



Rappel du cours

« Matlab »

SMI4

ZERHARI Btissam

Notion de programme

Ordinateur effectue des opérations élémentaires :

- o addition
- o soustraction
- o affectation
- o comparaison

Programme :

- o orchestre déroulement opérations dans le temps
- o Le programme est un **objet statique**
- o lors de son exécution il **évolue** de façon **dynamique** en fonction de l'état des variables.

Les vecteurs en Matlab

- Pour définir un vecteur v , par exemple $v = (11 \ 22 \ 33 \ 44 \ 55)$, on utilise

`>> v=[11 22 33 44 55]` La commande 2

`>> v'` donne ensuite la transposée de ce vecteur, c'est-à-dire ici un vecteur colonne. On peut aussi écrire

`>> z=[11 ; 22 ; 33 ; 44 ; 55]` pour obtenir directement un vecteur colonne z , mais il est bien plus simple d'écrire

`>> z=v'`

- Si on écrit `[1 :1 :5]`, on définit un vecteur allant de 1 à 5 avec une incrémentation unité. Ainsi

`>> w=[1 :1 :5]` ou encore

`>> w=[1 :5]` nous permettent de définir le vecteur

$$w = (1 \ 2 \ 3 \ 4 \ 5)$$

Opération sur les vecteurs

■ Exemple :

```
>> A = [ 7 5 9]; B = [2 6 4];
```

```
>> C= A.*B; %effectue la multiplication élément par élément
```

```
>> disp(C) % le vecteur C contient [ 7*2 5*6 9*4]
```

C =

14 30 36

NB: Le même raisonnement est valable pour la division et la puissance.

<code>n:m</code>	nombres de n à m par pas de 1
<code>n:p:m</code>	nombres de n à m par pas de p
<code>linspace(n,m,p)</code>	p nombres de n à m
<code>length(x)</code>	longueur de x
<code>x(i)</code>	i-ème coordonnée de x
<code>x(i1:i2)</code>	coordonnées i1 à i2 de x
<code>x(i1:i2)=[]</code>	supprimer les coordonnées i1 à i2 de x
<code>[x,y]</code>	concaténer les vecteurs x et y
<code>x*y'</code>	produit scalaire des vecteurs lignes x et y
<code>x'*y</code>	produit scalaire des vecteurs colonnes x et y
<code>reshape(x,u,v)</code>	crée une matrice de taille [u,v], à partir de x

Opérations matricielles

- Matlab est performant pour la manipulation de matrices, de vecteurs, ou de tableaux en général.

Exemples :

```
>> A = [1 2 3 ; 4 5 6] ;  
>> d = ones(2,1) ;  
>> A'*d  
ans = 5 7 9
```

- Il existe aussi des fonctions spécifiques extrêmement utiles :
 1. `inv(A)` calcule l'inverse de A.
 2. `det(A)` calcule le déterminant de A.
 3. `diag(A)` retourne la diagonale.
 4. `[m,n]=size(A)` m nombre de ligne n nombre de colonne.

Opérations matricielles

- Suite des exemples :

```
>> B = [3 4 ; 0 1] ;
```

```
>> C = [2 1 ; 0 1] ;
```

```
>> C*B    multiplier les éléments de A par ceux de B,
```

```
ans =
```

```
6 9
```

```
0 1
```

```
>> C.*B    multiplication élément par élément des vecteurs
```

```
ans =
```

```
6 4
```

```
0 1
```

- De même pour la division :

```
>> C/B
```

```
>> C./B
```

Opérations matricielles

▪ Matrices :

<code>size(A)</code>	nombre de lignes et de colonnes de <code>A</code>
<code>A(i, j)</code>	coefficient d'ordre <code>i, j</code> de <code>A</code>
<code>A(i1:i2, :)</code>	lignes <code>i1</code> à <code>i2</code> de <code>A</code>
<code>A(i1:i2, :) = []</code>	supprimer les lignes <code>i1</code> à <code>i2</code> de <code>A</code>
<code>A(:, j1:j2)</code>	colonnes <code>j1</code> à <code>j2</code> de <code>A</code>
<code>A(:, j1:j2) = []</code>	supprimer les colonnes <code>j1</code> à <code>j2</code> de <code>A</code>
<code>A(:)</code>	indexation linéaire de <code>A</code> , (concaténation des vecteurs colonnes de <code>A</code>)
<code>A(i)</code>	coefficient d'ordre <code>i</code> dans l'indexation linéaire
<code>diag(A)</code>	coefficients diagonaux de <code>A</code>

