*****Cours d'introduction à Matlab ***

 Matlab est un logiciel de calcul numérique.

Voici une introduction qui vous suffira largement si vous utilisez les outils mathématiques de façon ponctuelle.

N.B. Ce qui suit a été testé avec les versions 4.2. et 6.5.

[**Première partie : Eléments de base**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_index.htm#1)

[**Deuxième partie : Le langage de programmation Matlab : Utilisation des scripts**](http://pagesperso-orange.fr/fabrice.sincere/matlab/programmation/matlab_programmation.htm)

***Première partie : Eléments de base***

1. [**Utilisation de Matlab à la manière d’une calculatrice scientifique**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_calculatrice.htm)
2. [**Calcul sur les nombres complexes**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_nombre_complexe.htm)
3. [**Calcul sur les matrices**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_matrice.htm)
4. [**Résolution d’un système d’équations linéaires**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_systeme_equation_lineaire.htm)
5. [**Création du fichier .m d’une fonction y=f(x)**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fichier_dune_fonction.htm)
6. [**Création du fichier .m d’une fonction définie par morceaux y=f(x)**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fonction_par_morceaux.htm)
7. [**Graphe en 2D (2 axes)**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_plot.htm)
8. [**Graphe d’une fonction à une variable y = f (x)**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fplot.htm)
9. [**Graphe en 3D (3 axes)**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_plot3.htm)
10. [**Graphe d’une fonction à deux variables  z = f (x , y)**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_mesh.htm)
11. [**Calcul sur les polynômes**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_polynome.htm)
12. [**Recherche du minimun d’une fonction f(x)**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_minimum.htm)
13. [**Recherche de racines ; Equation non linéaire à une inconnue f(x)=0**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_zero.htm)
14. [**Dérivée d'une fonction f '(x)**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_derivate.htm)
15. [**Calcul d’intégrale**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_integrale.htm)
16. [**Equation différentielle**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_equation_differentielle.htm)
17. [**Diagramme de Bode**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_diagramme_bode.htm)
18. [**Scalaires, vecteurs, matrices et tableaux**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_scalaire_vecteur_matrice.htm)
19. [**Calcul numérique et calcul littéral**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_calcul_numerique.htm)
* [**Télécharger les fichiers .m**](http://pagesperso-orange.fr/fabrice.sincere/matlab/fichiers%20de%20fonction%20matlab.zip) 
* Site officiel de Matlab : [***www.mathworks.com***](http://www.mathworks.com/)
* A propos du logiciel libre [***Scilab***](http://pagesperso-orange.fr/fabrice.sincere/logiciel_libre.htm)

**(C) Fabrice Sincère**



***Matlab – Utilisation en calculatrice scientifique***

[**1-Variables et constantes spéciales**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_calculatrice.htm#ancre0)

[**2- Opérateurs mathématiques**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_calculatrice.htm#ancre1)

[**3- Fonctions mathématiques**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_calculatrice.htm#ancre2)

[**4- Utilisation de variables**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_calculatrice.htm#ancre3)

 **1-Variables et constantes spéciales**

|  |  |
| --- | --- |
| ans | réponse la plus récente |
| pi | nombre pi  |
| inf | plus l'infini |
| -inf | moins l'infini |
| NaN | (Not-a-Number)  |

**2-Opérateurs mathématiques**

|  |  |
| --- | --- |
| + | addition  |
| - | soustraction  |
| \* | multiplication  |
| / | division  |
| ^ | puissance |

**>> (2 + 5.2)\*10 / (5^3)**

ans =

 0.5760

**>> -2.52e3**

ans =

-2520

**>> 2\*pi**

ans =

6.2832

*format long e* affiche 16 chiffres :

**>> format long e**

**>> 2\*pi**

ans =

6.283185307179586e+000

 format short (format par défaut) affiche 5 chiffres :

**>> format short**

**>> 2\*pi**

ans =

6.2832

 **>> 1 / 0**

Warning: Divide by zero

ans =

Inf

**>> -1 / 0**

Warning: Divide by zero

ans =

-Inf

**>> 0 / 0**

Warning: Divide by zero

ans =

NaN

**3- Fonctions mathématiques**

|  |  |
| --- | --- |
| sin(X) | sinus  |
| asin(X) | sinus inverse  |
| cos(X) | cosinus  |
| acos(X) | cosinus inverse  |
| tan(X) | tangente |
| atan(X) | tangente inverse |

avec X : argument en **radians.**

|  |  |
| --- | --- |
| exp(X) | exponentielle  |
| log(X) | logarithme naturel (base e)  |
| log10(X) | logarithme décimal (base 10)  |
| sqrt(X) | racine carrée  |
| abs(X) | valeur absolue |

 **>> sin(2)**

ans =

0.9093

sinus (45°) :

**>> sin(45\*pi/180)**

ans =

0.7071

**>> 1 + exp(2.5)**

ans =

13.1825

**4- Utilisation de variables**

**>> 5\*3**

ans =

15

**>> ans+4**

ans =

19

**>> a= 2 + log(15)**

a =

4.7081

**>> b = - 45**

b =

-45

**>> a \* b**

ans =

-211.8623

**>> c = a - sqrt(abs(b))**

c =

-2.0002

**Matlab - Calcul sur les nombres complexes**

* Fonctions :

|  |  |
| --- | --- |
| i | imaginaire pur |
| j | imaginaire pur |
| conj(X) | conjugué du nombre complexe X |
| real(X)  | partie réelle |
| imag(X)  | partie imaginaire |
| abs(X) | module |
| angle(X) | argument (en radians) |

* Exemples :

**>> (4 - 2.5i)\*(-2 + i)/(1 + i)**

ans =

1.7500 + 7.2500i

**>> a = 1 + i**

a =

1.0000 + 1.0000i

**>> b = -2 + 3.5j**

b =

-2.0000 + 3.5000i

**>> a + b**

ans =

-1.0000 + 4.5000i

**>> a \* b**

ans =

-5.5000 + 1.5000i

**>> a / b**

ans =

0.0923 - 0.3385i

**>> conj(a)**

ans =

1.0000 - 1.0000i

**>> a \* conj(a)**

ans =

2

**>> real(a)**

ans =

1

**>> imag(conj(a))**

ans =

-1

**>> abs(a)**

ans =

1.4142

**>> angle(a)**

ans =

0.7854

*sqrt* : fonction racine carrée

**>> c = 2 - sqrt(3)\*i**

c =

2.0000 - 1.7321i

**>> abs(c)**

ans =

2.6458

**>> angle(c)**

ans =

-0.7137

Argument en degrés :

**>> angle(c)\*180/pi**

ans =

-40.8934

**Matlab **

**Calcul matriciel**

* Fonctions :

|  |  |
| --- | --- |
| **+**  | addition de matrices  |
| **-**  | soustraction de matrices  |
| **\***  | produit de matrices  |
| **^**  | puissance  |
| **eye (n)** | matrice unité (matrice identité) de taille n x n |
| **inv (X)**  | inverse de la matrice carrée X  |
| **rank (X)**  | rang de la matrice X (nombre de colonnes ou de lignes indépendantes)  |
| **det (X)**  | déterminant de la matrice carrée X  |
| **X '**  | transposée de la matrice X  |
| **/**  | division à droite : A / B est équivalent à A \* inv(B)  |
| **\**  | division à gauche : A \ B est équivalent à  inv(A) \* B  |

