

---

# Introduction à Matlab

---

Ecole Centrale de Pékin

Laurent GORNET



---

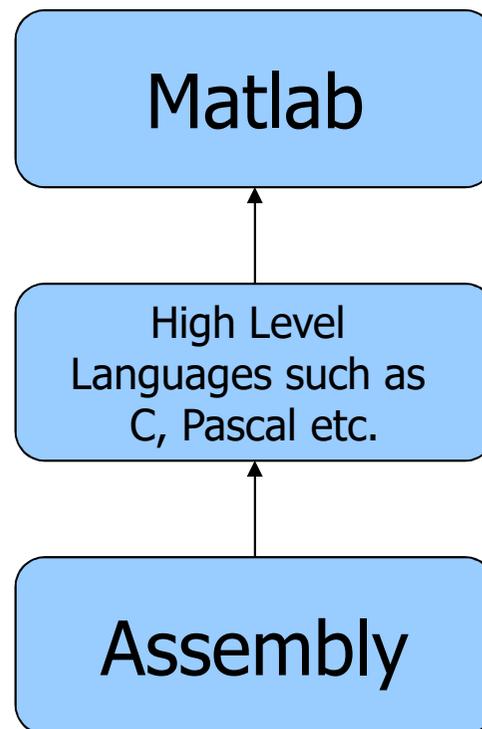
## Résumé :

- Matlab?
  - Ecran Matlab
  - Variable, tableau, matrice, index
  - Opérateur (Arithmetic, relational, logical )
  - Sorties graphiques
  - Instructions de contrôle
  - Utiliser un M-File
  - Ecrire une fonction utilisateur
  - Conclusion
-

---

# Matlab c'est quoi ?

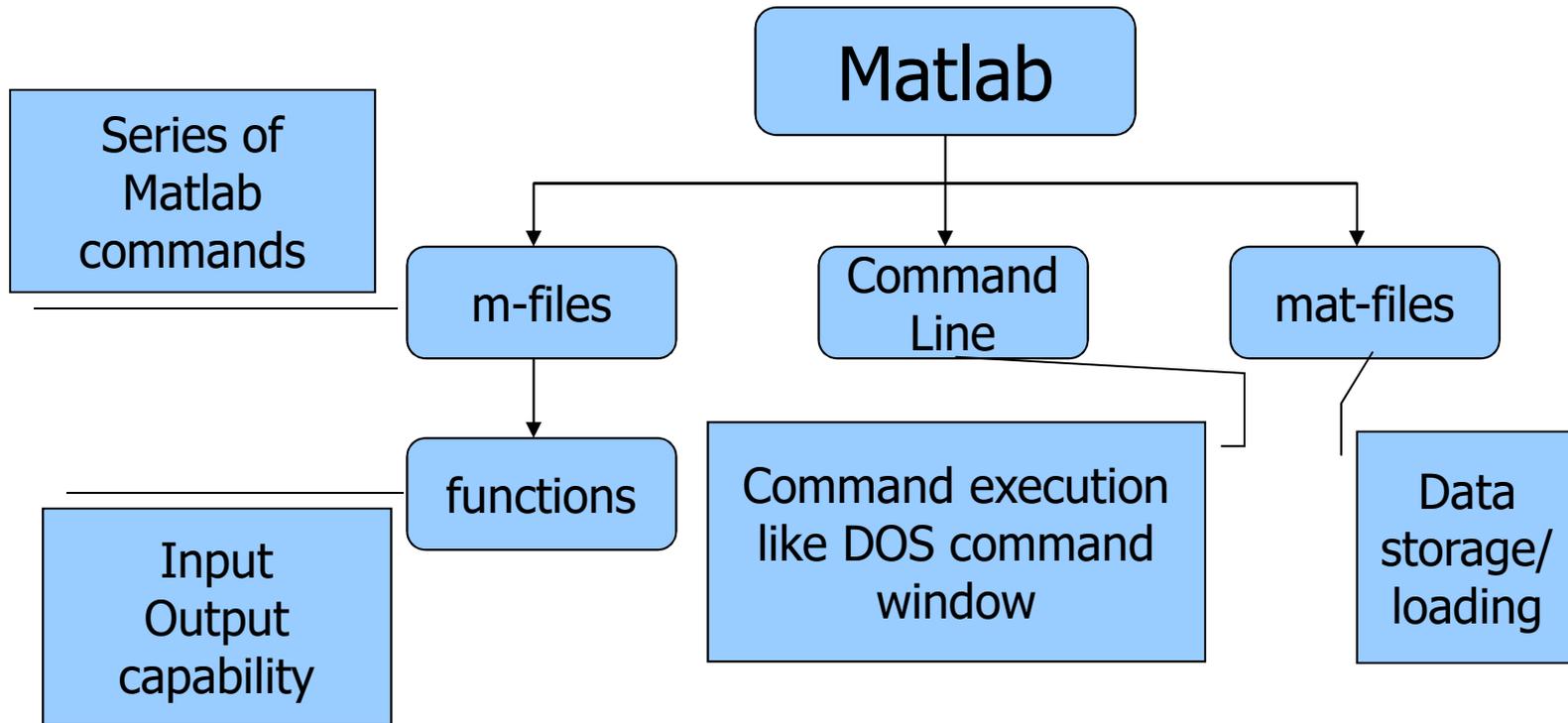
- MATLAB est un langage de haut niveau qui a de nombreuses boîtes à outils spécialisées pour rendre les choses plus facile pour nous.



