



Complément de programmation

Contenu de la matière:

- **Chapitre 1.** Rappel sur les techniques de programmation et structuration des programmes
Présentation du programme Matlab
- **Chapitre 2.** Utilisation des procédures et fonction
Rappel sur le calcul matriciel :
Pivot de Gauss ; Gauss-Seidel ; Gauss-Jordan ; Décomposition LU
- **Chapitre 3.** Programmation modulaire
Calcul de la réponse d'une structure (DDS)
- **Chapitre 4.** Exemples d'application
Calcul des systèmes à treillis
Calcul des barres (proche éléments finis)



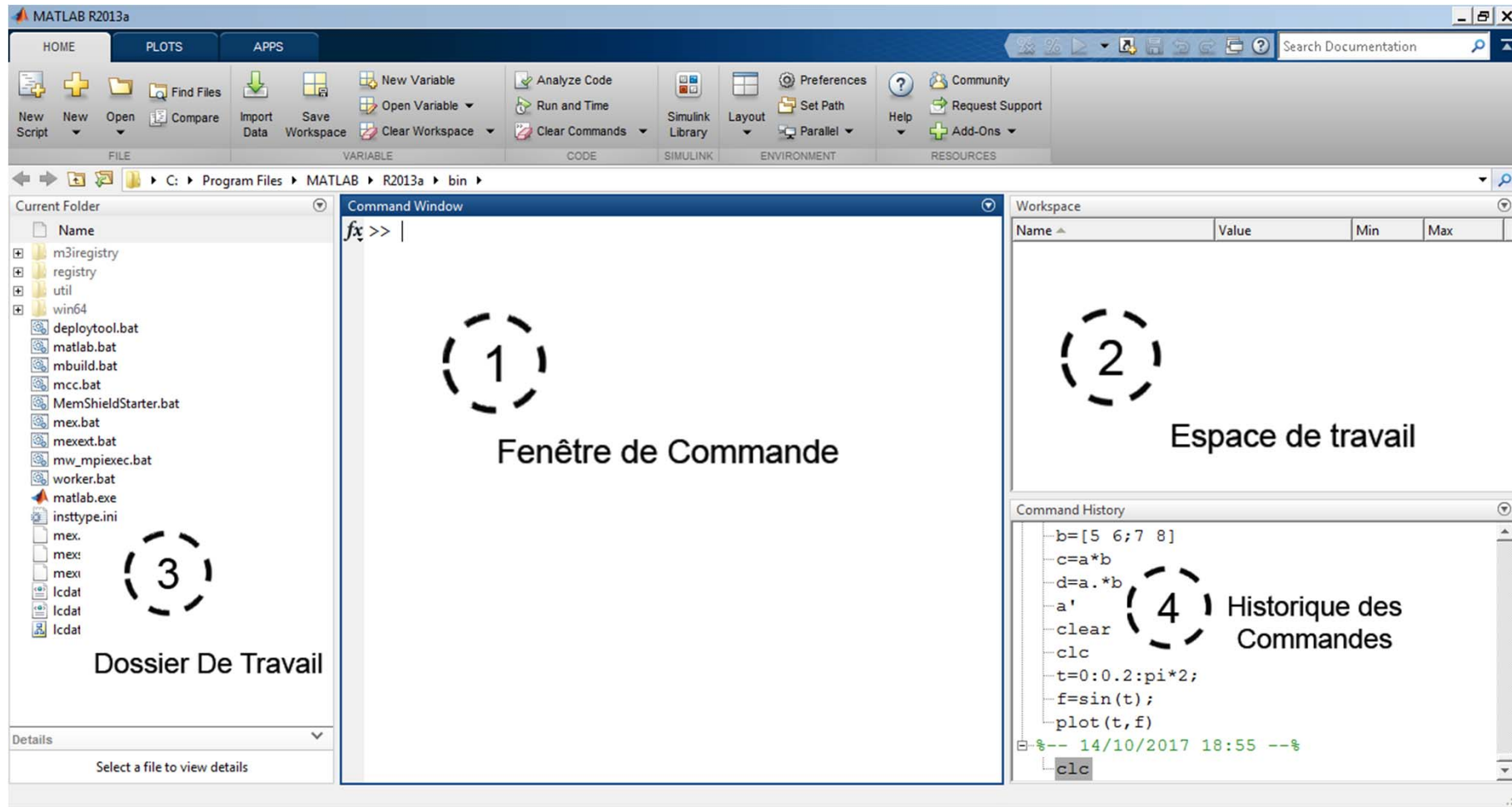
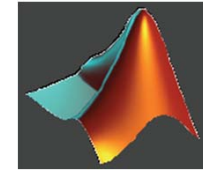
Mode d'évaluation: Contrôle continu: 40% ; Examen: 60%.