* *Exemples* :

Saisie d'une matrice carrée de taille 3 x 3 :

**>> A = [ 2 4 5 ; 1 5 7 ; -3 3 1]**

A =

2 4 5

1 5 7

-3 3 1

**>> A(2 , 3)**

ans =

7

**>> A(2 , 3) = 6**

A =

2 4 5

1 5 6

-3 3 1

**>> A'**

ans =

2 1 -3

4 5 3

5 6 1

**>> inv(A)**

ans =

1.0833 -0.9167 0.0833

1.5833 -1.4167 0.5833

-1.5000 1.5000 -0.5000

**>> D = A \* inv(A)**

D =

1.0000 0.0000 0.0000

0.0000 1.0000 0.0000

0.0000 0.0000 1.0000

**>> rank(A)**

ans =

3

**>> det(A)**

ans =

-12

**>> eye(7)**

ans =

1 0 0 0 0 0 0

0 1 0 0 0 0 0

0 0 1 0 0 0 0

0 0 0 1 0 0 0

0 0 0 0 1 0 0

0 0 0 0 0 1 0

0 0 0 0 0 0 1

**>> B = [ 1 1 0 ; 1 0 1 ; 0 1 1 ]**

B =

1 1 0

1 0 1

0 1 1

**>> A + B**

ans =

3 5 5

2 5 7

-3 4 2

**>> 2 + A**

ans =

4 6 7

3 7 8

-1 5 3

**>> 2 \* A**

ans =

4 8 10

2 10 12

-6 6 2

**>> A \* B**

ans =

6 7 9

6 7 11

0 -2 4

**>> B \* A**

ans =

3 9 11

-1 7 6

-2 8 7

**>> A\*A\*A**

ans =

-88 304 262

-98 314 268

-18 18 10

**>> A^3**

ans =

-88 304 262

-98 314 268

-18 18 10

Saisie d'une matrice à coefficients complexes de taille 2 x 3  :

**>> C = [ 1 + i 0 0 ; 1 - i i 2]**

C =

1.0000 + 1.0000i 0 0

1.0000 - 1.0000i 0 + 1.0000i 2.0000

**>> C \* A**

ans =

2.0000 + 2.0000i 4.0000 + 4.0000i 5.0000 + 5.0000i

-4.0000 - 1.0000i 10.0000 + 1.0000i 7.0000 + 1.0000i

**(C) Fabrice Sincère**

Matlab 

Résolution d'un système d'équations linéaires

* Fonction :

|  |  |
| --- | --- |
| **\** | division à gauche de matrices**A \ B** est équivalent à : **inv(A)\*B** |

* Exemple n°1 :

Soit à résoudre un système de 3 équations à 3 inconnues x1, x2 et x3 :



On saisit les différents coefficients dans une matrice 3 x 3 :

**>> A = [ 3 2 1 ; -1 5 2 ; 4 -2 3 ]**

A =

3 2 1

-1 5 2

4 -2 3

On complète avec un vecteur colonne 3 x 1 :

**>> B = [ 4 ; -1 ; 3 ]**

B =

4

-1

3

La solution est donnée sous la forme d'un vecteur colonne :

**>> A \ B**

ans =

1.3279

0.2951

-0.5738

Autre méthode :

**>> inv(A) \* B**

ans =

1.3279

0.2951

-0.5738

**>> format long e**

**>> A \ B**

ans =

1.327868852459016e+000

2.950819672131148e-001

-5.737704918032788e-001

**>> X = A \ B**

X =

1.327868852459016e+000

2.950819672131148e-001

-5.737704918032788e-001

**>> X(1)**

ans =

1.327868852459016e+000

**>> X(2)**

ans =

2.950819672131148e-001

**>> X(3)**

ans =

-5.737704918032788e-001

Vérification :

**>> 3\*X(1) + 2\*X(2) + X(3)**

ans =

4

**>> -X(1) + 5\*X(2) + 2\*X(3)**

ans =

-9.999999999999998e-001

**>> 4\*X(1) -2\*X(2) + 3\*X(3)**

ans =

3.000000000000000e+000

* Exemple n°2 :

Soit le système d'équations paramétriques :



On cherche à exprimer x1, x2 et x3 en fonction de b1, b2 et b3 :

**>> A = [ -1 2 1 ; -1 1 2 ; 1 -2 1 ]**

A =

-1 2 1

-1 1 2

1 -2 1

**>> inv(A)**

ans =

2.5000 -2.0000 1.5000

1.5000 -1.0000 0.5000

0.5000 0 0.5000

**>> format rational**

**>> inv(A)**

ans =

5/2 -2 3/2

3/2 -1 1/2

1/2 0 1/2

**>> format short**

**>> inv(A)**

ans =

2.5000 -2.0000 1.5000

1.5000 -1.0000 0.5000

0.5000 0 0.5000

Finalement :



**(C) Fabrice Sincère**

**Matlab - Création du fichier .m d’une fonction**



* Fonction :

function

* Exemple

Soit la fonction :



a) Commencez pour ouvrir un éditeur de texte :

Dans la fenêtre de commande de Matlab :

File -> New -> M-file

Avec la version 4.2. l'éditeur de texte par défaut est l'application "Bloc-notes".

Avec la version 6.5. l'éditeur de texte par défaut est l'application "M-File Editor".

b) Donnez un nom à cette fonction (dans cet exemple *fonc)* et saisissez son expression mathématique :



**Attention :** il faut mettre un [point](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_scalaire_vecteur_matrice.htm) devant les opérateurs multiplication, division et puissance :

**.\***

**./**

**.^**

c) Sauvegardez le fichier dans votre répertoire de travail (par exemple c:\USERS)

Nom : fonc

Extension : .m

d) Ajoutez le chemin du répertoire où se trouve votre fichier fonc.m

Avec la version 4.2. :

**>> path(path,'c:\USERS')**

Avec la version 6.5. :

File -> Set Path -> Add Folder

-> Save -> Close

Remarque : Vous pouvez utiliser n'importe quel nom pour les variables.

La fichier suivant donne le même résultat :



* Evaluation d'une fonction

Calcul de y ( x = 0 ) :

**>> fonc(0)**

ans =

2

Calcul de y ( x = 5 ) :

**>> fonc(5)**

ans =

10.2010

**>> fonc(-1)**

Warning: Divide by zero

ans =

Inf

Avec en argument un [vecteur](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_scalaire_vecteur_matrice.htm), la fonction retourne un vecteur :

**>> fonc( [0 1 2 3 4 5] )**

ans =

2.0000 3.8415 3.9099 6.9121 8.1121 10.2010

**>> x = 0 : 5**

x =

0 1 2 3 4 5

**>> y = fonc(x)**

y =

2.0000 3.8415 3.9099 6.9121 8.1121 10.2010

Avec en argument une [matrice](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_scalaire_vecteur_matrice.htm), la fonction retourne une matrice :

**>> fonc( [ 1 2 3 ; 4 5 6] )**

ans =

3.8415 3.9099 6.9121

8.1121 10.2010 11.2939

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



Matlab - Création du fichier .m d'une fonction définie par morceaux



* **Exemple :**

Soit la fonction y = f(x) :



