Université Mohamed Khider – Biskra
Faculté des Sciences et de la Technologie
Département Génie mécanique

جامعة محمد خيضر- بسكرة
كليّة العلوم والتكنولوجيا
قسم الهندسة الميكانيكية

Licence 2

Informatique 3
2025/2026

TP 4
Vecteurs et matrices (Matlab)

Ensg : Hadef Saddek

TP 04 : Matrices et vecteurs

Notations matricielles.

Une matrice se présente sous la forme de données encadrées par 2 crochets. Chaque élément d'une ligne est séparé du suivant par une virgule ou des espaces. Le point-virgule sert à passer à la ligne suivante.
$A = [a_{11} , \dots , a_{1q} ; \dots ; a_{p1} , \dots , a_{pq}]$ désigne une matrice de dimension (p , q) , c'est-à-dire de p lignes et q colonnes.
Les éléments $A(i , j) = a_{ij}$ sont référencés en colonne : $A(i , j) = A(i + (j - 1)*p)$ avec $i = 1, \dots , p$ et $j = 1, \dots , q$
La matrice transposée et conjuguée de A est notée A' : $A'(i, j) = \overline{A(j, i)}$. Noter A.' pour une transposition seule.
L'application d'une fonction f usuelle (sin, exp, sqrt, ...) aux éléments de A s'écrit f(A) : $f(A) = f([a_{ij}]) = [f(a_{ij})]$
Si $I = [i_1, \dots , i_r]$ et $J = [j_1, \dots , j_s]$ sont 2 vecteurs d'indices, alors $A(I , J) = [a_{ij}]_{i \in I, j \in J}$ est une matrice extraite de A.
<u>NB.</u> Par convention [λ] est assimilée au scalaire λ et [ ] symbolise la matrice vide.

Cas p = 1 ou q = 1 :

$x = [v_1 , \dots , v_q]$ désigne un vecteur ligne (1 , q) composé de q valeurs v_j $\Rightarrow$ $x(j) = v_j$
$y = [w_1 ; \dots ; w_p]$ désigne un vecteur colonne (p , 1) composé de p valeurs w_i $\Rightarrow$ $y(i) = w_i$
Soit n vecteurs lignes réels $x_1 , \dots , x_n$ de même dimension :
$X = [x_1 ; \dots ; x_n]$ désigne la matrice composée des vecteurs x_i en ligne $\Rightarrow$ $X(i , :) = x_i$
$X' = [x_1 ; \dots ; x_n]'$ désigne la matrice composée des vecteurs x_i en colonne $\Rightarrow$ $X'(: , j) = x_j'$

Opérations de base :

Les symboles +, -, *, /, ^ sont utilisés pour les opérations habituelles sur les matrices et les symboles .*, ./, .^ sont utilisés pour les opérations élément par élément : A *.op* B = $[a_{ij}\ op\ b_{ij}]$
L'antislash \ sert à inverser une matrice : $A \backslash B = A^{-1} * B$
L'expression x = A \ b revient à résoudre A* x = b , et l'expression x = A / b revient à résoudre x * b = A
Le symbole .*. représente le produit tensoriel : $A .*. B = A \otimes B = [a_{ij} * B]$
<u>NB.</u> Validité des opérations matricielles :

$(p , q) \pm (p , q) = (p , q)$
$(p , q) * (q , r) = (p , r)$
$(p , q) / (r , q) = (p , r)$
$(p , q) \backslash (p , r) = (q , r)$

Fonctions usuelles.

- zeros(p , q) fournit la matrice p lignes, q colonnes dont les éléments sont des 0.
- ones(p , q) fournit la matrice p lignes, q colonnes dont les éléments sont des 1.
- eye(n , n) fournit la matrice identité d'ordre n.
- norm(x) fournit la norme euclidienne du vecteur x.
- det(A) fournit le déterminant de la matrice A.
- inv(A) fournit la matrice inverse de la matrice A.
- spec(A) fournit les valeurs propres de la matrice A.

Discrétisation.

La fonction *linspace* permet de créer un vecteur ligne constitué de points régulièrement espacés entre une valeur a et une valeur b. Sa syntaxe est : linspace(valeur a, valeur b, nombre de points). Le pas est égal à (b - a) / (nb_pts - 1).
Par exemple : x = linspace(0, 10, 6) est équivalent à x = [0, 2, 4, 6, 8, 10]
Pour obtenir des points équidistants à partir d'une valeur a jusqu'à une valeur limite b, on peut aussi écrire :
valeur a : valeur du pas : valeur b. Par défaut, la valeur du pas est de +1 (a:b ⇔ a:1:b).
Par exemple : x = 0:2:11 est équivalent à x = [0, 2, 4, 6, 8, 10]
<u>NB.</u> Avec linspace les bornes a et b sont toujours atteintes.