---

# Matlab, What are we interested in?

- Les caractéristiques étudiées



# Fenêtre Matlab

- Fenêtre de commande

- Commandes matlab

- Curent Directory

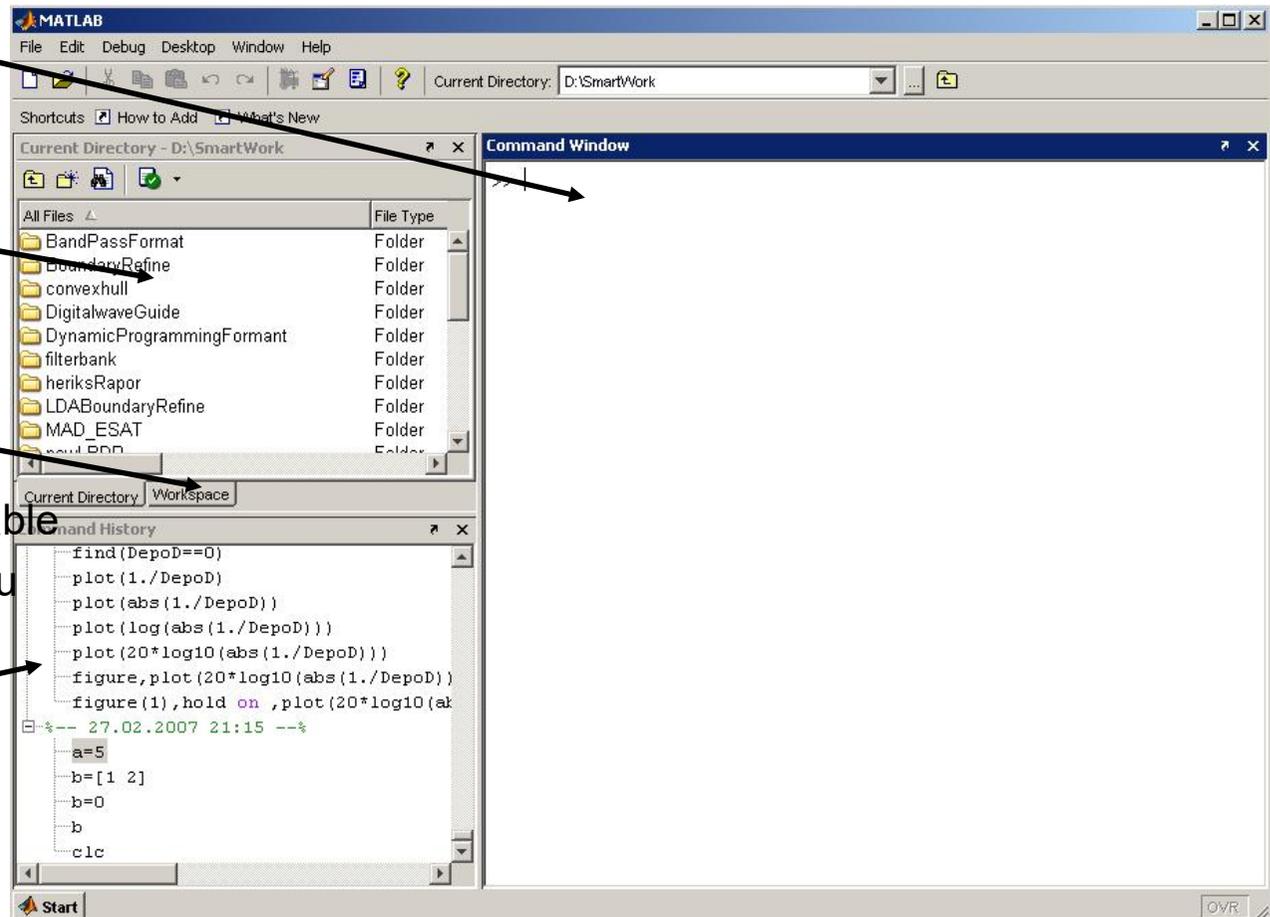
- Directory et m-files

- Workspace

- variables
- Double click sur une variable pour la voir dans le tableau

- Command History

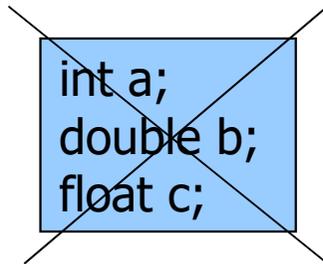
- Commandes pasées
- Sauve la session dans un journal



---

# Variables

- Pas de type pour les variables



```
int a;  
double b;  
float c;
```

- Les variables sont en double precision et sont des matrices.

Example:

```
>>x=5;  
>>x1=2;
```

- Après ces déclarations, les variables sont des matrices de 1 x 1 en double précision
-

# Tableau, Matrices

- un vecteur  $x = [1 \ 2 \ 5 \ 1]$

$$x = \begin{array}{cccc} 1 & 2 & 5 & 1 \end{array}$$

- une matrice  $x = [1 \ 2 \ 3; 5 \ 1 \ 4; 3 \ 2 \ -1]$

$$x = \begin{array}{ccc} 1 & 2 & 3 \\ 5 & 1 & 4 \\ 3 & 2 & -1 \end{array}$$

- La transposée  $y = x'$   $y =$

$$\begin{array}{c} 1 \\ 2 \\ 5 \\ 1 \end{array}$$

# Tableaux, Matrices

■ `t = 1:10`

```
t =  
 1  2  3  4  5  6  7  8  9 10
```

■ `k = 2:-0.5:-1`

```
k =  
 2  1.5  1  0.5  0 -0.5 -1
```

■ `x = [1:4; 5:8]`

```
x =  
 1  2  3  4  
 5  6  7  8
```

# Générer à partir de fonctions

- **zeros**(M,N) MxN matrix of zeros

```
x = zeros(1,3)
```

```
x =
```

```
0      0      0
```

---

- **ones**(M,N) MxN matrix of ones

```
x = ones(1,3)
```

```
x =
```

```
1      1      1
```

---

- **rand**(M,N) MxN matrix of uniformly distributed random numbers on (0,1)

```
x = rand(1,3)
```

```
x =
```

```
0.9501  0.2311  0.6068
```

---

# Index des Matrices

- Les indices de la matrice commencent à partir de 1 (pas de 0 comme en C)
- Les indices de la matrice doivent être des entiers positifs

$A = [3 \ 6 \ 2 ; 5 \ 8 \ 7 ; 3 \ 2 \ 3] ; A = A'$

```
A =  
  
     3     5     3  
     6     8     2  
     2     7     3  
  
>> A(6)  
ans =  
  
     7  
  
>> A(3,2)  
ans =  
  
     7  
  
>> A(2, :)  
ans =  
  
     6     8     2  
  
>> A(1:2,2)  
ans =  
  
     5  
     8
```

A(-2), A(0)

Error: ??? Subscript indices must either be real positive integers or logicals.

A(4,2)

Error: ??? Index exceeds matrix dimensions.

# Concatenation des Matrices 1/2

- $x = [1 \ 2], y = [4 \ 5], z = [0 \ 0]$

$$A = [x \ y]$$

1 2 4 5

$$B = [x ; y]$$

1 2

4 5

$$C = [x \ y ; z]$$

Error:

??? Error using ==> vertcat

All rows in the bracketed expression must have the same number of columns.