Le fichier f5.m associé à cette fonction s'écrit :



**>> f5 (0)**

ans =

0

**>> f5 (1)**

ans =

0.5000

**>>f5 (-10)**

ans =

0

**>> f5 (3)**

ans =

0.0556

**>> fplot ( 'f5' , [ -2 4 ] )**

**>> grid on**



© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



Matlab - Création du fichier .m d'une fonction définie par morceaux



* **Exemple :**

Soit la fonction y = f(x) :



Le fichier f5.m associé à cette fonction s'écrit :



**>> f5 (0)**

ans =

0

**>> f5 (1)**

ans =

0.5000

**>>f5 (-10)**

ans =

0

**>> f5 (3)**

ans =

0.0556

**>> fplot ( 'f5' , [ -2 4 ] )**

**>> grid on**



© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



Matlab - Représentation graphique d'une fonction à une variable y = f(x)





* Fonctions

|  |  |
| --- | --- |
| fplot  | trace point par point le graphe d'une fonction |
| grid  | ajoute une grille |
| xlabel  | ajoute une légende pour l'axe des abscisses |
| ylabel  | ajoute une légende pour l'axe des ordonnées |
| title | ajoute un titre |
| axis  | modifie les échelles des axes |
| zoom  | effectue un zoom |
| gtext | place une légende avec la souris |
| hold | ajoute un graphe dans la fenêtre courante |
| figure | crée une nouvelle fenêtre |

* Première méthode :

On veut tracer le graphe de la fonction :



**>> fplot('1+ 2\*x + sin(x\*x)', [ 1 5 ])**

Remarque : la variable doit nécessairement s'appeler x



**>> fplot('1+ 2\*x + sin(x\*x)', [ 2 3 2 10 ])**



**>> grid on**



**>> grid off**

**>> xlabel('axe des abscisses')**

**>> ylabel('axe des ordonnées')**

**>> title('y=f(x)')**



**>> zoom on**

click droit : zoom arrière

click gauche : zoom avant

click gauche et glissé : zoom d’une zone

**>> zoom off**

Pour tracer plusieurs graphes dans la même fenêtre :

**>> fplot('[1+ 2\*x + sin(x\*x) , 1+ 2\*x - sin(x\*x) ]', [ -2 2.5 ])**



**>> gtext('fonction 1')**

**>> gtext('fonction 2')**







* Deuxième méthode :



Nous allons [créer le fichier .m de la fonction](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fichier_dune_fonction.htm) :



**>> fplot('f2', [ 0 10])**



**>> hold on**

**>> fplot('sin(x) ', [ 0 10 ] ,'b')**



**>> hold off**

**>> [X Y] = fplot ( 'sin(x) ' , [ 0 10 ] )**

X =

0

0.0200

0.0600

...

9.8700

9.9350

10.0000

Y =

0

0.0200

0.0600

...

-0.4307

-0.4884

-0.5440

**>> fplot ( 'sin(x) ' , [ 0 10 ] , '.' )**



Autres options :

|  |  |
| --- | --- |
| y | Jaune |
| m | Magenta |
| c | Cyan |
| r | Rouge |
| g | Vert |
| b | Bleu |
| w | Blanc |
| k | Noir |

|  |
| --- |
| **.** |
| **o** |
| **x** |
| **+** |
| **-** |
| **\*** |
| **:** |
| **-.** |
| **--** |

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



**Matlab- Graphe en trois dimensions**





* Fonctions

|  |  |
| --- | --- |
| plot3 | trace point par point un graphe 3D |
| grid  | ajoute une grille |
| xlabel  | ajoute une légende pour l'axe des abscisses |
| ylabel  | ajoute une légende pour l'axe des ordonnées |
| zlabel | ajoute une légende pour l'axe des z |
| title | ajoute un titre |
| hold | ajoute un graphe dans la fenêtre courante |
| figure | crée une nouvelle fenêtre |
| view | ajuste l'angle de vue |

* Exemple 1

**>> x = [ 1 2 3 4 ]**

x =

1 2 3 4

**>> y = [ 2 4 6 7]**

y =

2 4 6 7

**>> z = [ 3 8 5 6 ]**

z =

3 8 5 6

**>> plot3 (x , y , z ,'r\*')**

**>> grid on**

**>> xlabel('axe des x')**

**>> ylabel('axe des y')**

**>> zlabel('axe des z')**

**>> title('plot3')**







* Exemple 2 : équations paramétriques

x = cos (t)

y = sin (t)

z = t

**>> t = 0 : pi/100 : 5\*pi**

**>> plot3 (cos(t) , sin(t) , t)**



© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



**Matlab- Graphe d’une fonction à deux variables z = f(x,y)**



* Fonctions

|  |  |
| --- | --- |
| meshgrid | (voir l'exemple) |
| mesh | (voir l'exemple) |
| meshc | (voir l'exemple) |
| meshz | (voir l'exemple) |
| contour | (voir l'exemple) |
| view | ajuste l'angle de vue  |
| grid  | ajoute une grille |
| xlabel  | ajoute une légende pour l'axe des abscisses |
| ylabel  | ajoute une légende pour l'axe des ordonnées |
| zlabel | ajoute une légende pour l'axe des z |
| title | ajoute un titre |
| hold | ajoute un graphe dans la fenêtre courante |
| figure | crée une nouvelle fenêtre |

* Exemple 1



>> **x = -2 : 2**

x =

-2 -1 0 1 2

**>> y = -3 : 3**

y =

-3 -2 -1 0 1 2 3

**>> [X , Y] = meshgrid(x , y)**

X =

-2 -1 0 1 2

-2 -1 0 1 2

-2 -1 0 1 2

-2 -1 0 1 2

-2 -1 0 1 2

-2 -1 0 1 2

-2 -1 0 1 2

Y =

-3 -3 -3 -3 -3

-2 -2 -2 -2 -2

-1 -1 -1 -1 -1

0 0 0 0 0

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

**>> Z = sqrt (X.^2 + 2.\*Y.^2)**

Z =

4.6904 4.3589 4.2426 4.3589 4.6904

3.4641 3.0000 2.8284 3.0000 3.4641

2.4495 1.7321 1.4142 1.7321 2.4495

2.0000 1.0000 0 1.0000 2.0000

2.4495 1.7321 1.4142 1.7321 2.4495

3.4641 3.0000 2.8284 3.0000 3.4641

4.6904 4.3589 4.2426 4.3589 4.6904

**>> mesh (X , Y , Z)**

**>> grid on**



Pour une meilleure résolution :

**>> x = -2 : 0.1 : 2**

**>> y = -3 : 0.1 : 3**

**>> [X , Y] = meshgrid(x , y)**

**>> Z = sqrt (X.^2 + 2.\*Y.^2)**

**>> mesh (X , Y , Z)**



**>> meshc (X , Y , Z)**



**>> contour (x , y , Z)**



**>> meshz (X , Y , Z)**



**>> mesh (X , Y , Z)**

**>> grid on**

**>> view(-80 , 10)**



Remarque : view (-37.5 , 30) par défaut

* Exemple 2

z = f (x , y ) = y.exp( - x² - y² )