Références bibliographiques:

- « **Concepts in programming languages** », J.C. Mitchel, Prentice Hall 1997
- « **Méthodes numériques appliquées** », M. BOUMAH RAT, A. GOURDIN OPU 1993
- « **Calcul scientifique Cours, exercices corrigés et illustrations en MATLAB et Octave** », Alfio Quarteroni & Fausto Saleri, Springer, 2006. ([biblio](#))
- « **Analyse Numérique avec Matlab** », J.L. Merrien, Dunod, 2007. ([biblio](#))
- « **Matlab for engineers explained** », F. Gustafsson & N. Bergman, Springer, 2004. ([biblio](#))



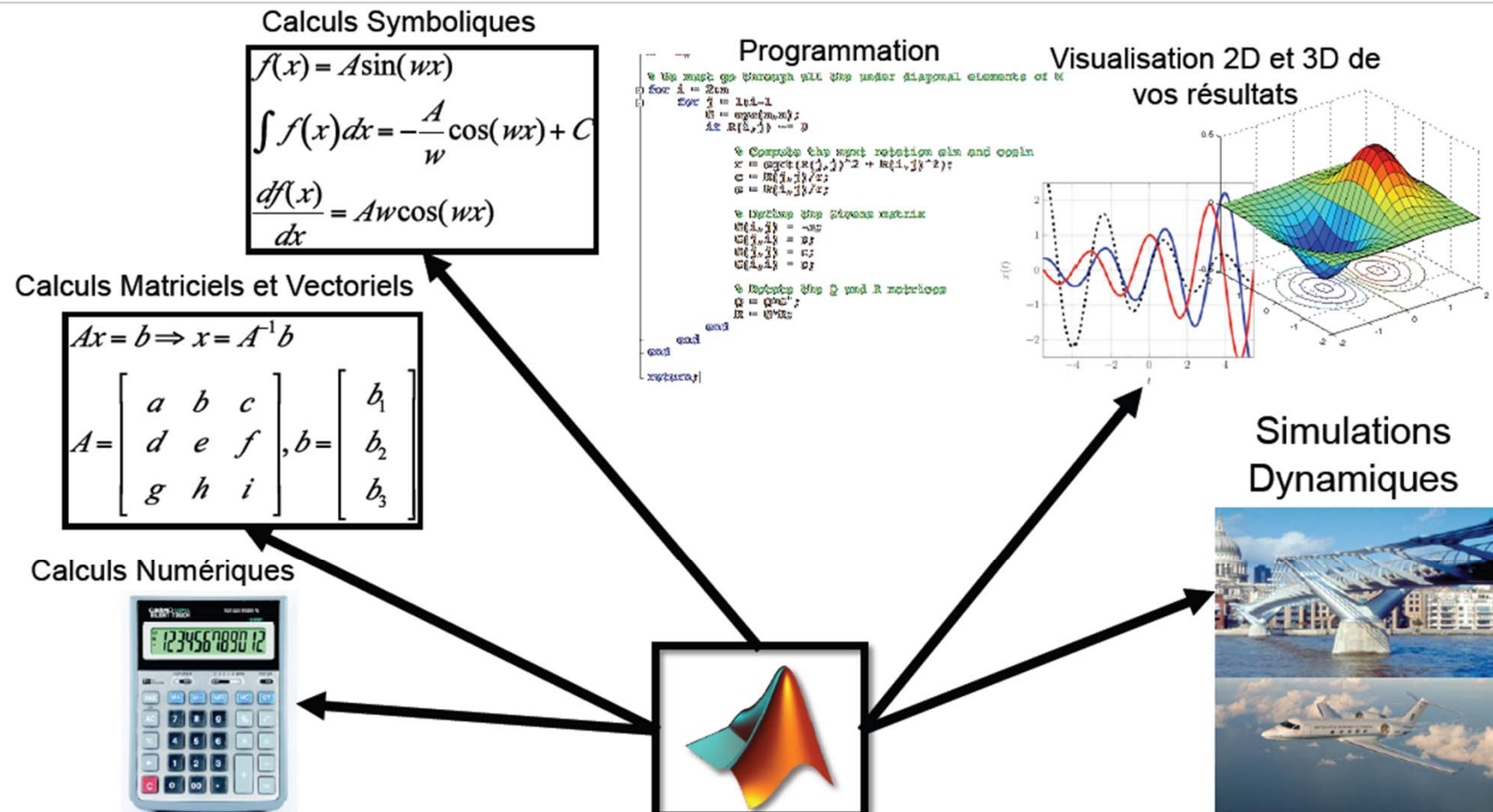
Interface MATLAB





MATLAB vous offre

- Quelques fonctionnalités MATLAB utiles pour ce cours





Calculs Numériques

- La fenêtre de commande vous permet de faire toute les opérations scientifiques possible :

The screenshot displays the MATLAB Command Window and Workspace. The Command Window shows a sequence of calculations: `>> sin(pi*2/3)` resulting in `ans = 0.8660`, `>> asin(0.8660)` resulting in `ans = 1.0471`, `>> log(2)` resulting in `ans = 0.6931`, and `>> exp(3)` resulting in `ans = 20.0855`. The Workspace window shows the variable `ans` with a value of `2`. Red arrows indicate the flow of data from the Command Window to the Workspace.

Command Window:

```
>> sin(pi*2/3)
ans =
    0.8660
>> asin(0.8660)
ans =
    1.0471
>> log(2)
ans =
    0.6931
>> exp(3)
ans =
    20.0855
```

Workspace:

Name	Value	Min	Max
ans	2	2	2

- L'espace de travail fait office de mémoire



Calcul Matriciels (1)

- Création de Variables : Vecteurs et Matrices

```
Command Window
>> x = [1,2,3,4]
x =
     1     2     3     4
Vecteur Ligne

>> y = [1; 2; 3; 4]
y =
     1
     2