---

# Concatenation des Matrices 2/2

$C = [x \ ; \ y \ ; \ z]$

$C =$

1	2
4	5
0	0



---

# Opérateurs arithmétiques

+ addition

- soustraction

\* multiplication

/ division matricielle droite

^ puissance

' conjugué du complexe, transposition

(\ division matricielle gauche)

---

# Operations sur les Matrices

Donner A et B:

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

A =

1	2	3
4	5	6
7	8	9

```
>> B = [3 5 2; 5 2 8; 3 6 9]
```

B =

3	5	2
5	2	8
3	6	9

Addition

```
>> X = A + B
```

X =

4	7	5
9	7	14
10	14	18

Soustraction

```
>> Y = A - B
```

Y =

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

Produit

```
>> Z = A * B
```

Z =

22	27	45
55	66	102
88	105	159

Transposition

```
>> T = A'
```

T =

1	4	7
2	5	8
3	6	9

---

# Opérateurs (élément par élément)

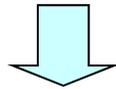
- . \* multiplication élément par élément
  - ./ division élément par élément
  - .^ Puissance élément par élément
-

# Utilisation “.” opérations sur éléments

```
A = [1 2 3; 5 1 4; 3 2 -1]
```

```
A =
```

```
1 2 3
5 1 4
3 2 -1
```



```
x = A(1,:)
```

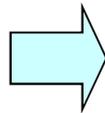
```
y = A(3 ,:)
```

```
x =
```

```
1 2 3
```

```
y =
```

```
3 2 -1
```



```
b = x .* y
```

```
c = x ./ y
```

```
d = x.^2
```

```
b =
```

```
3 4 -3
```

```
c =
```

```
0.33 1 -3
```

```
d =
```

```
1 4 9
```

```
K = x^2
```

```
Error:
```

```
??? Error using ==> mpower Matrix must be square.
```

```
B = x*y
```

```
Error:
```

```
??? Error using ==> mtimes Inner matrix dimensions must agree.
```

---

## Systeme classique à résoudre

- $3x + 2y = 12$
- $x + y = 5$

% instruction matlab

$a = [ 3 \ 2 ; 1 \ 1 ]$

$b = [ 12 ; 5 ]$

% ou !

$b = [ 12 \ 5 ] ; b = b'$

- système

$$a u = b$$

$$u = a^{-1} b$$

- Solution

$$u = a \setminus b$$

$$x = u(1)$$

$$y = u(2)$$

---

---

# La solution est un vecteur (une colonne)

```
a = [ 3 2 ; 1 1 ]  
b = [ 12 ; 5]
```

```
a =
```

```
 3  2  
 1  1
```

```
>> u = a \ b
```

```
u =
```

```
b =
```

```
12  
 5
```

```
 2.0000  
 3.0000
```

**U est une colonne**

---

# Systeme transposé à résoudre

- $3x + 2y = 12$
- $x + y = 5$

$$a u = b$$

$$(a u)^T = b^T$$

$$u^T a^T = b^T$$

- résoudre le système transposé :  $u^T a^T = b^T$

$$u^T a^T (a^T)^{-1} = b^T (a^T)^{-1}$$

$$a = [ 3 \ 2 ; 1 \ 1 ]; a = a'$$

$$b = [ 12 ; 5 ]; b = b'$$

$$u^T = b^T (a^T)^{-1}$$

$$u = b / a$$

---

# La solution est un vecteur transposé (une ligne)

a =

```
3  1  
2  1
```

```
>> b = [ 12 ; 5] ; b= b'
```

b =

```
12  5
```

```
>> u = b / a
```

u =

**U est une ligne**

```
2.0000  3.0000
```

---

# Tracer d'une fonction $\sin(x)$ entre $0 \leq x \leq 4\pi$

- Un tableau x de 100 valeurs entre 0 et  $4\pi$

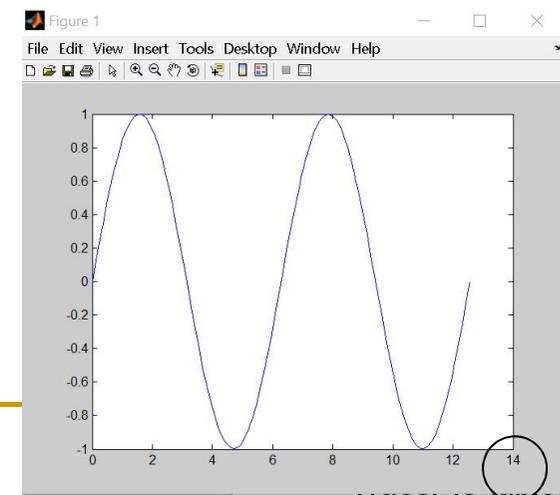
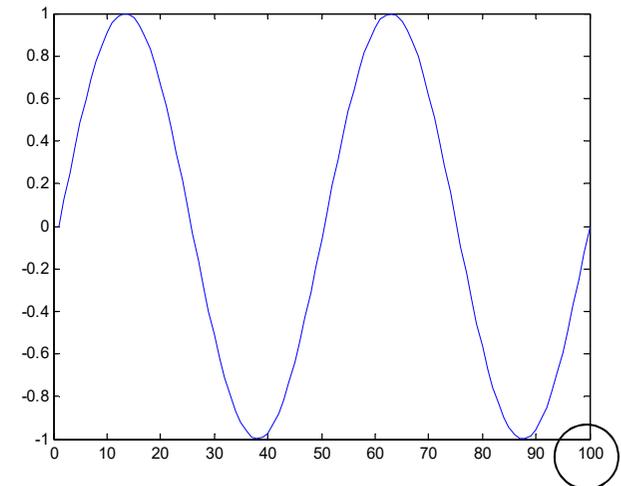
```
x = linspace(0 , 4*pi , 100) ;
```

- Calculer  $\sin(\cdot)$  du vecteur x

```
y = sin(x) ;
```

- Tracer le tableau y et (x,y)

```
plot(y)  
plot(x,y)
```



---

Tracer  $e^{-x/3}\sin(x)$  sur  $0 \leq x \leq 4\pi$

- Un tableau x de 100 valeurs entre 0 et  $0,4\pi$

```
x = linspace(0,4*pi,100) ;
```

- Calculer  $\sin(\cdot)$  du vecteur x

```
y = sin(x) ;
```

- Calculer  $e^{-x/3}$  du vecteur x

```
y1 = exp(-x/3) ;
```

- Multiplie le vecteur y par y1

```
y2 = y * y1 ;
```

??? Error using ==> mtimes  
Inner matrix dimensions must agree.