**>> x = -2 : 0.1 : 2**

**>> y= -2 : 0.1 : 2**

**>> [X , Y] = meshgrid(x , y)**

**>> Z = Y.\*exp ( -X.^2 - Y.^2)**

**>> mesh (X , Y , Z)**



**>> meshc (X , Y , Z)**



**>> contour (x , y , Z)**



© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



Matlab - Calcul sur les polynômes





[**1- Racines d'un polynôme**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_polynome.htm#1)

[**2- Détermination des coefficients d’un polynôme à partir des ses racines**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_polynome.htm#2)

[**3- Produit de polynômes**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_polynome.htm#3)

[**4- Fractions rationnelles : Décomposition en éléments simples**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_polynome.htm#4)

[**5- Représentation graphique**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_polynome.htm#5)

* Fonctions :

|  |  |
| --- | --- |
| conv | produit de polynômes |
| residue | décomposition en éléments simples |
| roots | trouve les racines d'un polynôme |
| poly | trouve le polynôme à partir des ses racines |
| polyval  | évalue le polynôme |





**1- Racines d'un polynôme**

* 1er exemple :

3x² - 5x + 2 = 0

On commence par définir un " vecteur " qui contient les coefficients du polynôme :

**>> p = [ 3 -5 2 ]**

p =

3 -5 2

**>> roots(p)**

ans =

1.0000

0.6667

**>> roots( [ 3 -5 2 ])**

ans =

1.0000

0.6667

* 2ème exemple :

x² - 4x + 4 = 0

**>> p= [ 1 -4 4 ]**

p =

1 -4 4

**>> roots(p)**

ans =

2

2

* 3ème exemple :

x² + 3x + 8 = 0

**>> p= [ 1 3 8 ]**

p =

1 3 8

**>> roots(p)**

ans =

-1.5000 + 2.3979i

-1.5000 - 2.3979i

* 4ème exemple :



**>> p = [ 1 2 -2 4 3 5 ]**

p =

1 2 -2 4 3 5

**>> roots(p)**

ans =

-3.0417

0.9704 + 1.0983i

0.9704 - 1.0983i

-0.4495 + 0.7505i

-0.4495 - 0.7505i

**>> format long e**

**>> roots(p)**

ans =

-3.041684725314715e+000

9.703604093970790e-001 +1.098343294996758e+000i

9.703604093970790e-001 -1.098343294996758e+000i

-4.495180467397220e-001 +7.504870344729816e-001i

-4.495180467397220e-001 -7.504870344729816e-001i

* 5ème exemple :

polynôme à coefficients complexes :

(1+i)x² + (2-5i)x + 3,5 = 0

**>> format short**

**>> p = [ 1+i 2-5i 3.5]**

p =

1.0000 + 1.0000i 2.0000 - 5.0000i 3.5000

**>> roots(p)**

ans =

1.7116 + 4.0248i

-0.2116 - 0.5248i





**2- Détermination des coefficients d’un polynôme à partir des ses racines**

* 1er exemple :

**>> a = [ 2 1 ]**

a =

2 1

**>> poly(a)**

ans =

1 -3 2

(c’est-à-dire : x² -3x +2)

* 2ème exemple :

**>> a = [ 2 2 3 -5 ]**

a =

2 2 3 -5

**>> poly(a)**

ans =

1 -2 -19 68 -60

* 3ème exemple :

**>> a = [ 2+i 2-3i 5]**

a =

2.0000 + 1.0000i 2.0000 - 3.0000i 5.0000

**>> poly(a)**

ans =

1.0000 -9.0000 + 2.0000i 27.0000 -14.0000i -35.0000 +20.0000i

Vérification :

**>> p = ans**

p =

1.0000 -9.0000 + 2.0000i 27.0000 -14.0000i -35.0000 +20.0000i

**>> roots(p)**

ans =

2.0000 - 3.0000i

5.0000 - 0.0000i

2.0000 + 1.0000i

****

****

**3- Produit de polynômes**

* 1er exemple

( x –2 )( x – 1 ) = ?

**>> p1=[ 1 -2 ]**

p1 =

1 -2

**>> p2=[ 1 -1 ]**

p2 =

1 -1

**>> conv( p1 , p2 )**

ans =

1 -3 2

Autrement dit :

( x –2 )( x – 1 ) = x² -3x +2

* 2ème exemple

(3x² - 5x + 2)( x² + 3x + 8) = ?

**>> p1=[ 3 -5 2 ]**

p1 =

3 -5 2

**>> p2=[ 1 3 8 ]**

p2 =

1 3 8

**>> conv( p1 , p2 )**

ans =

3 4 11 -34 16

Autrement écrit :

(3x² - 5x + 2)( x² + 3x + 8) = 3x4 + 4x3 + 11 x² -34 x +16

****

****

**4- Décomposition en éléments simples**



p1 , p2 …désignent les " pôles ".

* Exemple :



polynôme du numérateur :

**>> n =[ 6 ]**

n =

6

polynôme du dénominateur :

**>> d =[ 1 6 11 6 0 ]**

d =

1 6 11 6 0

**>> [ r , p , k ] = residue ( n , d)**

r =

-1.0000

3.0000

-3.0000

1.0000

p =

-3.0000

-2.0000

-1.0000

0

k =

[ ]



****

****

**5- Représentation graphique**

* Exemple :

y = f(x) = 3x² - 5x + 2

**5-1- Utilisation de la fonction** [**plot**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_plot.htm)

**>> p = [ 3 -5 2 ]**

p =

3 -5 2

Calcul de f( x = 1) :

**>> polyval( p , 1 )**

ans =

0

Calcul de f( x = 2) :

**>> polyval( p , 2)**

ans =

4

Création du vecteur x :

**>> x = 0 : 0.01 : 2**

x =

Columns 1 through 7

0 0.0100 0.0200 0.0300 0.0400 0.0500 0.0600

…

Columns 197 through 201

1.9600 1.9700 1.9800 1.9900 2.0000

Création du vecteur y :

**>> y = polyval( p , x)**

y =

Columns 1 through 7

2.0000 1.9503 1.9012 1.8527 1.8048 1.7575 1.7108

…

Columns 197 through 201

3.7248 3.7927 3.8612 3.9303 4.0000

**>> plot (x , y)**

**>> grid on**

****

**5-2- Utilisation de la fonction** [**fplot**](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fplot.htm)

**5-2-1- Première méthode**

**>> fplot ( '3\*x^2 - 5\*x + 2' , [ 0 2 ] )**

**>> grid on**

****

**5-2-2- Deuxième méthode**

Il faut [créer le fichier .m de la fonction](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fichier_dune_fonction.htm) :

****

**>> f7(2)**

ans =

4

**>> f7(0)**

ans =

2

**>> fplot ('f7', [ 0 2] )**

**>> grid on**



* Remarque

On peut également définir le polynôme de la manière suivante :



© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



Matlab – minimum d’une fonction f(x)



* Fonction :

Fmin (version 4.2.)

Fminbnd (version 6.5.)