▪ Matrices particulières :

<code>zeros(m, n)</code>	matrice nulle de taille <code>m, n</code>
<code>ones(m, n)</code>	matrice de taille <code>m, n</code> dont tous les coefficients valent 1
<code>eye(n)</code>	matrice identité de taille <code>n</code>
<code>diag(x)</code>	matrice diagonale dont la diagonale est le vecteur <code>x</code>
<code>magic(n)</code>	carré magique de taille <code>n</code>
<code>rand(m, n)</code>	matrice de taille <code>m, n</code> à coefficients i.i.d. de loi uniforme sur $[0, 1]$
<code>randn(m, n)</code>	matrice de taille <code>m, n</code> à coefficients i.i.d. de loi normale $\mathcal{N}(0, 1)$

Syntaxe Matlab pour répétition et choix

- o Nombre de répétitions fini

```
for v = expression  
    instructions  
end
```

- o Nombre de répétitions indéfini

```
while expression  
    instructions  
end
```

- o Choix simple

```
if expression  
    instructions  
end
```

- o Choix multiple

```
if expression1  
    instructions  
elseif expression2  
    instructions  
elseif expression3  
    instructions  
else  
    instructions  
end
```

Exemple

- Moyenne éléments d'un vecteur en utilisant la **boucle for** :

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + \cdots + x_n) \equiv \frac{1}{n} \sum_{i=1}^n x_i$$

```
s = 0;  
for i = 1:n  
    s = s + x(i);  
end  
xbar = s/n;
```

- Moyenne éléments d'un vecteur en utilisant la **boucle while**:

```
s = 0;  
i = 1;  
while i < n  
    s = s + x(i);  
    i = i+1;  
end  
xbar = s/n;
```

Exemple

- Choix simple :

```
t = 0;  
if t>0  
signe = +1  
else  
disp('nul ou neg')  
end
```

- Choix simple :

```
point = 8  
if point >= 9, apprec = 'Excellent'  
elseif point >= 8, apprec = 'Très bien'  
elseif point >= 7, apprec = 'bien'  
elseif point >= 6, apprec = 'assez bien'  
else apprec = 'moyen'  
end
```

Scripts et fonctions

Script

- ✓ Suite d'instructions
- ✓ Pas de paramètre d'entrée
- ✓ Ne renvoie aucune valeur
- ✓ Crée ou modifie des variables d'environnement
- ✓ Appels à d'autres scripts ou d'autres fonctions

Fonction

- ✓ peut prendre des arguments d'entrée
- ✓ retourne une ou plusieurs valeurs
- ✓ n'accède pas aux variables de l'environnement
- ✓ les variables locales inaccessibles depuis l'extérieur
- ✓ Contrainte syntaxique : seule la fonction portant le nom du M-fichier est accessible

Exemples de script

Fichier `exempleScript.m`

```
x = 1 ;  
y = 2 ;  
z = x + y ;
```

Dans la fenêtre de commandes :

`>> exempleScript`

→ Exécution du script stocké dans le fichier `exempleScript.m`

`>> x`

→ Renvoie les valeurs des variables x,y et z.

`>> y`

→ Les variables déclarées dans le script sont connues

`>> z`

Exemples de fonction

```
function [s,p] = SommeEtProduit(x,y) ← arguments d'entrée
    s = x + y ;
    p = x * y ;
(end function)                                     ← arguments de sortie
```

Dans la fenêtre de commandes :

```
>> a = 1 ;
>> b = 2 ;                                → Définition des variables a et b
```



```
>> [c,d] = SommeEtProduit(a,b) → Appel et exécution de la fonction
                                         SommeEtProduit (c = 3, d=2)
```



```
>> x                                     → Erreur, x n'est pas connue
```

Fonction anonyme

- Il est possible de définir une fonction sans créer un fichier qui lui est associé

nom de fonction = @(liste des arguments) expression.