---

---

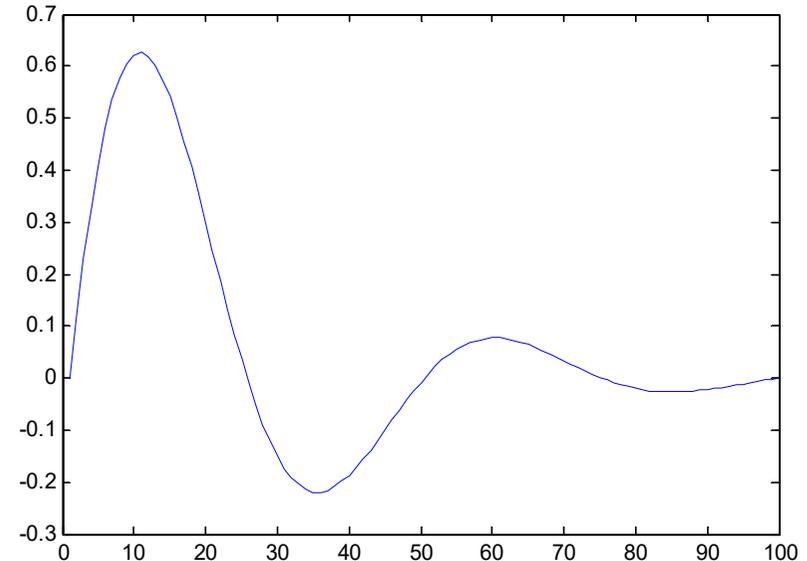
Tracer  $e^{-x/3}\sin(x)$  entre  $0 \leq x \leq 4\pi$

- Multiplie le vecteur y par y1 **correctement**

```
y2=y .* y1 ;
```

- Tracer le tableau y2

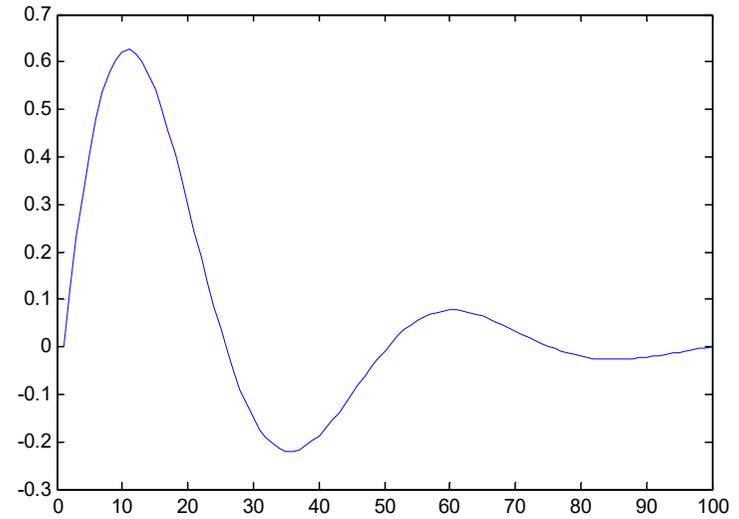
```
plot(y2)
```



# Les options des tracés

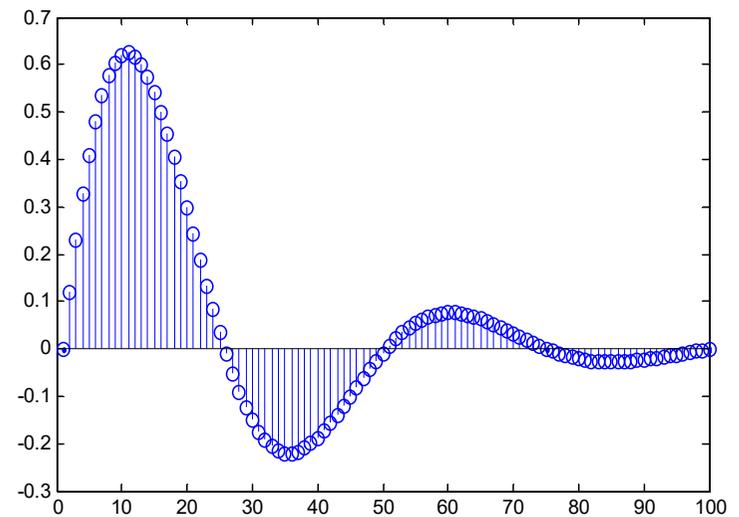
## ■ plot(.)

```
Example:  
X = linspace(0 , 4*pi , 100) ;  
Y = sin(x);  
plot(y)  
plot(x,y)
```



## ■ stem(.)

```
Example:  
stem(y)  
stem(x,y)
```



---

# Les options des tracés

- **title(.)**

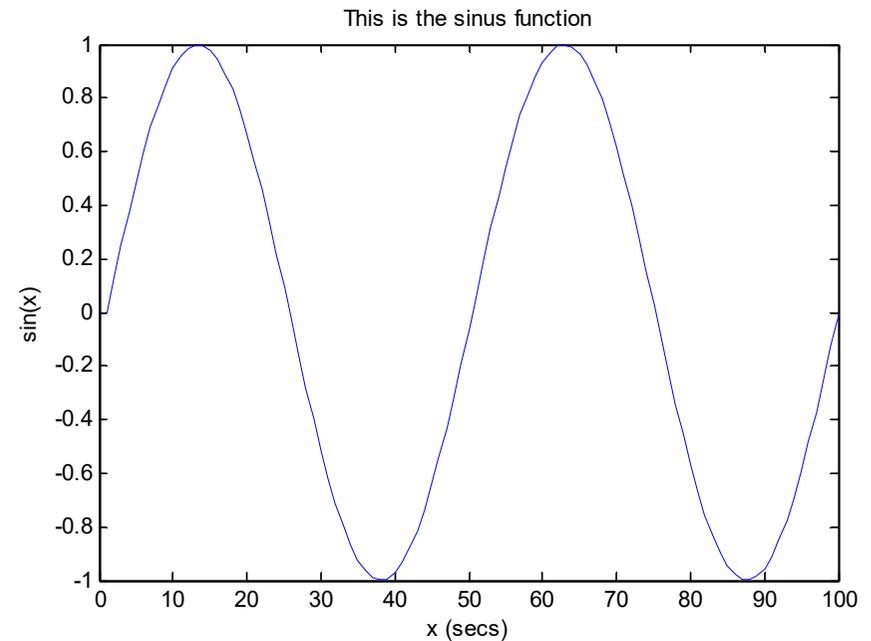
```
title('This is the sinus function')
```