* Exemple :



Il faut commencer par [créer le fichier .m associé à cette fonction](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fichier_dune_fonction.htm) :



Dessinons maintenant le graphe de cette fonction :

**>> fplot( 'f1' , [ -10 10 ] )**

**>> grid on**



Cherchons le minimum de cette fonction dans l’intervalle 0 à 5 :

**>> fmin ( 'f1' , 0 , 5 )**

ans =

2.1177

**>> f1(ans)**

ans =

0.0903

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



**Matlab - Création du fichier .m d’une fonction**



* Fonction :

function

* Exemple

Soit la fonction :



a) Commencez pour ouvrir un éditeur de texte :

Dans la fenêtre de commande de Matlab :

File -> New -> M-file

Avec la version 4.2. l'éditeur de texte par défaut est l'application "Bloc-notes".

Avec la version 6.5. l'éditeur de texte par défaut est l'application "M-File Editor".

b) Donnez un nom à cette fonction (dans cet exemple *fonc)* et saisissez son expression mathématique :



**Attention :** il faut mettre un [point](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_scalaire_vecteur_matrice.htm) devant les opérateurs multiplication, division et puissance :

**.\***

**./**

**.^**

c) Sauvegardez le fichier dans votre répertoire de travail (par exemple c:\USERS)

Nom : fonc

Extension : .m

d) Ajoutez le chemin du répertoire où se trouve votre fichier fonc.m

Avec la version 4.2. :

**>> path(path,'c:\USERS')**

Avec la version 6.5. :

File -> Set Path -> Add Folder

-> Save -> Close

Remarque : Vous pouvez utiliser n'importe quel nom pour les variables.

La fichier suivant donne le même résultat :



* Evaluation d'une fonction

Calcul de y ( x = 0 ) :

**>> fonc(0)**

ans =

2

Calcul de y ( x = 5 ) :

**>> fonc(5)**

ans =

10.2010

**>> fonc(-1)**

Warning: Divide by zero

ans =

Inf

Avec en argument un [vecteur](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_scalaire_vecteur_matrice.htm), la fonction retourne un vecteur :

**>> fonc( [0 1 2 3 4 5] )**

ans =

2.0000 3.8415 3.9099 6.9121 8.1121 10.2010

**>> x = 0 : 5**

x =

0 1 2 3 4 5

**>> y = fonc(x)**

y =

2.0000 3.8415 3.9099 6.9121 8.1121 10.2010

Avec en argument une [matrice](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_scalaire_vecteur_matrice.htm), la fonction retourne une matrice :

**>> fonc( [ 1 2 3 ; 4 5 6] )**

ans =

3.8415 3.9099 6.9121

8.1121 10.2010 11.2939

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



Matlab - Remarques à propos des scalaires, vecteurs, matrices et tableaux

Où l'on verra également la différence entre :

|  |  |
| --- | --- |
| **\*** | **.\*** |
| **/** | **./** |
| **^** | **.^** |



* 1- Matlab manipule des scalaires, vecteurs et matrices.

**>> a = 6**

a =

6

**>> b = [ 1 3 8 ]**

b =

1 3 8

**>> c = [ 2 10 ; 3 8]**

c =

2 10

3 8

**>> d = [ 1 2 ; 3 4 ]**

d =

1 2

3 4

**>> e = [ -2 -8 –3 ]**

e =

-2 -8 -3

**>> f = [ 8 ; 5 ; 2 ]**

f =

8

5

2

**>> whos**

Name Size

a 1 by 1

b 1 by 3

c 2 by 2

d 2 by 2

e 1 by 3

f 3 by 1

a est un scalaire (autrement dit une matrice 1x1)

b est un vecteur ligne (autrement dit une matrice 1x3)

c est une matrice (2x2)

d est une matrice (2x2)

e est un vecteur ligne (autrement dit une matrice 1x3)

f est un vecteur colonne (autrement dit une matrice 3x1)

* 2- Tableau (Array) et matrice

|  |  |
| --- | --- |
| Multiplication de matrice | **\*** |
| Multiplication de tableau (élément par élément) | **.\*** |
| Puissance de matrice | **^** |
| Puissance de tableau (élément par élément) | **.^** |
| Division de matriceB / A est équivalent à B \* inv(A)  | **/** |
| Division de tableau (élément par élément) | **. /** |

**\*** est le produit matriciel :

**>> [ 2 10 ; 3 8] \*[ 1 2 ; 3 4 ]**

ans =

32 44

27 38

**.\*** désigne la multiplication élément par élément, ce qui est complètement différent :

**>> [ 2 10 ; 3 8] .\* [ 1 2 ; 3 4 ]**

ans =

2 20

9 32

**>> [ 2 10 ; 3 8] / [ 1 2 ; 3 4 ]**

ans =

11 -3

6 -1

**>> [ 2 10 ; 3 8] ./ [ 1 2 ; 3 4 ]**

ans =

2 5

1 2

>> **[ 2 10 ; 3 8] ^2**

ans =

34 100

30 94

>> **[ 2 10 ; 3 8] .^2**

ans =

4 100

9 64

**>> 6\*[ 1 3 8 ]**

ans =

6 18 48

**>> 6.\*[ 1 3 8 ]**

ans =

6 18 48

**>> 6\*[ 2 10 ; 3 8]**

ans =

12 60

18 48

**>> 6.\*[ 2 10 ; 3 8]**

ans =

12 60

18 48

**>> [ 1 3 8 ].\*[ -2 -8 –3 ]**

ans =

-2 -24 -24

Le calcul suivant n'a pas de sens :

**>> [ 1 3 8 ] \*[ -2 -8 –3 ]**

??? Error using ==> \*

Inner matrix dimensions must agree.

**>> [ 1 3 8 ] .^2**

ans =

1 9 64

Le calcul suivant n'a pas de sens :

**>> [ 1 3 8 ] ^2**

??? Error using ==> ^

Matrix must be square.

Soit un vecteur ligne 1x7 :

**>> x = 0 : 6**

x =

0 1 2 3 4 5 6

**>> y = (x+1).^2**

y =

1 4 9 16 25 36 49

y est un vecteur ligne 1x7.

**>> y = (x+1)^2**

??? Error using ==> ^

Matrix must be square.

Testons avec une matrice carrée 2x2 :

**>> x = [ 1 2 ; 3 4 ]**

x =

1 2

3 4

**>> y = (x+1).^2**

y =

4 9

16 25

**>> y = (x+1)^2**

y =

16 21

28 37

**>> t = 0 : 5**

t =

0 1 2 3 4 5

**>> u = t.^2 + 3.\*t./ (5 + 2.\*t)**

u =

0 1.4286 4.6667 9.8182 16.9231 26.0000

**>> u = t^2 + 3\*t /(5 + 2\*t)**

??? Error using ==> ^

Matrix must be square.

**>> z = 2 : 2 : 12**

z =

2 4 6 8 10 12

**>> t./z**

ans =

0 0.2500 0.3333 0.3750 0.4000 0.4167

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



Matlab - Remarques à propos des scalaires, vecteurs, matrices et tableaux

Où l'on verra également la différence entre :

|  |  |
| --- | --- |
| **\*** | **.\*** |
| **/** | **./** |
| **^** | **.^** |



* 1- Matlab manipule des scalaires, vecteurs et matrices.

