
EDU>> f{1}(x)
ans =
    1
EDU>> f{2}(y)
ans =
    20
EDU>> f{3}(x,y)
ans =
    21
```

80



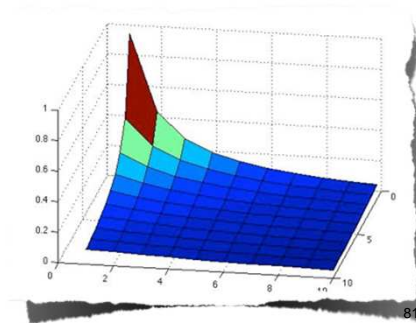
## Boucles «for»

→Format:

```
FOR variable = expr,
    statement,
    ...,
    statement
END
```

→Exemple:

```
for R = 1:N
    for C = 1:N
        A(R,C) = 1/(R+C-1);
    end
end
```



## Boucles «While»

→Format:

```
WHILE expression
    statements
END
```

→Exemple:

```
num=0;
eps=1;
while (1+eps)>1
    eps=eps/2;
    num=num+1;
end
```

→Résultat:

```
EDU>> num
num =
    53
EDU>> eps
eps =
    1.1102e-16
```

82

## Structure «if-then-else»

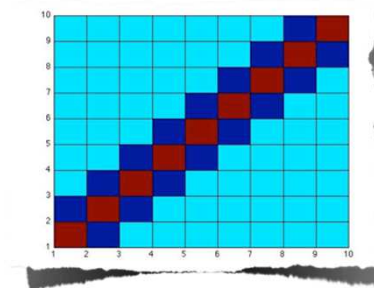
→Format:

```
IF expression
  statements
ELSEIF expression
  statements
ELSE
  statements
END
```

→Résultat:

→Exemple:

```
if i == j
  A(i,j) = 2;
elseif abs(i-j) == 1
  A(i,j) = -1;
else
  A(i,j) = 0;
end
```



→ **Remarque:** else et elseif sont optionnels!

83

## Opérateurs relationnels

Les opérateurs relationnels permettent de tester

1. l'égalité avec ==
2. l'infériorité stricte avec <
3. l'infériorité avec <=
4. la supériorité stricte avec >
5. la supériorité avec >=
6. la non égalité avec ~=

84

## Timing

---

Nous pouvons faire usage de deux fonctions qui nous permettent de mesurer le temps écoulé pour l'exécution du code. Il faut appeler `tic` avant l'exécution et `toc` ensuite, par exemple:

```
EDU>> tic;x1 = 10 * ones(1000000,100);t1=toc;
EDU>> t1
t1 =
    1.0040
```

Il est aussi possible d'utiliser la fonction `cputime` mais sa résolution est moins bonne que celle de `tic/toc`. Pour déterminer quelles sont les parties du programme Matlab qui utilisent le plus le CPU: `profile on/off`.

85

## Debug

---

Pour commencer à «debogguer» un M-file, tapons:

`debug`

dans la console Matlab

Nous pouvons placer des «breakpoints» dans le programme Matlab (M-file) à l'aide de la fonction `dbstop`. Pour savoir où les placer nous ferons usage de la fonction `dbtype`, par exemple:

```
EDU>>dbtype avg
1
2
3           function mean = avg(x,n)
4               mean = sum(x)/n;
5           end
6
EDU>>dbstop avg 4
```

86

## Debug (2)

Maintenant nous allons simplement faire tourner notre programme

```
EDU>> x=[1 3 4 5 6 3]
x =
     1     3     4     5     6     3
EDU>> avg(x,6)
4
mean = sum(x)/n;
K>>
```

Arrivé à ce point, nous avons plusieurs possibilités:

Commande	Fonction
dbstep	Avance d'une ligne dans le prgm
dbcont	Continue l'exécution
dbstop	Ajoute un breakpoint
dbclear	Supprime les breakpoints
dbquit	Quitte le debogguer

87

## Nombres pseudo-aléatoires

«A random sequence is a vague notion... in which each term is unpredictable to the uninitiated and whose digits pass a certain number of tests traditional with statisticians...», D. H. Lehmer, UC Berkeley, inventeur du générateur congruentiel linéaire (algorithme pour générer des nombres pseudo-aléatoires)

Générateur:  $X_{n+1} = (a \cdot X_n + c) \bmod m$   $X_0$ graine, «seed»

Exemple:

$$X_{n+1} = a \cdot X_n - \text{int} \left( \frac{a \cdot X_n}{m} \right) \cdot m$$

avec  $a = 13, c = 0, m = 31, X_0 = 1$

