Notion de base MATLAB

• Varible spéciales et constantes

>> pi : 3.14159265358979,
>> eps : 2.22044 e-016
>> 1/0 —> Warning : Divide by zero —> Inf
>> 0/0 —> " : " " —> NaN

Operation arithmétiques

>> 5+7-3+2*5
ans = .
19

>> 4/3 —> 1.3333 ← $\frac{4}{3}$
>> 4\3 —> 0.7500 ← $\frac{3}{4}$

>> 4*(-5)+12 —> -8
>> 2.3*(4-6)/(3+15) —> -0.2556

>> 3^4 —> 81
5,3 ^ (-1,5) —> 0.082

9^1/2 = 4.5 —> $\frac{9}{2}$
9^(1/2) = 3 —> $\sqrt{9}$

Format des nombres et précision des calculs

Matlab a la possibilité d'afficher les valeurs des variables dans differentes formats :
flottants courts (short) short e | format short
flottants longs (long) long e | format long.

Vecteurs ou tableaux à 1 dimension

x = [6 4 7]

x = [6, 4, 7]

size(x)

ans = 1 3

[m n] = size(x)

m = 1

n = 3

longeur_x = length(x) → longeur_x = 3

```
>> v = [x 1 2 8]
   v =
        6  4  7  1  2  8
>> w = [1 v]
>> w = [1 x 1 2 8]
>> w(3)  ───> ans = 4
>> d = 0;  ─> % debut
>> f = 1;  ─> % fin
>> pas = 0.25;
>> t = d:pas:f  ─> t = 0 0.2500 0.5000 0.7500 1.0000

>> x = [0 4 3];
>> y = [2 5 7];
>> x - y   ans> -2 -1 -4
>> x + y   ans> 2 9 10
>> 3 + x   ─> 3 7 6
>> x - 2   ─> -2 2 1
>> 2 * x   ─> 0 8 6
>> x / 4   ─> 0 1.0000 0.7500
>> Ex = x'  ─>  0
                4
                3
```

>> x * tx —> 25

>> tx * y —> 0 0 0
8 20 28
6 15 21

>> x .* y => 0 20 21

>> x .^ 2 => 0 16 9

>> x ./ y => 0 0,8 0,4286

y .\ x => 0 0,8 0,4286

plusieurs fonctions opérant directement sur des vecteurs sont disponibles sous MATLAB.

sum : Somme des composantes d'un vecteur.
prod : produit " "
sqrt : racines carrées des composantes d'un vecteur
mean : moyenne des composantes d'un vecteur.

sum(x) —> 7
prod(y) —> 70
sqrt(x) —> 0 2 1,732
mean(x) —> 2,333

Matrice ou tableau à 2 dimensions

Le tableau à 2 dimension (matrice) est l'élément de base de MATLAB. Un vecteur n'est autre qu'une matrice à une ligne ou à une colonne.

x = [0 1 2 ; 3 4 5] —> 0 1 2
3 4 5

x = [0 1 2
3 4 5] —> 0 1 2
3 4 5

[m, n] = size(x)

m = 2
n = 3

x(2,1) ——> ans = 3

Matrice particulière.

% identité

$I = eye(3) \longrightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

% Matrice nulle.

n_lignes = 2 ou n = 2

n_colonnes = 3 ou m = 3 colonnes

mat0 = zeros(2,3) ou mat = zeros(n,m)

mat_1 = ones(3,2)

mat_rand = rand(3,2) % opérations élémentaires sur les matrices.

A = [1 2 3; 0 3 1; -2 -1 4]

det(A) => 27

inv(A) => 0,4815 -0,4074 -0,2593
-0,0741 0,3704 -0.0370
0,2222 -0.1111 0,1111

Les chaînes de caractères caractères

cH = 'Matlab' Matlab

[n, m] = size(CH)

n = 1
m = 6

on peut concaténer plusieurs chaînes de caractères imbriqué in série en les écrivant, comme des élément d'un vecteur

cch = ['langage' CH]

cch langage Matlab.

Les nombre complexe : i ou j

$z = a + ib$

$z = a + jb$

>> i^2

ans =

-1.0000 + 0.0000i

>> z1 = 4 - 3i

z1 = 4.0000 - 3.0000i

>> z1c = conj(z1) → 4.0000 + 3.000i % Conjugué de z1

>> z2 = 2 + 3i

>> z1 + z2   ans = 6

>> z1 * z2 → 17 + 6i

>> z1 .* z1c → 25

>> z1 / z2 → -0,0769 - 1,3846i

>> z2^3 => -46.0000 + 9.0000i

>> i = -3j

>> z = 4 + 2i → 4.000 + 2.000i

>> z = 4 + 2*i → z = -2

>> a = real(z1)

>> b = imag(z1)

>> r = sqrt(z1*z1c)

r = abs(z1) → 5

theta = angle(z1) → theta = -0,643r

Rappel

Forme algébrique $z = a + ib$

" trigonométrique : $z = [r, theta]$

r et theta sont respectivement le module et l'argument de z

z = r*(cos(theta) + i*sin(theta)) → ans = 4 - 3i

Il est possible d'utiliser la notation exponentielle.

r*exp(j*theta) $\xrightarrow{ans}$ 4.000 − 3.000i

Les polynômes

MATLAB représente un polynôme sous forme d'un tableau de ses coefficients classés dans l'ordre des puissances décroissantes.