**>> a = 6**

a =

6

**>> b = [ 1 3 8 ]**

b =

1 3 8

**>> c = [ 2 10 ; 3 8]**

c =

2 10

3 8

**>> d = [ 1 2 ; 3 4 ]**

d =

1 2

3 4

**>> e = [ -2 -8 –3 ]**

e =

-2 -8 -3

**>> f = [ 8 ; 5 ; 2 ]**

f =

8

5

2

**>> whos**

Name Size

a 1 by 1

b 1 by 3

c 2 by 2

d 2 by 2

e 1 by 3

f 3 by 1

a est un scalaire (autrement dit une matrice 1x1)

b est un vecteur ligne (autrement dit une matrice 1x3)

c est une matrice (2x2)

d est une matrice (2x2)

e est un vecteur ligne (autrement dit une matrice 1x3)

f est un vecteur colonne (autrement dit une matrice 3x1)

* 2- Tableau (Array) et matrice

|  |  |
| --- | --- |
| Multiplication de matrice | **\*** |
| Multiplication de tableau (élément par élément) | **.\*** |
| Puissance de matrice | **^** |
| Puissance de tableau (élément par élément) | **.^** |
| Division de matriceB / A est équivalent à B \* inv(A)  | **/** |
| Division de tableau (élément par élément) | **. /** |

**\*** est le produit matriciel :

**>> [ 2 10 ; 3 8] \*[ 1 2 ; 3 4 ]**

ans =

32 44

27 38

**.\*** désigne la multiplication élément par élément, ce qui est complètement différent :

**>> [ 2 10 ; 3 8] .\* [ 1 2 ; 3 4 ]**

ans =

2 20

9 32

**>> [ 2 10 ; 3 8] / [ 1 2 ; 3 4 ]**

ans =

11 -3

6 -1

**>> [ 2 10 ; 3 8] ./ [ 1 2 ; 3 4 ]**

ans =

2 5

1 2

>> **[ 2 10 ; 3 8] ^2**

ans =

34 100

30 94

>> **[ 2 10 ; 3 8] .^2**

ans =

4 100

9 64

**>> 6\*[ 1 3 8 ]**

ans =

6 18 48

**>> 6.\*[ 1 3 8 ]**

ans =

6 18 48

**>> 6\*[ 2 10 ; 3 8]**

ans =

12 60

18 48

**>> 6.\*[ 2 10 ; 3 8]**

ans =

12 60

18 48

**>> [ 1 3 8 ].\*[ -2 -8 –3 ]**

ans =

-2 -24 -24

Le calcul suivant n'a pas de sens :

**>> [ 1 3 8 ] \*[ -2 -8 –3 ]**

??? Error using ==> \*

Inner matrix dimensions must agree.

**>> [ 1 3 8 ] .^2**

ans =

1 9 64

Le calcul suivant n'a pas de sens :

**>> [ 1 3 8 ] ^2**

??? Error using ==> ^

Matrix must be square.

Soit un vecteur ligne 1x7 :

**>> x = 0 : 6**

x =

0 1 2 3 4 5 6

**>> y = (x+1).^2**

y =

1 4 9 16 25 36 49

y est un vecteur ligne 1x7.

**>> y = (x+1)^2**

??? Error using ==> ^

Matrix must be square.

Testons avec une matrice carrée 2x2 :

**>> x = [ 1 2 ; 3 4 ]**

x =

1 2

3 4

**>> y = (x+1).^2**

y =

4 9

16 25

**>> y = (x+1)^2**

y =

16 21

28 37

**>> t = 0 : 5**

t =

0 1 2 3 4 5

**>> u = t.^2 + 3.\*t./ (5 + 2.\*t)**

u =

0 1.4286 4.6667 9.8182 16.9231 26.0000

**>> u = t^2 + 3\*t /(5 + 2\*t)**

??? Error using ==> ^

Matrix must be square.

**>> z = 2 : 2 : 12**

z =

2 4 6 8 10 12

**>> t./z**

ans =

0 0.2500 0.3333 0.3750 0.4000 0.4167

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



Matlab - Remarques à propos des scalaires, vecteurs, matrices et tableaux

Où l'on verra également la différence entre :

|  |  |
| --- | --- |
| **\*** | **.\*** |
| **/** | **./** |
| **^** | **.^** |



* 1- Matlab manipule des scalaires, vecteurs et matrices.

**>> a = 6**

a =

6

**>> b = [ 1 3 8 ]**

b =

1 3 8

**>> c = [ 2 10 ; 3 8]**

c =

2 10

3 8

**>> d = [ 1 2 ; 3 4 ]**

d =

1 2

3 4

**>> e = [ -2 -8 –3 ]**

e =

-2 -8 -3

**>> f = [ 8 ; 5 ; 2 ]**

f =

8

5

2

**>> whos**

Name Size

a 1 by 1

b 1 by 3

c 2 by 2

d 2 by 2

e 1 by 3

f 3 by 1

a est un scalaire (autrement dit une matrice 1x1)

b est un vecteur ligne (autrement dit une matrice 1x3)

c est une matrice (2x2)

d est une matrice (2x2)

e est un vecteur ligne (autrement dit une matrice 1x3)

f est un vecteur colonne (autrement dit une matrice 3x1)

* 2- Tableau (Array) et matrice

|  |  |
| --- | --- |
| Multiplication de matrice | **\*** |
| Multiplication de tableau (élément par élément) | **.\*** |
| Puissance de matrice | **^** |
| Puissance de tableau (élément par élément) | **.^** |
| Division de matriceB / A est équivalent à B \* inv(A)  | **/** |
| Division de tableau (élément par élément) | **. /** |

**\*** est le produit matriciel :

**>> [ 2 10 ; 3 8] \*[ 1 2 ; 3 4 ]**

ans =

32 44

27 38

**.\*** désigne la multiplication élément par élément, ce qui est complètement différent :

**>> [ 2 10 ; 3 8] .\* [ 1 2 ; 3 4 ]**

ans =

2 20

9 32

**>> [ 2 10 ; 3 8] / [ 1 2 ; 3 4 ]**

ans =

11 -3

6 -1

**>> [ 2 10 ; 3 8] ./ [ 1 2 ; 3 4 ]**

ans =

2 5

1 2

>> **[ 2 10 ; 3 8] ^2**

ans =

34 100

30 94

>> **[ 2 10 ; 3 8] .^2**

ans =

4 100

9 64

**>> 6\*[ 1 3 8 ]**

ans =

6 18 48

**>> 6.\*[ 1 3 8 ]**

ans =

6 18 48

**>> 6\*[ 2 10 ; 3 8]**

ans =

12 60

18 48

**>> 6.\*[ 2 10 ; 3 8]**

ans =

12 60

18 48

**>> [ 1 3 8 ].\*[ -2 -8 –3 ]**

ans =

-2 -24 -24

Le calcul suivant n'a pas de sens :

**>> [ 1 3 8 ] \*[ -2 -8 –3 ]**

??? Error using ==> \*

Inner matrix dimensions must agree.

**>> [ 1 3 8 ] .^2**

ans =

1 9 64

Le calcul suivant n'a pas de sens :

**>> [ 1 3 8 ] ^2**

??? Error using ==> ^

Matrix must be square.

Soit un vecteur ligne 1x7 :

**>> x = 0 : 6**

x =

0 1 2 3 4 5 6

**>> y = (x+1).^2**

y =

1 4 9 16 25 36 49

y est un vecteur ligne 1x7.

**>> y = (x+1)^2**

??? Error using ==> ^

Matrix must be square.

