II. Les Matrices

1. Introduction

L'élément de base de Matlab est une matrice de dimension n x m composée de valeurs numériques (n = **li**gnes, m = **col**onnes) [1]. Une matrice de dimension 1 x 1 représente donc un scalaire. Une matrice n x 1 est un vecteur colonne de dimension n; de même pour un vecteur ligne, qui est une matrice 1 x m. Pour la machine cependant, il s'agit seulement d'une collection de nombres ou liste de caractères (strings) arrangés en lignes et colonnes. Cet arrangement a la forme d'un tableau (*array*) permettant de stocker de l'information et de manipuler des données [2]. Toute variable dans Matlab est une matrice qui est déclarée quand elle est initialisée. Une matrice doit toujours être régulière (même taille dans toutes ses dimensions). Il est aussi possible de créer des tableaux de données aux dimensions irrégulières (cells arrays : { }) et types différents (voir chapitre suivant).

2. Construction de matrices élémentaires

```
A = eye(n)              matrice identité n x m

A = zeros(n,m)          matrice de zéros de dimension n x m

A = ones(n,m)           matrice de uns de dimension n x m

A = diag(v)             matrice carrée avec les elements du
                        vecteur v dans la diagonale ou extraction
                        de la diagonale d'une matrice

A = rand(n,m)           matrice aléatoire de dimension n x m

A = meshgrid(x,y)       génère 2 grilles de points à partir des
                        vecteurs fournis → « coordonnées »

A = linspace(x1,xend,n) génère un vecteur entre x1 et xend
                        linéairement espacé de n intervalles.
```

Interrogation des dimensions d'une variable:

```
size(A)           dimension de la matrice A

length(X)         dimension d'un vecteur
```

3. Manipulations de matrices

Une fois qu'une variable existe, son type ainsi que sa taille peuvent être modifiés. Par ex. la taille d'une matrice peut être réduite ou augmentée, ou on peut changer sa forme (ses dimensions et l'arrangement de ses éléments). Ces modifications peuvent effectuées de différente manières.

Exemple : transformation d'éléments dans une matrice

```
% define a 3 x 3 matrix & add the value 5 in the 2nd row 1st col
>> E = zeros(3,3);
>> E(2,1) = 5;

E =
     0     0     0
     5     0     0
     0     0     0
```

Exemple : extraction d'éléments d'une matrice

```
% define a 3 x 3 matrix + extract the value of the 1st column
>> E = rand(3,3);
>> E(:,1)

ans =
    0.9501
    0.2311
    0.6068
```

Exemple : transformation groupée d'éléments dans une matrice (attention aux tailles !)

```
% define a 3 x 3 matrix E
>> E = ones(3,3);
% add the vector v = [3 6 9] as the third row of E
>> E(3,:) = [3:3:9];

E =
     1     1     1
     1     1     1
     3     6     9
```

Exemple : concaténation de matrices (attention à la correspondance des tailles !)

```
% define a 3 x 3 matrix B
>> B = eye(3,3);
% append the matrix B to the matrix E
>> G = [E B];

G =
     1     1     1     1     0     0
     1     1     1     0     1     0
     3     6     9     0     0     1
```

Exemple : suppression de colonnes/lignes dans une matrice

```
% remove columns 2 to 3
>> G(:,2:3) = []

G =
     1     1     0     0
     1     0     1     0
     3     0     0     1
```

Réarrangement de la taille d'une matrice

```
reshape(A, m, n)   réarrange une matrice A de dimension
                   donnée en une matrice de dimension m x n
                   Le nombre d'éléments doit correspondre !
```

Exemple : reshape

```
% reshape the matrix A of dimension 2 x 3 in dimension 3 x 2
>> A = [1 2 3 ; 4 5 6];
>> B = reshape (A, 3, 2)

B =
     1     5
     4     3
     2     6
```

4. Opérations sur les matrices

Différents types d'opérations mathématiques peuvent être effectuées sur les matrices :