- **xlabel(.)**

```
xlabel('x (secs)')
```

- **ylabel(.)**

```
ylabel('sin(x)')
```



# Opérateurs (relationnelles, logiques)

- == Equal to
- ~= Not equal to
- < Strictly smaller
- > Strictly greater
- <= Smaller than or equal to
- >= Greater than equal to
- & And operator
- | Or operator

---

# Les contrôles, instructions :

- if
  - for
  - while
  - break
  - ....
  - **clear all**
  - **clc**
-

# Structure de contrôle

## ■ If : Syntaxe

```
if (Condition_1)
    Commandes Matlab
elseif (Condition_2)
    Commandes Matlab
elseif (Condition_3)
    Commandes Matlab
else
    Commandes Matlab
end
```

## Some Dummy Examples

```
if ((a>3) & (b==5))
    Some Matlab Commands;
end
```

```
if (a<3)
    Some Matlab Commands;
elseif (b~=5)
    Some Matlab Commands;
end
```

```
if (a<3)
    Some Matlab Commands;
else
    Some Matlab Commands;
end
```

# Structures de Contrôle

## ■ Syntaxe des boucles

```
for i = vecteur indexe
    Matlab Commands
end
```

### Some Dummy Examples

```
for i=1:100
    Some Matlab Commands;
end
```

```
for j=1:3:200
    Some Matlab Commands;
end
```

```
for m=13:-0.2:-21
    Some Matlab Commands;
end
```

```
for k=[0.1 0.3 -13 12 7 -9.3]
    Some Matlab Commands;
end
```

# Boucles itératives et vectorisation

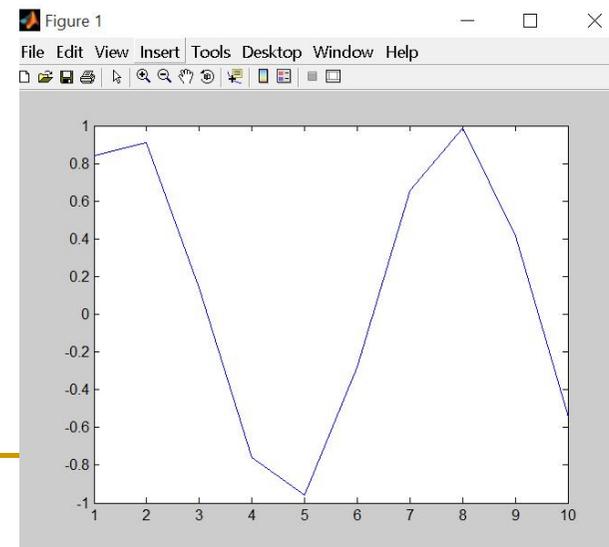
## ■ Boucle For

Subscript indices must either be real positive integers or logicals

```
tic;  
for i=1:10;  
    x1(i)=i ;  
    y1(i) = sin(i) ;  
end;  
toc;  
plot(x1,y1) ;
```

## ■ Vectorisation

```
tic;  
x=1:10 ;  
y=sin(x) ;  
toc;  
plot(x,y) ;
```



# Tracer des fonctions

## ■ Vectorisation

```
x=0:0.1:10
```

```
size (x)
```

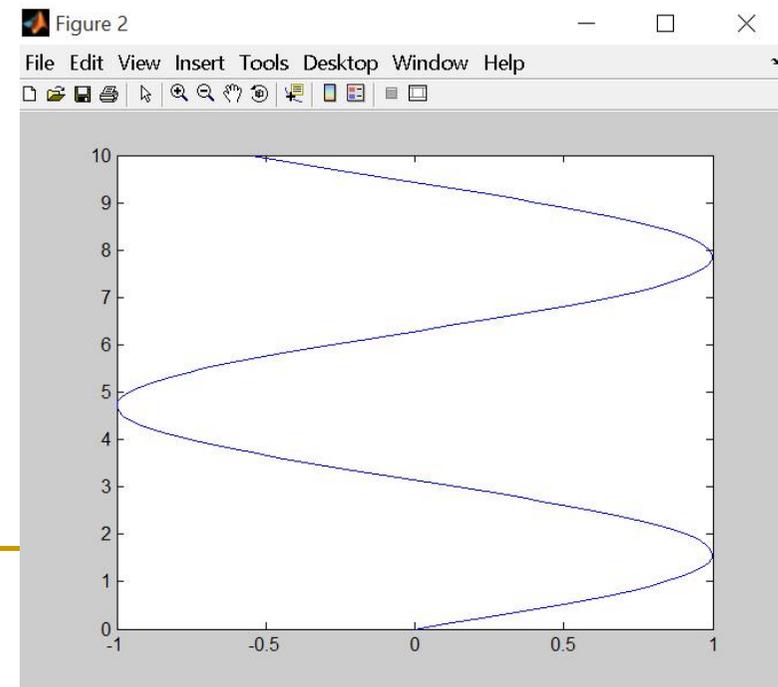
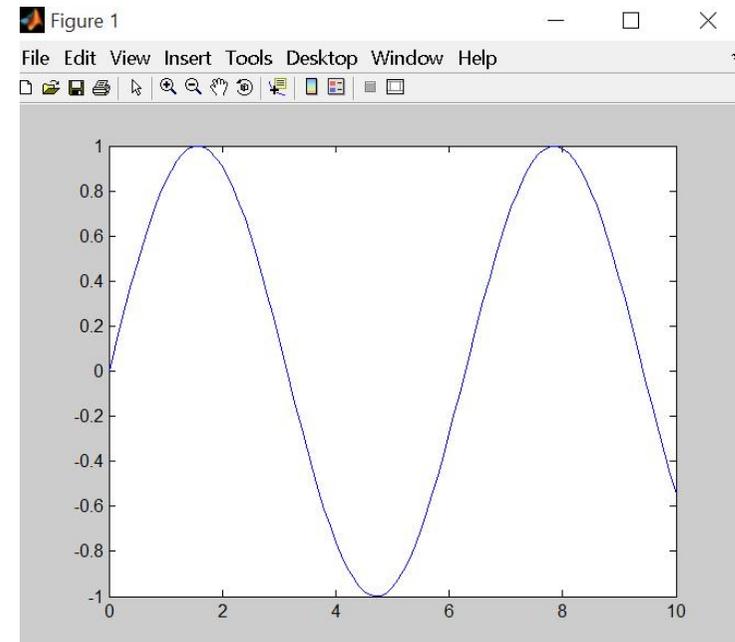
```
y=sin(x) ;
```

```
figure (1)
```

```
plot(x,y) ;
```

```
figure (2)
```

```
plot(y,x) ;
```