EXEMPLES.

```
A = [1,2;3,4] = 1.   2.  ;  B = [5,6;7,8] = 5.   6.
                3.   4.                     7.   8.

x = [9,10] = 9.   10.  ;  y = [11,12] = 11.   12.  ;  z = [13;14] = 13.
                                                                    14.
```

Addition :

```
A + B = B + A = 6.    8.  ;  A + 1 = 2.   3.  ;  x + y = 20.   22.
                10.   12.            4.   5.
```

Soustraction :

```
A - B = -4.   -4.  ;  B - A = 4.   4.  ;  x - y = -2.   -2.
        -4.   -4.           4.   4.
```

Multiplication :

```
5 * A = 5 .* A = A * 5 = A .* 5 = 5.    10.
                                  15.   20.

A * B = 19.   22.  ;  B * A = 23.   34.  ;  A .* B = B .* A = 5.    12.
        43.   50.           31.   46.                         21.   32.

A * z = 41.  ;  x * y' = 219.  ;  x' * y = 99.    108.
        95.                               110.   120.
```

Division :

```
A / 2 = A ./ 2 = 0.5   1.  ;  3 / A = -6.     3.  ;  3 ./ A = 3.   1.5

                 1.5   2.              4.5   -1.5             1.   0.75

A / B = 3.   -2.  ;  A ./ B = 0.2         0.3333333
        2.   -1.              0.4285714   0.5

x / A = -3.   4.  ;  x ./ y = 0.8181818   0.8333333
```

Exponentiation :

```
A ^ 2 = 7.    10.  ;  A .^ 2 = 1.   4.  ;  x ^ 3 = x .^ 3 = 729.   1000.
        15.   22.            9.   16.
```

Inversion :

```
A \ B = -3.   -4.  ;  A \ 1 = -2.    1.  ;  A \ z = -12.
         4.    5.             1.5  -0.5              12.5
```

Fonction :

```
exp(A) = 2.7182818   7.3890561  ;  spec(A) =  5.3722813
         20.085537   54.59815                -0.3722813
```

Produit tensoriel :

```
A .*. B = 5.    6.    10.   12.  ;  x .*. y = 99.   108.   110.   120.
          7.    8.    14.   16.
          15.   18.   20.   24.
          21.   24.   28.   32.
```

Résolutions de système

On recherche de résoudre le système d'équation :

$$\begin{cases} x + \sigma y = 2 \\ 2x + 7y = 3 \end{cases}$$

On va écrire sous forme matricielle on 'a :

$$\begin{cases} x + \sigma y = 2 \\ 2x + 7y = 3 \end{cases} \Leftrightarrow \begin{pmatrix} 1 & 6 \\ 2 & 7 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

On suppose $A = \begin{pmatrix} 1 & 6 \\ 2 & 7 \end{pmatrix}; X = \begin{pmatrix} x \\ y \end{pmatrix} et\ Y = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$

Donc on 'a : $AX = Y$

<u>On résoudre sous Matlab</u>

Premiers cas : avec \	Deuxième cas : avec la fonction linsolve(a,b) pour résoudre l'équation ax+b=0	Troisième cas : avec l'inverse d'une matrice
$AX = Y \Rightarrow X = A\backslash Y$ Matlab --> A=[1 6; 2 7] A = 1. 6. 2. 7. --> Y=[2;3] Y = 2. 3. --> X=A\Y X = 0.8 0.2	$AX = Y \Rightarrow AX + (-y) = 0$ --> linsolve(A,-Y) ans = 0.8 0.2	$AX = Y \Rightarrow X = A^{-1}Y$ --> inv(A)*Y ans = 0.8 0.2

Quatrième cas : on utilise un algorithme

Exemple : Gauss

$$\begin{cases} x + 6y = 2 & L1 \\ 2x + 7y = 3 & L2 \end{cases}$$

$$\begin{cases} x + \sigma y = 2 & L1 \\ -5y = -1 & L2 \rightarrow L2 - 2L1 \end{cases}$$

$$\begin{cases} -5x = -4 & L1 \rightarrow -5L1 - 6L2 \\ -5y = -1 \end{cases}$$

$$\begin{cases} L1 \rightarrow -\frac{1}{5}L1 \\ L2 \rightarrow -\frac{1}{5}L2 \end{cases}$$

$$B = \begin{bmatrix} 1 & 6 & \vdots & 2 \\ 2 & 7 & & 3 \end{bmatrix}$$

```
// Algorithme (Gauss)
clear;clc
A=[1 6; 2 7];Y=[2;3];B=[A,Y];
if A(1,1)*A(2,2)-A(2,1)*A(1,2)==0 then
   x="pas de solution"
   y="pas de solution"
   disp(x,y)
else
   B(2,:)=B(1,1)*B(2,:) - B(2,1)*B(1,:)
   B(1,:)=B(2,2)*B(1,:) - B(1,2)*B(2,:)
   B(1,:)=B(1,:)/B(1,1)
   B(2,:)=B(2,:)/B(2,2)
   x=B(1,3)
   y=B(2,3)
   disp(x,y)
```

2 ligne de B

1ère ligne de B

```
clear;clc
n= input('entrer la
valeur n:')
m= input('entrer la
valeur m:')
for i=1:n
for j = 1:m
  if i == j then
    a(i,j) = 3;
  elseif (i-j) == 1 then
    a(i,j) = -1;
elseif i-j==-1
    a(i,j)=2
  else a(i,j) = 0;
  end,
end
end
disp(a)
```

```
entrer la valeur n:4


entrer la valeur m:4

  3.   2.   0.   0.

 -1.   3.   2.   0.

  0.  -1.   3.   2.

  0.   0.  -1.   3.
```