4.1. Opérations coefficient par coefficient

symboles	Description	symboles	Description
.*	multiplication	./	division droite
.^	exponentation	.\	division gauche

Les opérations coefficient par coefficient s'effectuent en faisant précéder l'opérateur d'un point :

```
C = A .* B   produit coeff par coeff, Cij = Aij Bij

C = A ./ B   division droite coeff par coeff, Cij = Aij/Bij

C = A .\ B   division gauche coeff par coeff, Cij = Bij\Aij

C = A.^n     nième puissance coeff par coeff, Cij= Aij^n
```

4.2. Opérations matricielles

symboles	Description
+	somme matricielle
–	soustration matricielle
*	produit matriciel
^	puissance matricielle
/	division matricielle droite*
\	division matricielle gauche*

* permet de résoudre les systèmes linéaires de type A x = B

Attention, ces opérations ne sont possibles que si les matrices ont des dimensions qui le permettent !

```
C = A + B    somme matricielle, Cij = Aij + Bij

C = A * B    produit matriciel

C = A / B    division matricielle

C = A^n      nième puissance matricielle (matrices carrées uniqu.)
```

Nb: une explication pratique concernant la division matricielle droite et gauche est donnée dans l'exercice.

4.3. Opérations matricielles de base en algèbre linéaire

symboles	Description
`	transposée de la matrice
inv	matrice inverse (A^{-1})
range	portée de la matrice par colonne
rank	rang de la matrice (# lignes indépendantes)
eig	valeurs propres d'une matrice (diagonalisation)
det	déterminant de la matrice
norm	norme de la matrice ou vecteur
trace	trace de la matrice (∑ éléments diagonaux)
orth	vecteurs de base orthonormés de la matrice

`C = A'`	transposée de A, $C_{ij} = A^T_{ij} = A_{ji}$
`rg = range(A)`	retourne la différence entre le maximum et le minimum des éléments de A (par colonne).
`C = inv(A)`	inverse de A, $C = A^{-1}$
`d = det(A)`	déterminant de A
`r = rank(A)`	rang de A
`n = norm(A)`	norme de A
`b = trace(A)`	somme des éléments de la diagonale de A
`B = orth(A)`	retourne les vecteurs de base orthonormés tel que B'*B = I. La matrice I est la matrice identité.
`[V, D] = eig(A)`	Valeurs propres et vecteurs propre de A A*V = V*D (D: valeur propre, V: vecteur propre)

Exemple: calcul du produit de deux vecteurs x et y

```
%define the column vector x and y
>> x = [1;2;3]
>> y = [4;5;6]

%compute the inner product of the two vectors

>> x'*y

ans =
    32
```

Exemple : calcul de la norme d'un vecteur $\|x\|_2 = \sqrt{\left(\sum_{i=1}^{M} |x_i|^2\right)}$

```
% the euclidian norm of the v = [3 8 1] is:
>> sum(abs(v).^2)^(1/2)

ans =
    8.6023

% using the operator
>> norm(v)

ans =
    8.6023
```

Exemple : calcul de la matrice inverse

```
% define the matrix A
>> A = [1 2; 3 4]

A =
     1     2
     3     4

% compute the determinant of A
>> det_A = (1*4)-(2*3)

det_A =
    -2

% compute the inverse of A
>> A_inv = 1/det_A*[4 (-2); (-3) 1]

A_inv =
   -2.0000    1.0000
    1.5000   -0.5000

% using the operator
>>inv(A)

ans =
   -2.0000    1.0000
    1.5000   -0.5000

% according to the theory : A*A-1 = A-1*A = I
>> A*A_inv

ans =
     1     0
     0     1
```

5. Références

[1] Wikipedia.org

[2] Amos, Gilat, 2007. *Matlab, an introduction with application,* Johne Willey and Sohn, Inc.

[3] Hudon Nicolas, 2004. *Initiation à MATLAB*. URCPC, Ecole Polytechnique de Montréal.

[4] MATLAB Help

6. Auteurs

Alexandre Loye (2009)

Pascal Horton (2015)