- Cela peut se faire directement dans le terminal, dans un script ou même dans une fonction. Par exemple, on entre dans le terminal

```
>> cosXtanY = @(x,y) cos(x)*tan(y);  
>> cosXtanY(0.3,2.4)
```

```
ans =
```

```
-0.8751
```

Fonctions graphiques

Mot clé	Fonction
plot	Graphe en 2D avec une échelle linéaire
plotyy	Graphe avec deux axes y différents à gauche et à droite
loglog	Graphe en 2D avec une échelle logarithmique pour les deux axes
semilogx	Graphe en 2D avec une échelle logarithmique pour l'axe des x
semilogy	Graphe en 2D avec une échelle logarithmique pour l'axe des y
figure, close	Permet d'ouvrir ou de fermer une figure
subplot	Tracer plusieurs graphes alignés sur une même figure

Fonction	Usage
plot3	Tracé d'une ligne paramétrique en 3D
mesh	Tracé d'une surface en 3D, à partir de matrices de maillage
meshgrid	Définition de matrices de maillage à partir de deux vecteurs
surf	Tracé d'une surface en 3D avec dégradé de couleur, à partir de matrices de maillage
surf	Tracé d'une surface en 3D avec dégradé de couleur et lignes d'iso-valeurs
ezmesh ,ezmeshc	Tracé facile de surface (matrices de maillage définies par défaut)
ezsurf , ezsurf	Tracé facile de surface avec dégradé de couleur (matrices de maillage définie par défaut)
sphere	Définition de matrices de maillage pour le tracé d'une sphère
cylinder	Définition de matrices de maillage pour le tracé d'un cylindre

Fonctions graphiques

Options du graphe : titre, labels, axes

Mot clé	Fonction
title	Définir le titre du graphe
xlabel	Label de l'axe des x
ylabel	Label de l'axe des y
zlabel	Label de l'axe des z
legend	Ajouter une légende sur le graphe
text	Permet d'ajouter du texte sur le graphe
axis	Définir xmin, xmax, ymin et ymax du graphe. voir aussi axis equal

Style des lignes/couleur	Option dans plot
ligne continue rouge	-r
traits longs noirs	- -k
pointillés mauves	:m
traits longs + pointillés bleus	-.b
cercles verts	og
ligne continue + cercles jaunes	-yo
carrés bleu foncé	sb
croix bleu clair	xc
losanges + pointillés	.d
étoiles	h
Astérisques	*

Exemple

- `fplot` : trace point par point le graphe d'une fonction

Function $y=f1(x)$

```
1+ 2*x + sin(x*x);
```

```
>>fplot('f1',[1 2 3]) tq x=[1 2 3]
```

```
>>fplot ('sin(x) ', [ 0 10 ] ,'.') '.' c'est pour tracer la courbe en pointillé
```

```
>>fplot('sin(x) ', [ 0 10 ],'b') 'b' c'est la couleur bleu
```

Fonctions graphiques

Exemple :

- L'exemple ci-dessous illustre quelques unes des fonctions de bases pour la visualisation d'un graphique

```
>> f1 = @(x) (1./(sqrt(2*pi))).*exp(-(x.^2)./2) ;
>> f2 = @(x) (1./(2*sqrt(2*pi))).*exp(-(x.^2)./8);
>> f3 = @(x) (1./(0.5*sqrt(2*pi))).*exp(-(x.^2)./(2*0.5^2));
>> plot(x1, f1(x1), x1, f2(x1), x1, f3(x1)), xlabel('x'), ylabel('y'),
grid on, title('densité de la loi normale')
```

- Dans ce qui suit, un simple exemple qui illustre un graphique 3D sur MATLAB.

```
>> x = linspace(0,5,6) ;
>> y = linspace(0,5,6) ;
>> [X Y]=meshgrid(x,y) ;
>> Z = X.^2 .* Y.^2;
>> mesh(Z)
```