$P(x) = x^2 - 6x + 9$

P = [1 -6 9]

$Q(x) = x^3 + 2x^2 - 3$ ⟶ Q=[1 2 0 -3]

Racines d'un polynômes

On peut déterminer les racines à l'aide de la fonction roots

roots(P) $\xrightarrow{ans=}$ 3
3

$P(x) = (x-3)^2$

roots(Q) ⟶ −1,5 + 0,866 i
−1,5 − 0,866 i
1.000

Détermination d'un polynôme à partir de ses racines

r = [1, 2, 3]

K = poly(r) ⟶ K = 1 -6 11 -6

$K(x) = x^3 - 6x^2 + 11x - 6$

pour évaluer un polynôme en un point, on utilise la fonction Polyval.

Exp : La valeur du polynome P en 1 et celle du polynôme Q en 0

>>polyval (P, 1) ans = 4

>> Polyval(Q, 0) —> -3

Représentation graphique.

Pour tracer la représentation graphique du polynôme K(x), définissons un domaine pour la variable x qui contient les racines de K.

Domaine des valeurs de la variables x et évaluation du polynôme K :

```
>> x = 0:0.1:4;
>> y = polyval(K, x); % y=polyval(K, 0:0.1:4)
```

Tracer de la fonction y = K(x)

```
>> plot(x, y)
>> grid
>> title('tracé de y = x^3-6x^2+11x-6')
>> xlabel('x')
>> ylabel('y')
```

x^3 6x^2

% plot : tracé d'une représentation graphique.

% grid : affiche une grille.

% title : attribue titre au graphique

% xlabel : attribue un texte à l'axe des abscisses,

% ylabel : attribue un texte à l'axe des ordonnées.

Multiplication et division de polynômes

La multiplication et la division de polynômes peuvent êtres réalisés facilement avec MATLAB.
Soit deux polynômes P_1 et P_2 définie par :

$P_1(x) = x + 2$

$P_2(x) = x^2 - 2x + 1$

$P_1 = [1 \quad 2] \Rightarrow P_1 = 1 \quad 2$

$P_2 = [1 \quad -2 \quad 1] \Rightarrow 1 \quad -2 \quad 1$

Le résultat de la multiplication de P_1 et P_2 est le polynome P_3 qui s'obtient avec la fonction conv

>> $P_3 = conv(P_1, P_2) \rightarrow P_3 = 1 \quad 0 \quad -3 \quad 2$

La division de deux polynômes se fait par la fonction deconv. Le quotient Q et le rest R de la division peuvent être obtenus sous forme d'éléments d'un tableau.

>> $[Q, R] = deconv(P_2, P_1) \Rightarrow Q = 1 \quad -4$

$R = 0 \quad 0 \quad 9$

En divison P_3 par P_1, on retrouve bien le polynôme P_2 (le rest est nul).

>> $[Q, R] = deconv(P_3, P_1)$

$Q = 1 \quad -2 \quad 1$

$R = 0 \quad 0 \quad 0 \quad 0$

Graphiques 2D et 3D

Matlab peut produire des graphiques couleurs 2D et 3D impressionnants. Il ~~fournit~~ aussi ~~les~~ outils et moyens de personnaliser et de modifier pratiquement tous leurs aspects, facilement et pratiquement de manière parfaitement contrôlée.

fournit aussi les outils

Graphique 2D

```
>> x=-pi:0.1:pi; % Définition de l'intervalle de x
                 % et calcul des valeurs de la
>> y=sin(x);       fonction y=f(x)

>> plot(x,y) % Tracé de la fonction.
% Documentation du graphique
>> grid
>> xlabel('x'), ylabel('y')
>> title('y=f(x)')
```

```
% On peut aussi tracer des courbes paramétriques
>> t=0:0.001:2*pi;
>> x=cos(3*t);
>> y=sin(2*t);
>> plot(x,y)
>> grid
>> xlabel('x')
>> ylabel('y')
>> Title('courbe de lissajous').
```