     3
     4
Vecteur Colone

>> A = [1, 2, 3, 4; 5, 6, 7, 8; 9, 10, 11, 12; 13, 14, 15, 16]
A =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
Matrice carrée
```

Name ▲	Value
A	<4x4 double>
ans	2
x	[1,2,3,4]
y	[1;2;3;4]

- On peut aussi travailler sur les matrices :

$$Ax = b \Rightarrow x = A^{-1}b$$

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$



Calcul Matriciels (2)

◆ Multiplication Matricielle à Droite et Gauche :

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

>> y
y =
     1
     2
     3
     4

>> A*y
ans =
    30
    70
   110
   150
```

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

>> y
y =
     1
     2
     3
     4

>> y'*A
ans =
    90    100    110    120
```

La transposée

◆ Inversion Matricielle

```
>> A
A =
     1     0     1     0
     0     1     0     2
     1     0     3     0
     3     1     2     3

>> inv(A)
ans =
    1.5000     0    -0.5000     0
    7.0000    3.0000   -1.0000   -2.0000
   -0.5000     0     0.5000     0
   -3.5000   -1.0000     0.5000     1.0000
```

```
>> x = inv(A)*y
x =
     0
    2.0000
    1.0000
    0.0000

>> x = A\y
x =
     0
    2.0000
    1.0000
    0.0000
```



Calcul Matriciels (3)



- Création de suite de nombre :

```
>> x = 0:0.2:1
```

```
x =
```

```
0    0.2000    0.4000    0.6000    0.8000    1.0000
```

- Retravailler sur cette suite :

```
>> x/2
```

```
ans =
```

```
0    0.1000    0.2000    0.3000    0.4000    0.5000
```

```
>> x./x
```

```
ans =
```

```
NaN    1    1    1    1    1
```

- Accéder à un élément désirer :

```
>> x(1)
```

```
ans =
```

```
0
```

```
>> x(4)
```

```
ans =
```

```
0.6000
```