---

# Structure de controle

## ■ Boucle While

**while** (condition)

Commandes Matlab

**end**

### **Dummy Example**

```
while ((a>3) & (b==5))  
    Some Matlab Commands;  
end
```



---

# Définir des fonctions : 'inline'

- Fonction inline

```
g= inline('sin(t)+2');
```

```
h=inline('tan(x+theta)+c','theta','x','c')
```

```
g(2)
```

```
ans =
```

```
2.9093
```

```
h =
```

```
Inline function:  
h(theta,x,c) =  
tan(x+theta)+c
```

```
h(0,0,2)
```

```
ans =
```

```
2
```

---

---

# Définir une fonction 'snapshot'

- La fonction utilise les valeurs des variables du contexte.

```
c=2;
```

```
g=@(theta,x)(tan(x+theta)+c)
```

```
g(0,0)
```

```
ans =
```

```
2
```

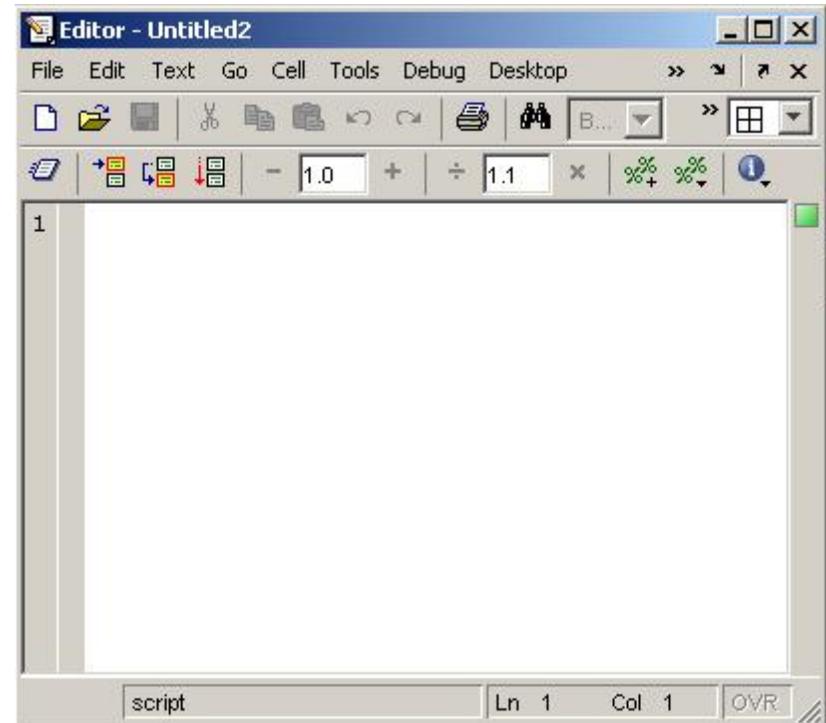
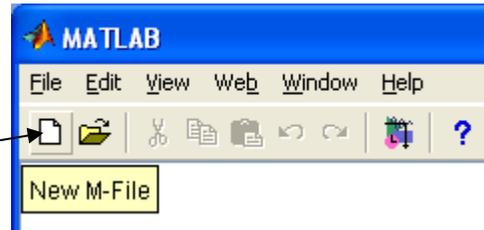
theta,x

Paramètres de la fonction

---

# Use of M-File

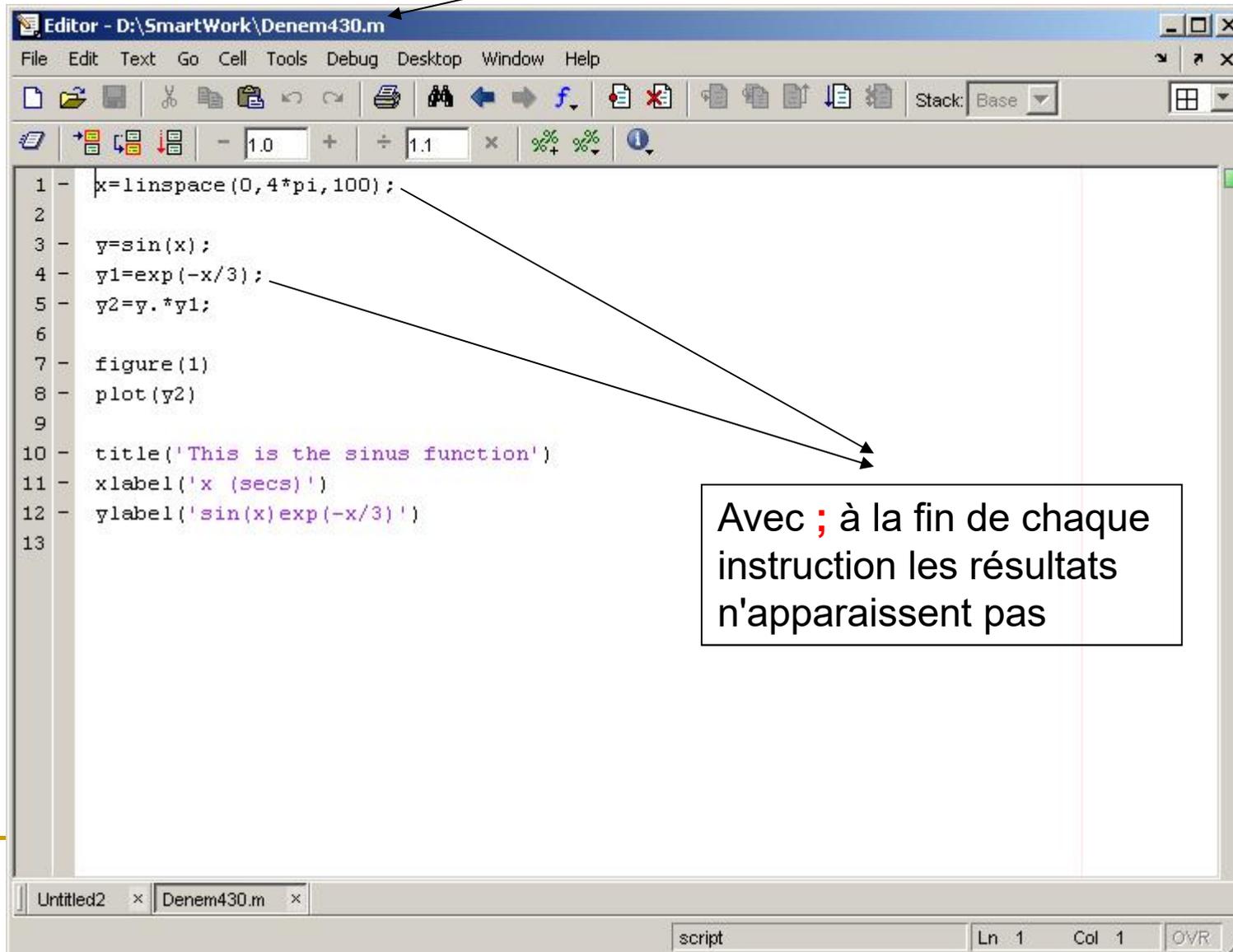
Cliquez pour créer un nouveau fichier M-File