Graphiques 3D

Soit l'exemple suivant d'une fonction à 2 variables

$$z = \frac{\sin(x^2+y^2)}{x^2+y^2}$$

pour x et y variant de $-\pi$ à π avec un pas de $\pi/10$

```
>> x = -pi:pi/10:pi;
>> y = x;
```

Dans la prochaine étape on génère deux matrices carrées X et Y qui définissent le domaine de calcul de z, on utilisera pour ceci la fonction meshdom

```
>> [X, Y] = meshdom(x, y);
```

On évalue la fonction z et on stocke les données dans la variable Z1

```
>> Z1 = sin(X.^2+Y.^2)./(X.^2+Y.^2);
```

On dessine la surface représentative de la fonction

```
>> mesh(Z1)
```

Nous pouvons rajouter un titre pour le tracé (title), des légendes pour les axes (xlabel, ylabel et zlabel) ainsi qu'un quadrillage (grid).

Voir → surf, plot3 etc.

D'autre commandes (surf, plot3 ... etc.)

, save, load ... clear all, clc, whos

fichier de commandes (scripts)

Instruction de contrôle

MATLAB dispose des instructions de contrôle suivantes : for, while et if. La syntaxe de chacune de ces instruction est semblable à celles des langages classiques.

1- L'instruction for

```
for compteur = valDébut : pas : valFin
   Instruction
end
```

L'exemple suivant permet de générer, à l'aide de la boucle for, les carrés des n premiers entiers naturels

fichier ncarres.m

```
% tableau des carrés des n premiers entiers naturels

n=10;
x=[ ];
for i=1:n
    x =[x, i^2];
end
x
>> ncarres
x = 1   4   9   16   25   36   49   64   81   100
```

Sur la même ligne, les instructions sont séparées par des virgules ou des points virgules

```
x =[ ]; for i=1:n , x=[x , i^2]; end
```

2- L'instruction while

```
while condition
      instruction
end
```

Dans l'exemple suivant, on affiche le plus petit entier naturel n tel que 2^n est supérieur ou égal à un nombre donné x

fichier nx.m

```
x = 15; n = 0
while 2^n < x
      n = n+1
end
      n
>> n_x
   n = 4
```

sur une même ligne

```
x = 15; n = 0; while 2^n < x, n = n+1 ; end; n
```

3- L'instruction if

```
if condition
   instruction
end
```

```
if condition
   instruction
else if condition
   instruction
else instruction
end
```

```
if condition
   instruction (si les conditions sont vérifiées)
else
   instruction (si les conditions ne sont pas vérifiées)
end
```

- L'exemple suivant permet de vérifier si un entier naturel donné n est pair ou impair

ex:

```
if rem(n,2) == 0
   disp('nombre pair')
else
   disp('nombre impair')
end
```

rem : retourne le rest de la division de deux nombres.

disp : affiche le message spécifié sous forme d'une chaîne de caractères.

Operateurs relationnel et logiques (operateurs booléens)

Des expressions relationnelles et logiques peuvent être utilisées dans Matlab exactement comme dans les autres langages de programmation tels que Fortran ou le C.

Operateurs relationnels:

Les operateurs relationnels sont: <, <=, >=, ==, ~=

Operateurs logiques

Les expressions relationnelles peuvent être combinées en utilisant les operateurs logiques suivants:

&, |, ~ qui signifient respectivement:

"et" (and), "ou" (OR) et "non" (NOT).

Syntaxe du branchement switch

```
switch expression
    case 1   value 1,
             instructions
    case 2   value 2
             instructions
    |
    |
    otherwise
             instruction
end.
```

$(x(j), y(i), z(i,j))$

$(x(i), y(j), z(j,i))$

$x_i \downarrow N_1$ $\quad$ $y_j \rightarrow N_2$

$Z(i,j)$

(n_1, n_2)

x = - Pi : Pi/10 : Pi

y = - Pi : Pi/10 / Pi

[m1, n1] = size(x);

[m2, n2] = size(y);

$$\boxed{z = \frac{\sin(x^2+y^2)}{x^2+y^2}}$$

```
For i=1:n1
    For j=1:n2
        z(j,i) = sin(x.^2 + y.^2) / (x^2 + y^2)
        z(j,i) = sin(x(i).^2 + y(j).^2) / x(i).^2 + y(j).^2
    End
End
```

→ mesh(x, y, z)

ou.

```
For i=1:n1
    z(:,i) sin(x(i).^2, y(:).^2) / x(i).^2 + y(:).^2
End For
```

```
clc
clear all

A=[77,3,25;8,2,4;0,7,9]

[N,M]=size A;

For i=1:1:M
    For j=i+1,:1:M
        if A(1,j)<A(1,i)
            x=A(:,j);
            A=(:,j)=A(:,i);
            A(:,i)=x;
        end
    end
end

Disp(A)
```

inhibition