Calcul Symboliques (1)



- Vous pouvez faire des calculs mathématiques sans donner de valeur à vos variables

```
>> syms x
>> syms x real;
>> f = x^2 + x + 1

f =
x^2 + x + 1
```

$$\int f(x) dx = \frac{1}{3}x^3 + \frac{1}{2}x^2 + x + c$$

```
>> int(f)
ans =
(x*(2*x^2 + 3*x + 6))/6
```

$$\int_0^1 f(x) dx = \frac{1}{3}x^3 + \frac{1}{2}x^2 + x \Big|_0^1 = \frac{11}{6}$$

```
>> int(f,0,1)
ans =
11/6
```

$$\frac{df(x)}{dx} = 2x + 1$$

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

$$\left. \frac{df(x)}{dx} \right|_1 = 2x + 1 \Big|_1 = 3$$

```
>> df = diff(f)
df =
2*x + 1
>> x = 1;
>> df = eval(f)
df =
3
```




Calcul Symboliques (2)



- Vous pouvez résoudre des équations algébriques :

$$\frac{1}{3}x^2 + \frac{1}{2}x + 2 = 0$$

```
>> syms x;  
>> solve('x^2/3 + x/2 + 2')  
  
ans =  
- 3/4 - (3^(1/2)*29^(1/2)*i)/4  
- 3/4 + (3^(1/2)*29^(1/2)*i)/4
```

$$\begin{aligned} x + y &= 1 \\ x - 11y &= 5 \end{aligned}$$

```
>> syms x y;  
>> S = solve('x + y = 1', 'x - 11*y = 5')  
  
S =  
  
x: [1x1 sym]  
y: [1x1 sym]  
  
>> S.x  
  
ans =  
  
4/3  
  
>> S.y  
  
ans =  
  
-1/3
```

- Vous pouvez résoudre des équations différentielles ordinaires :

$$\begin{aligned} \dot{x}(t) &= -ax(t) \\ x(t) &= -K \exp(-at) \end{aligned}$$

```
>> syms x;  
>> dsolve('Dx = -a*x')  
  
ans =  
  
C2/exp(a*t)
```

$$\begin{aligned} \ddot{x}(t) + c\dot{x}(t) + kx(t) &= 0 \\ x(t) &= ? \end{aligned}$$

```
>> syms x  
>> dsolve('D2x + c*Dx + k*x = 0')  
  
ans =  
  
C4/exp(t*(c/2 - (c^2 - 4*k)^(1/2)/2))  
+ C5/exp(t*(c/2 + (c^2 - 4*k)^(1/2)/2))
```



Affichage 2D (1)