- Extension “.m”
- Un fichier texte contenant un script ou une fonction ou un programme à exécuter

# Utilisation d'un fichier : M-File

Sauver le fichier : *Denem430.m*



```
1 - x=linspace(0,4*pi,100);
2
3 - y=sin(x);
4 - y1=exp(-x/3);
5 - y2=y.*y1;
6
7 - figure(1)
8 - plot(y2)
9
10 - title('This is the sinus function')
11 - xlabel('x (secs)')
12 - ylabel('sin(x)exp(-x/3)')
13
```

Avec ; à la fin de chaque instruction les résultats n'apparaissent pas

---

# Écriture d'une fonction utilisateur

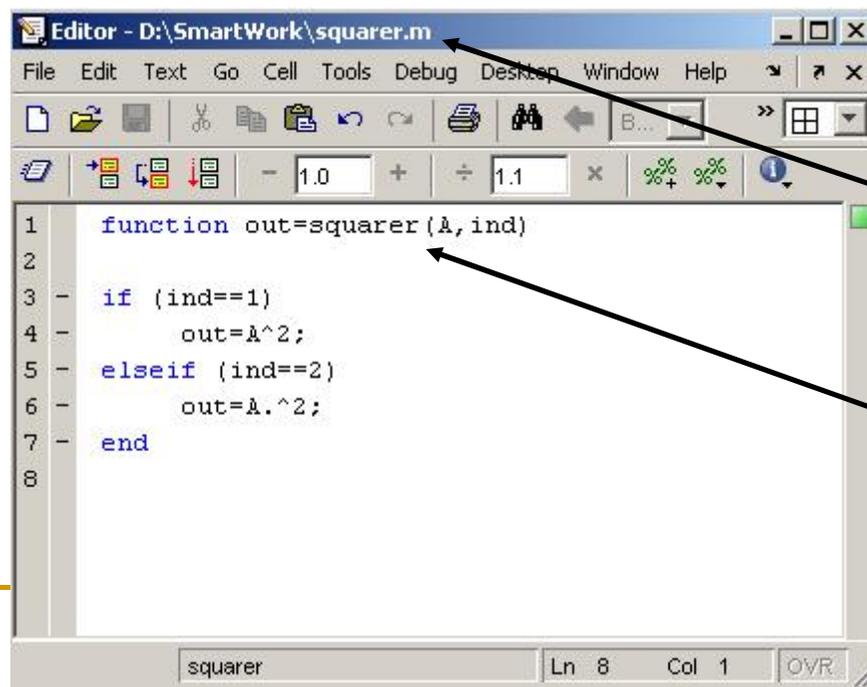
- Les fonctions sont des m-files qui peuvent être exécutés en spécifiant les arguments
- Le code indiquant qu'un m-file est une fonction est

```
function out1=functionname(in1)
function out1=functionname(in1,in2,in3)
function [out1,out2]=functionname(in1,in2)
```

- Vous devriez écrire cette commande au début du fichier.
  - Le nom du fichier est celui du nom de la fonction
-

# Écriture d'une fonction utilisateur

- Exemples
  - Ecrire la fonction : `out = squarer(A, ind)`
    - Qui prend le carré de la matrice `A` si l'indicateur est 1
    - Qui prend le carré des composantes de la matrice `A` si l'indicateur est 2



```
Editor - D:\SmartWork\squarer.m
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons]
[Icons]
1 function out=squarer(A, ind)
2
3 - if (ind==1)
4 -     out=A^2;
5 - elseif (ind==2)
6 -     out=A.^2;
7 - end
8
```

The screenshot shows a MATLAB editor window titled "Editor - D:\SmartWork\squarer.m". The window contains the following code:

```
1 function out=squarer(A, ind)
2
3 - if (ind==1)
4 -     out=A^2;
5 - elseif (ind==2)
6 -     out=A.^2;
7 - end
8
```

The status bar at the bottom indicates "squarer", "Ln 8", "Col 1", and "OVR".

Même nom  
pour le fichier  
et la fonction



---

## Remarques :

- « % » introduit un commentaire Matlab (équivalent de "//" en C). La ligne est négligée par le compilateur Matlab.
- Vous pouvez utiliser la commande « pause » pour ralentir l'exécution

```
pause %wait until any key  
pause(3) %wait 3 seconds
```

---

---

# Commandes utiles

- Les commandes les plus utilisées dans Matlab

`help` nom-de-la -fonction

`lookfor` mot-clé

---

---

# Le calcul symbolique

- Définir une ou des variables

```
a = sym('a')
```

```
syms b c
```

```
syms e
```

- Définir une expression

```
z = sin(a)^2 + cos(a)^2
```

- Simplifier une expression

```
simplify(z)
```

```
ans =
```

```
1
```

---

---

# Le calcul symbolique

`syms a z`

`z= 'sin(a)^2+cos(a)^2'`

- Simplifier marche pas !!

`simplify(z)`

`syms a z`

`z= sin(a)^2+cos(a)^2`

`simplify(z)`

---

---

# Dérivées symboliques

- Définir des variables

`syms x y`

ans =

- Définir une expression

$2*y+10$

$f = x^2 + (y + 5)^2$

- Dériver, intégrer

ans =

`diff(f,y)`

$\frac{1}{3}x^3 + (y+5)^2x$

`int(f,x)`

ans =

`int(f,y,0,10)`

$\frac{3250}{3} + 10x^2$

---

# Résoudre une équation symbolique

```
syms x
```

```
f= 2 * x^2 + 4 * x -8
```

```
solve(f,x)
```

```
eval(ans)
```

```
subs(f,x,2)
```

```
syms x
```

```
f= 2 * x^2 + 4 * x -8
```

```
b=solve(f,x)
```

```
eval(b)
```

```
b =
```

```
5^(1/2)-1  
-1-5^(1/2)
```

```
ans =
```

---

```
1.2361  
-3.2361
```

# Résoudre une équation

```
syms a b c x
```

```
s = a*x^2 + b*x + c;
```

```
solve(s,x)
```

- Pas de solution analytique

Matlab donne une solution numérique

```
[x,y] = solve('sin(x+y)-exp(x)*y = 0','x^2-y = 2')
```

```
??? [x,y] = solve(sin(x+y)-exp(x)*y = 0,x^2-y = 2)
```

---

Error: The expression to the left of the equals sign is not a valid target for an assignment.