Testons avec une matrice carrée 2x2 :

**>> x = [ 1 2 ; 3 4 ]**

x =

1 2

3 4

**>> y = (x+1).^2**

y =

4 9

16 25

**>> y = (x+1)^2**

y =

16 21

28 37

**>> t = 0 : 5**

t =

0 1 2 3 4 5

**>> u = t.^2 + 3.\*t./ (5 + 2.\*t)**

u =

0 1.4286 4.6667 9.8182 16.9231 26.0000

**>> u = t^2 + 3\*t /(5 + 2\*t)**

??? Error using ==> ^

Matrix must be square.

**>> z = 2 : 2 : 12**

z =

2 4 6 8 10 12

**>> t./z**

ans =

0 0.2500 0.3333 0.3750 0.4000 0.4167

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



Matlab - Equation non linéaire à une inconnue & Racines d’une fonction



* Fonction :

fzero

* Exemple :

Soit à résoudre l’équation :



Cela revient à trouver les racines de la fonction :



Il faut commencer par [créer le fichier .m de cette fonction](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fichier_dune_fonction.htm) :



Représentation graphique de la fonction :

**>> fplot('f3', [ 0 5 ])**

**>> grid on**



Cela fait apparaître 3 racines, aux environs de 2,7 3,1 et 3,4.

**>> fzero('f3', 2.7)**

ans =

2.6545

Pour les maniaques de la précision :

**>> format long e**

**>> fzero('f3', 2.7)**

ans =

2.654461193307640e+000

**>> format short**

**>> fzero('f3', 3.1)**

ans =

3.1039

**>> fzero('f3', 3.4)**

ans =

3.4078

Vérification :

**>> f3(ans)**

ans =

5.5511e-016

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>

**Matlab - Création du fichier .m d’une fonction**



* Fonction :

function

* Exemple

Soit la fonction :



a) Commencez pour ouvrir un éditeur de texte :

Dans la fenêtre de commande de Matlab :

File -> New -> M-file

Avec la version 4.2. l'éditeur de texte par défaut est l'application "Bloc-notes".

Avec la version 6.5. l'éditeur de texte par défaut est l'application "M-File Editor".

b) Donnez un nom à cette fonction (dans cet exemple *fonc)* et saisissez son expression mathématique :



**Attention :** il faut mettre un [point](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_scalaire_vecteur_matrice.htm) devant les opérateurs multiplication, division et puissance :

**.\***

**./**

**.^**

c) Sauvegardez le fichier dans votre répertoire de travail (par exemple c:\USERS)

Nom : fonc

Extension : .m

d) Ajoutez le chemin du répertoire où se trouve votre fichier fonc.m

Avec la version 4.2. :

**>> path(path,'c:\USERS')**

Avec la version 6.5. :

File -> Set Path -> Add Folder

-> Save -> Close

Remarque : Vous pouvez utiliser n'importe quel nom pour les variables.

La fichier suivant donne le même résultat :



* Evaluation d'une fonction

Calcul de y ( x = 0 ) :

**>> fonc(0)**

ans =

2

Calcul de y ( x = 5 ) :

**>> fonc(5)**

ans =

10.2010

**>> fonc(-1)**

Warning: Divide by zero

ans =

Inf

Avec en argument un [vecteur](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_scalaire_vecteur_matrice.htm), la fonction retourne un vecteur :

**>> fonc( [0 1 2 3 4 5] )**

ans =

2.0000 3.8415 3.9099 6.9121 8.1121 10.2010

**>> x = 0 : 5**

x =

0 1 2 3 4 5

**>> y = fonc(x)**

y =

2.0000 3.8415 3.9099 6.9121 8.1121 10.2010

Avec en argument une [matrice](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_scalaire_vecteur_matrice.htm), la fonction retourne une matrice :

**>> fonc( [ 1 2 3 ; 4 5 6] )**

ans =

3.8415 3.9099 6.9121

8.1121 10.2010 11.2939

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



### http://pagesperso-orange.fr/fabrice.sincere/matlab/logo_matlab.gifMatlab – Graphe de la dérivée d’une fonction f ’(x)

### http://pagesperso-orange.fr/fabrice.sincere/matlab/prev1.gif

Voici "ma" méthode :

* Prenons l'exemple suivant :



Commençons par [créer le fichier .m de la fonction](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fichier_dune_fonction.htm) :

****

**>> [ x y ] = fplot ( 'f2' , [ 0 2\*pi ] )**

x =

0

0.0126

0.0251

…

6.2455

6.2643

6.2832

y =

0.5000

0.4997

0.4986

…

0.4966

0.4992

0.5000

Une approximation de la dérivée est donnée par :

**>> dy = diff(y)./diff(x)**

dy =

-0.0276

-0.0804

-0.1292

…

0.1370

0.0438

**>> plot (x , dy)**

??? Error using ==> plot

Vectors must be the same lengths.

**>> length(x)**

ans =

307

**>> length(dy)**

ans =

306

Problème !

Le vecteur dy a un élément de moins que le vecteur x, ce que n'apprécie pas la fonction plot.

Nous allons contourner la difficulté en ignorant le dernier élément du vecteur x :

**>> plot ( x (1 : length(x)-1 , : ) , dy , '. ')**

**>> grid on**



Comparons avec la dérivée exacte (couleur bleue) :

**>> hold on**

**>> fplot ( 'cos(x) - 1.5\*sin(3\*x) - cos(5\*x)' , [ 0 2\*pi ] , 'b.' )**



En vert, la fonction f2(x) :

**>> fplot ( 'f2' , [ 0 2\*pi ] , 'g.' )**



© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>

****

**Matlab - Calcul d'intégrale**



* Fonctions :

quad : algorithme de Simpson

quad8 : algorithme de Newton-Cote (version 4.2.)

quadl (version 6.5.)

* Exemple :



Il faut commencer par [créer le fichier .m de la fonction](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fichier_dune_fonction.htm) à intégrer :



**>> fplot( 'f4', [ 0 1 ])**

**>> grid on**



**>> quad( 'f4' , 0 , 1 )**

ans =

3.1416

**>> format long e**

**>> quad( 'f4' , 0 , 1 )**

ans =

3.141592502458707e+000

La valeur exacte de cette intégrale est pi.

**>> pi**

ans =

3.141592653589793e+000

Il y a un petit écart ...

Améliorons la précision avec une tolérance de calcul de 1e-10 (la tolérance par défaut est 1e-3) :

**>> quad( 'f4' , 0 , 1 , 1e-10 )**

ans =

3.141592653589792e+000

... ce qui est beaucoup mieux !

**>> quad8( 'f4' , 0 , 1 , 1e-10)**

ans =

3.141592653589793e+000

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



Matlab - Résolution d’équations différentielles





* Fonctions (Matlab version 4.2.)

ode23 : algorithme de Runge-Kutta du 2ème et 3ème ordres

ode45 : algorithme de Runge-Kutta du 4ème et 5ème ordres

* **Exemple 1 : équation différentielle du premier ordre**

Soit une fonction y1(t) soumise à l’équation différentielle :



Créons le fichier f10.m :



**>> [ t , y ] = ode23 ( 'f10' , 0 , 20 , [ 5 ] )**

t =

0

0.1562

0.4833

...

18.0327

19.2827

20.0000

y =

5.0000

5.3758

6.0735

...