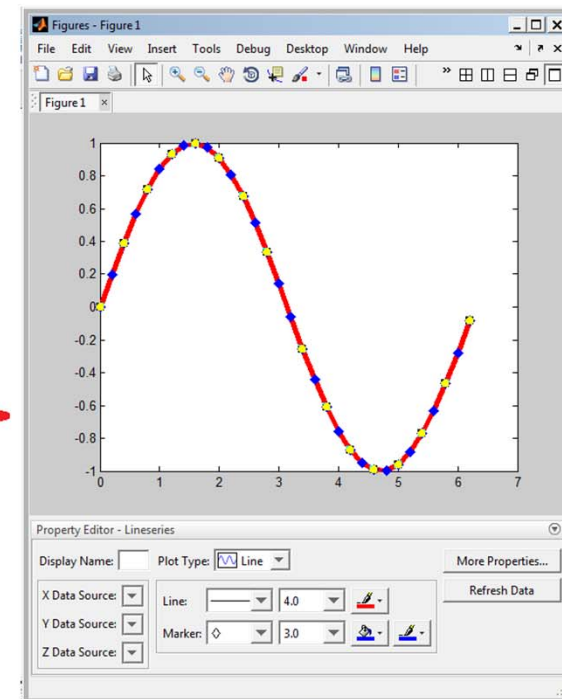
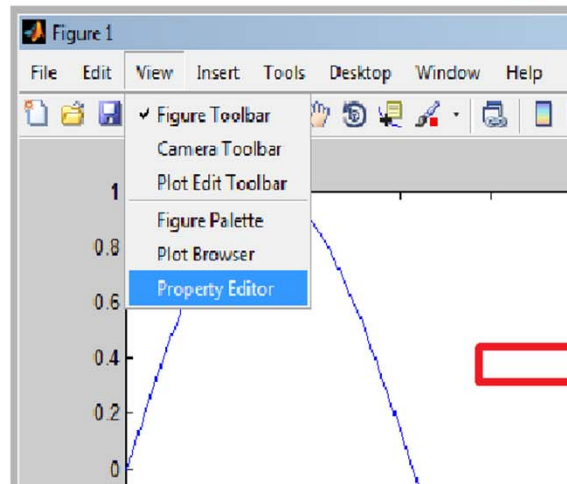
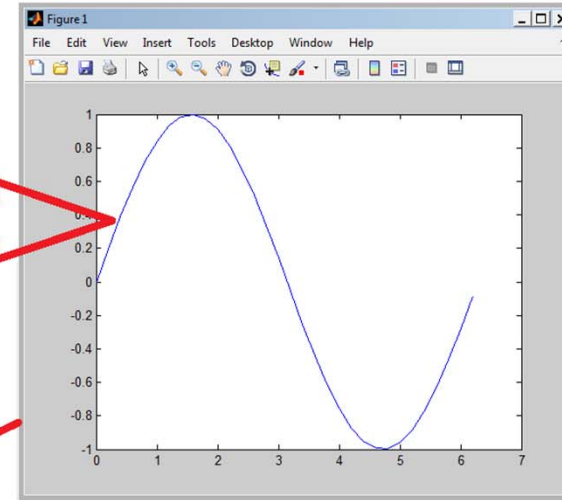
Vous pouvez afficher des graphes :

Methode 1:

```
>> t = 0:0.2:2*pi;  
>> f = sin(t);  
>> plot(t,f)  
>> |
```

Methode 2:

```
>> t = linspace(0,2*pi,30);  
>> f = sin(t);  
>> plot(t,f)  
>>
```





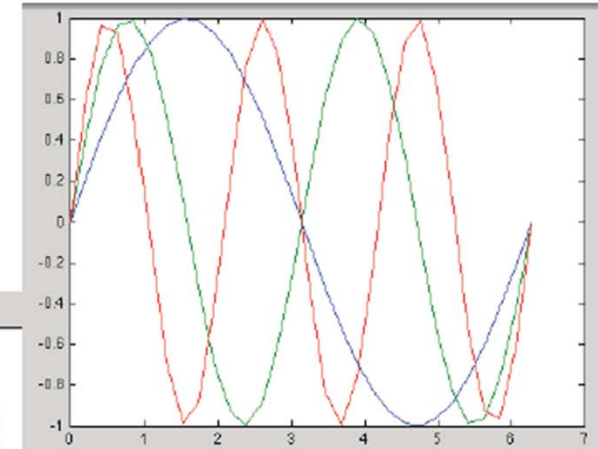
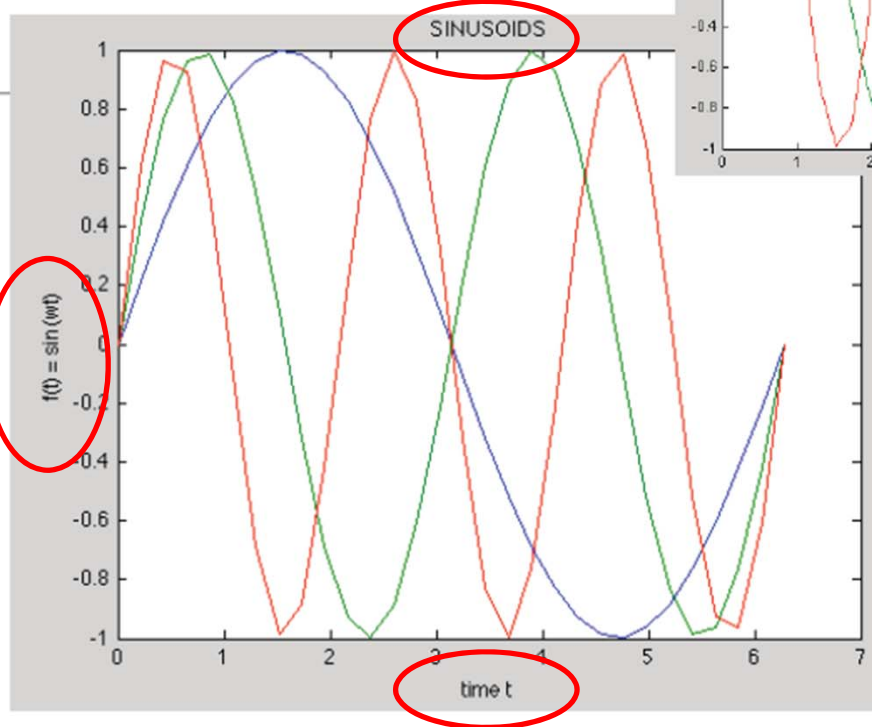
Affichage 2D (2)