---

# Résoudre les équations différentielles

- Résoudre équation différentielle : **dsolve**
- **D** représente la dérivée
- **D2** représente la deuxième dérivée

**syms** a;

**y = dsolve( 'Dy = -a\*y' )**

y =

C1\*exp(-a\*t)

---

---

```
syms a;
```

```
y = dsolve('Dy = -a*y')
```

```
syms C1 t ;
```

```
y= subs( y,a,2)
```

```
y= subs( y,C1,1);
```

```
t=1:10; y=subs(y,t);
```

```
plot(t,y)
```

---

# équations différentielles 2<sup>ème</sup> ordre

- **D2** représente la deuxième dérivée

**syms** x w02

b = 'D2x = -w02 \* x'

x = **dsolve** (b)

x=**dsolve**('D2x = -w02 \* x')

x =

C1\*sin(w02^(1/2)\*t)+C2\*cos(w02^(1/2)\*t)

# Calculs symboliques

- Donner une expression

`x = sym('x')`

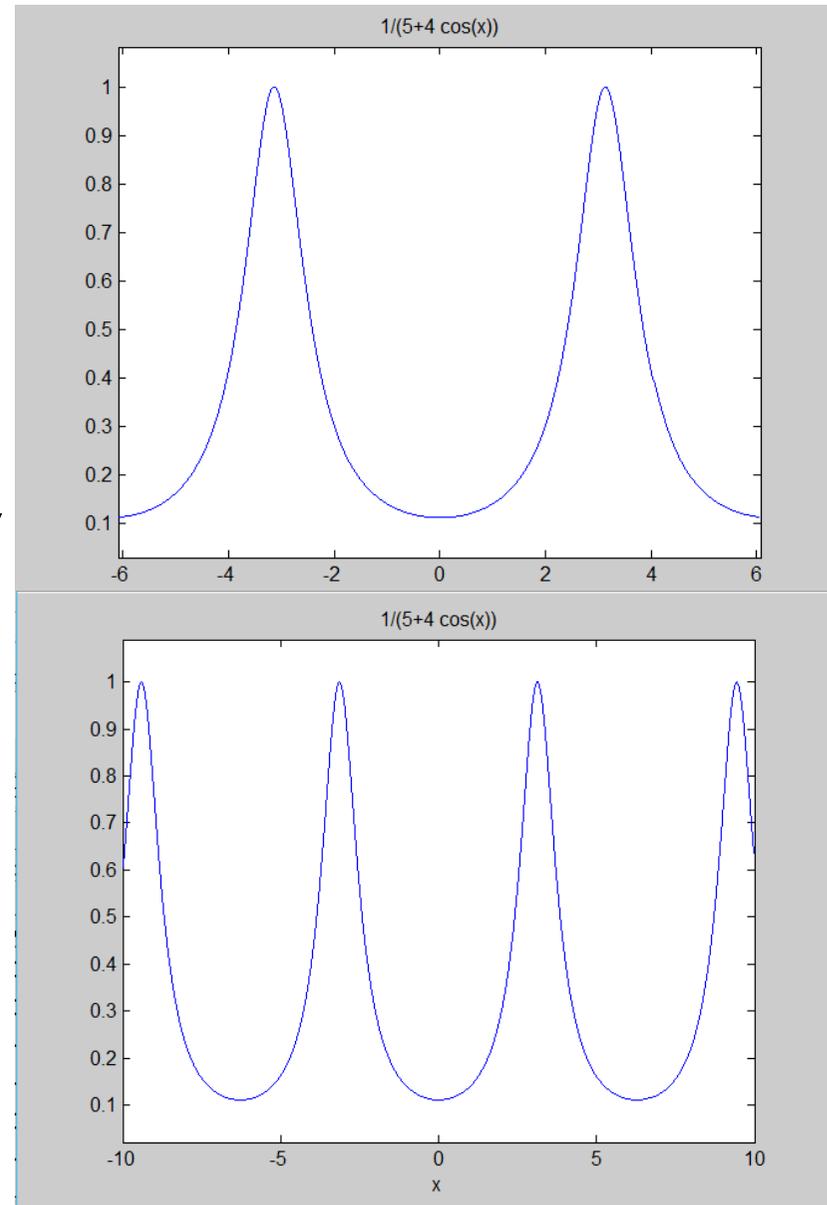
`f = 1/(5+4*cos(x))`

`ezplot(f)`

`ezplot(f,-10,10)`

`f1 = diff(f)`

`ezplot(f1)`



---

# équations différentielles 2<sup>ème</sup> ordre

## ■ Exemple avec conditions initiales

`syms x w0`

`b = 'D2x = -w0 * w0 * x'`

`x = dsolve(b, 'x(0)=1', 'Dx(0)=0')`

`simplify(x(1))`

`x=dsolve(b, 'x(0)=1', 'Dx(0)=0')`

`x =`

`cos(w0*t)`

---

# Calcul symbolique matriciel

- Définir les éléments symboliques

`syms a b c d e f g h`

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

$$C = A + B$$

$$D = A * B$$

$$E = \text{inv}(A)$$

$$A = \begin{bmatrix} a, b \\ c, d \end{bmatrix}$$

$$B = \begin{bmatrix} e, f \\ g, h \end{bmatrix}$$

$$C = \begin{bmatrix} a+e, b+f \\ c+g, d+h \end{bmatrix}$$

Produit lignes colonnes

$$D = \begin{bmatrix} a*e+b*g, a*f+b*h \\ c*e+d*g, c*f+d*h \end{bmatrix}$$

---

# Passage symbolique numérique

- Affectation des variables

a=1; b=2 ; c=3; d= 4; e= 5; f=6; g= 7; h=8;

eval(A); eval(B); eval(C);

- Inversion symbolique et valeurs numériques

D= inv(A)

D= eval(inv(A))

- Inversion numérique

DF= inv(eval(A))

---

---

# Questions

- ?
  - ?
  - ?
  - ?
  - ?
-

---

RDV

dans la salle informatique

---