9.9995

9.9997

9.9998

**>> plot ( t , y )**

**>> grid on**



Pour avoir une tolérance sur les calculs de 1e-7 :

**>> [ t , y ] = ode23 ('f10' , 0 , 20 , [ 5 ] , 1e-7)**

A tester :

**>> [t , y ] = ode45 ( 'f10' , 0 , 20 , [ 5 ] )**





* **Exemple 2 : équation différentielle du deuxième ordre**



Créons le fichier f11.m :



**>> [ t , y ]=ode23 ( 'f11' , 0 , 10 , [ 0 2 ] )**

t =

0

0.0781

0.1858

...

9.7555

9.9671

10.0000

y =

0 2.0000

-0.3058 1.9880

-0.7017 1.9334

...

-0.2114 0.1188

-0.2284 0.0719

-0.2291 0.0644

**>> plot ( t , y )**



Pour n'afficher que la première colonne, c'est-à-dire y1(t) ou dy2(t)/dt :

**>> plot ( t , y ( : , 1 ) )**



Pour n'afficher que la deuxième colonne, c'est-à-dire y2(t) :

**>> plot ( t , y ( : , 2 ) )**







* **Exemple 3 : équation différentielle du troisième ordre**



Créons le fichier f12.m :



**>> [ t , y ] = ode23 ( 'f12' , 0 , 2 , [ 2 1 1 ] )**

t =

0

0.0156

0.1290

...

1.8692

1.9942

2.0000

y =

2.0000 1.0000 1.0000

2.0938 1.0320 1.0159

2.7741 1.3080 1.1478

...

13.2149 15.2195 12.8932

13.9649 16.9183 14.9009

14.0000 17.0000 15.0000

**>> plot ( t , y )**



Pour n'afficher que la troisième colonne, c'est-à-dire y3(t) :

**>> plot ( t , y ( : , 3 ) )**



Nous allons comparer avec la solution exacte : y3(t) = t^3 + t^2 + t + 1

**>> hold on**

**>> fplot ( 'x^3 + x^2 + x + 1 ' , [ 0 2 ] , 'b--' )**



Les deux courbes sont superposées.





* **Exemple 4 : Equation non linéaire du 2ème ordre**

Equation de Van der Pol



Créons le fichier f9.m :



**>> [ t , y ]=ode23 ( 'f9' , 0 , 20 , [ 0 0.25 ] )**

**>> plot ( t , y )**



© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>



**Matlab **

**Diagramme de Bode**

* Fonctions :

Log10 : logarithme décimal

Semilogx : identique à la fonction [***plot***](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_plot.htm) mais avec une échelle logarithmique pour l’axe des abscisses

Loglog : identique à la fonction [***plot***](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_plot.htm) mais avec une échelle logarithmique pour l’axe des abscisses et des ordonnées

* **Premier exemple**

Considérons un filtre passe-bas du 2ème ordre, de fonction de transfert :



avec : K = 10, f1 = 1 kHz et f2 = 10 kHz.

Il faut commencer par [***créer le fichier t4.m de la fonction***](http://pagesperso-orange.fr/fabrice.sincere/matlab/matlab_fichier_dune_fonction.htm) de transfert :



Valeur de la fonction de transfert pour f = 0 Hz (régime continu) :

**>> t4(0)**

ans =

10

Valeur de la fonction de transfert pour f = 100 Hz :

**>> t4(100)**

ans =

9.8901 - 1.0890i

Le module donne l'amplification :

**>> abs(t4(100))**

ans =

9.9499

L'argument donne le déphasage (en radians) :

**>> angle(t4(100))**

ans =

-0.1097

Etudions la fonction de transfert sur la gamme de fréquence 100 Hz à 100 kHz :

**>> log10(100)**

ans =

2

**>> log10(100000)**

ans =

5

Nous allons créer un vecteur logf :

**>> logf = 2 : 0.01 : 5**

**>> f = 10.^logf**

f est un vecteur qui nous donne 300 points uniformément répartis de 100 Hz à 100 kHz (compte tenu de l’échelle logarithmique).

G est un vecteur qui nous retourne le gain (en dB) pour les fréquences précédentes :

**>> G = 20.\*log10 (abs (t4(f)) )**

**>> semilogx ( f , G )**

**>> ylabel ( 'gain (dB)' )**

**>> xlabel ( 'fréquence (Hz)' )**

**>> title ( 'diagramme de Bode du gain' )**

**>> grid on**



Déphasage en degrés :

**>> dephasage = angle(t4(f))\*180/pi**

**>> semilogx ( f , dephasage )**

**>> grid on**



* **Deuxième exemple** : système du deuxième ordre

Fichier de définition de la fonction t2 :



**>> logpulsation = 3 : 0.01 : 6**

**>> pulsation = 10.^logpulsation**

**>> G = 20.\*log10(abs(t2(pulsation)))**

**>>** **semilogx ( pulsation , G )**

**>> grid on**



**>> dephasage = angle(t2(pulsation))\*180/pi**

**>> semilogx ( pulsation , dephasage )**

**>> grid on**



**>> amplification = abs(t2(pulsation))**

>> **loglog ( pulsation , amplification)**

**>> grid on**

**>> xlabel ( 'pulsation(rad/s)' )**

**>> ylabel ( 'amplification' )**

**>> title ( 'Diagramme de Bode' )**

****

[***Compléments***](http://pagesperso-orange.fr/fabrice.sincere/matlab/programmation/matlab_programmation.htm)

**© Fabrice Sincère**

Matlab - A propos du calcul numérique et du calcul littéral



Matlab effectue du calcul numérique à partir d’algorithmes plus ou moins sophistiqués.

* Prenons l’exemple d’un calcul d’intégrale :



Les outils mathématiques permettent d’obtenir la valeur exacte de cette intégrale :



Matlab procède tout autrement : il calcule une valeur numérique approchée de cette intégrale.

Ainsi, la fonction *quad* utilise un algorithme de Simpson :

**>> quad('cos' , 0 , pi/4 , 1e-12)**

ans =

7.071067811865476e-001

A comparer avec la valeur exacte (ou plutôt sa valeur approchée à 1e-16 près) :

**>> sqrt(2)/2**

ans =

7.071067811865476e-001

(La fonction *sqrt* évalue la racine carrée d’un nombre avec un algorithme spécifique, comme vous l’aurez certainement compris.

Notez qu’une calculatrice de poche utilise la même technique).

* Alors pourquoi s’embêter à évaluer une intégrale alors que l’on peut obtenir sa valeur exacte ?

Oui effectivement : mais le calcul littéral n’est possible que dans le cas d’expressions simples, ce qui est rare quand on cherche à mettre en équations le monde qui nous entoure.

Les techniciens et ingénieurs ne s’ennuient plus depuis bien longtemps à faire du calcul littéral.

Avec un ordinateur, on sait très bien faire du calcul numérique, de la simulation, de le C.A.O. ….

* Il existe cependant des logiciels de calcul littéral (on parle aussi de calcul symbolique ou de calcul formel) : [Maxima](http://maxima.sourceforge.net/) (logiciel libre), [Maple](http://www.maplesoft.com/), [Mathematica](http://www.wolfram.com/) …

© Fabrice Sincère

<http://perso.orange.fr/fabrice.sincere/>