- On peut mettre plusieurs graphes sur le même axe :

```
>> t = linspace(0,2*pi,30);  
>> f1 = sin(t);  
>> f2 = sin(2*t);  
>> f3 = sin(3*t);  
>> plot(t',[f1',f2',f3'])  
>> |
```

```
>> t = linspace(0,2*pi,30);  
>> f1 = sin(t);  
>> f2 = sin(2*t);  
>> f3 = sin(3*t);  
>> plot(t',[f1',f2',f3'])  
>> title('SINUSOIDS')  
>> xlabel('time t')  
>> ylabel('f(t) = sin(wt)')  
>> |
```

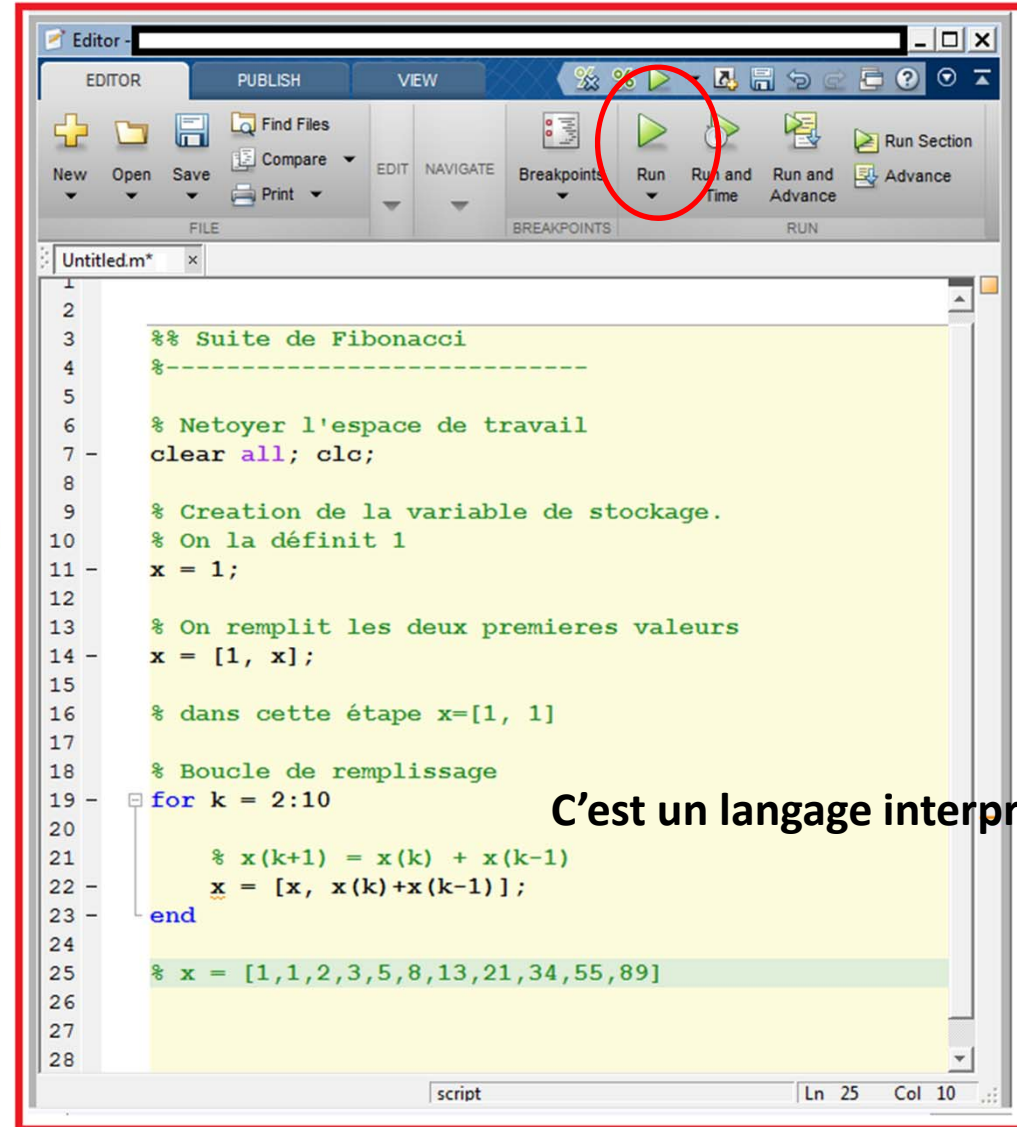
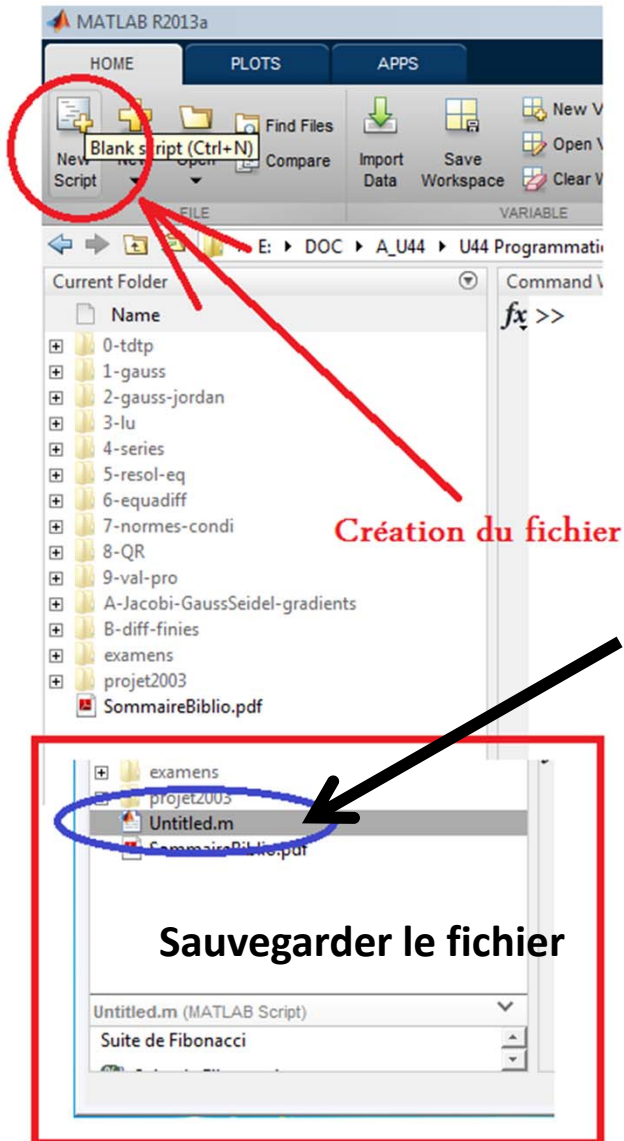




Programmation MATLAB (1)



- MATLAB permet de faire de la programmation :





Programmation MATLAB (2)



- MATLAB a aussi d'autres structures de décision :
- **if** : test sur conditions logique
- **for** : boucle de répétition
- **while** : boucle sur une condition
- **switch** : comparaison d'une variable à des cas connus

```
if expression 1
    code
else if expression 2
    code
else
    code
end
```

```
for compteur
    code
end
```

```
while expression
    code
end
```

```
switch expression
    case expression 1
        code
    case expression 2
        code
    otherwise
        code
end
```