```
X1 = 13.1 mod 31 = 13
X2 = 13.13 mod 31 = 169 - 5.31 = 14
X3 = 13.14 mod 31 = 182 - 5.31 = 27
X4 = 13.27 mod 31 = 351 - 11.31 = 10
X5 = 6
X6 = 16
X7 = 22
...
```

Quelles sont les deux  
prochaines valeurs?  
88

## Nombres pseudo-aléatoires (2)

«A random sequence is a vague notion... in which each term is unpredictable to the uninitiated and whose digits pass a certain number of tests traditional with statisticians...», D. H. Lehmer, UC Berkeley, inventeur du générateur congruentiel linéaire (algorithme pour générer des nombres pseudo-aléatoires)

Générateur:  $X_{n+1} = (a \cdot X_n + c) \bmod m$   $X_0$  : graine, «seed»

Exemple:

avec  $a = 13, c = 0, m = 31, X_0 = 1$

$$\begin{aligned} X_1 &= 13 \cdot 1 \bmod 31 = 13 \\ X_2 &= 13 \cdot 13 \bmod 31 = 169 - 5 \cdot 31 = 14 \\ X_3 &= 13 \cdot 14 \bmod 31 = 182 - 5 \cdot 31 = 27 \\ X_4 &= 13 \cdot 27 \bmod 31 = 351 - 11 \cdot 31 = 10 \\ X_5 &= 6 \\ X_6 &= 16 \\ X_7 &= 22 \\ &\dots \end{aligned}$$

7, 29, 5, 3, 8, ... 89

## Nombres pseudo-aléatoires (3)

Constats:

- les 30 premiers termes sont une permutation d'entiers entre 1 et 30. La période de la séquence se répète avec une période égale à  $m-1$ .
- Si nous voulons des nombres réels uniformément distribués entre 0 et 1, il suffit de diviser la séquence par  $m$ .

Valeurs de l'ancien générateur de Matlab (basées sur un article de Park&Miller, 1988):

$$\begin{aligned} a &= 7^5 = 16807 \\ c &= 0 \\ m &= 2^{31} - 1 = 2147483647 \end{aligned}$$

Nouveau générateur (1995): George Marsaglia. Etat initial: 35 nombres (32 entre 0 et 1, un index qui varie entre 0 et 31, un entier aléatoire et un drapeau)

90

## Génération de nombres aléatoires

---

Génération d'un nombre aléatoire entre 0 et 1

```
EDU>> format long
EDU>> x=rand
x =
    0.141083175787354
```

Génération d'une matrice aléatoire m x n

```
EDU>> x=rand(3,2)
x =
    0.297824309822895    0.238705917932550
    0.722899373878980    0.134032386040728
    0.883218674022024    0.964248334060640
```

91

## Génération de nombres aléatoires (2)

---

Génération d'une matrice aléatoire carrée n x n

```
EDU>> x=rand(3)
x =
    0.447346461850559    0.621290309454844    0.507944251941602
    0.823857736898164    0.758946918267190    0.203164208480559
    0.394646900312169    0.605637449099472    0.886501037801865
```

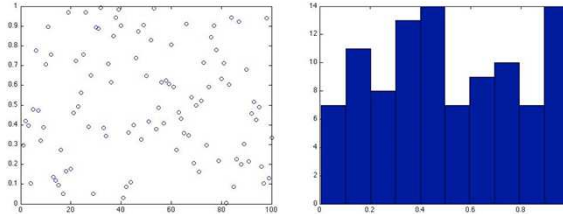
92

## Distribution uniforme

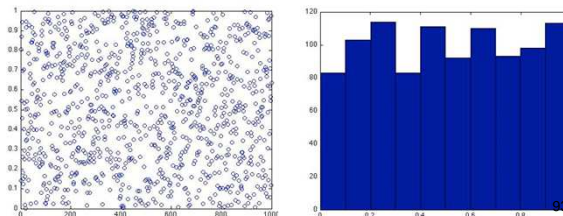
Nous avons vu que nous pouvons obtenir une distribution «pseudo-uniforme» (nombres aléatoires uniformément répartis entre 0 et 1) avec les générateurs précédents.

### Exemples

```
EDU>> x=rand(1,100)
EDU>> plot(x,'o')
EDU>> hist(x)
```



```
EDU>> x=rand(1,1000)
EDU>> plot(x,'o')
EDU>> hist(x)
```

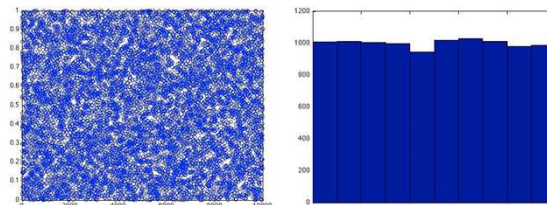


## Distribution uniforme (2)

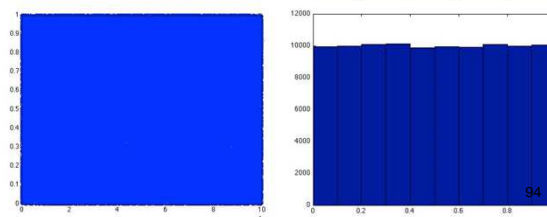
Nous avons vu que nous pouvons obtenir une distribution «pseudo-uniforme» (nombres aléatoires uniformément répartis entre 0 et 1) avec les générateurs précédents.

### Exemples

```
EDU>> x=rand(1,10000)
EDU>> plot(x,'o')
EDU>> hist(x)
```



```
EDU>> x=rand(1,100000)
EDU>> plot(x,'o')
EDU>> hist(x)
```



## A voir...

---

- Démonstrations (Demo dans le menu d'aide)
- Animations (getframe et movies)
- Graphical User Interface
- Autres Toolboxes (Signal Processing, Wavelet, Optimization, Simulink)
- <http://www.mathworks.com/matlabcentral/fileexchange>
- <http://blogs.mathworks.com/loren/>
- GUI
- `urlread('http ://manyeyes.alphaworks.ibm.com/manyeyes/datasets/us-zipcodes-with-city-state-fips-lat/versions/1.txt')`
- <http://blogs.mathworks.com/videos/2009/03/03/read-data-from-the-web-with-urlread/>

95

## Bibliographie

---

- Michel Kocher, Cours MatLab, HEIG-Vd.
- Cleve Moler, Experiments with MatLab, 2011.
- Cleve Moler, Numerical Computing with MatLab, 2004.
- Hanselman & Littlefield, Mastering Matlab 5.
- Phil Spector, An Introduction to Matlab, UC Berkeley.
- Aide, MatLab.

96