Université Batna 2.
Faculté des Mathématiques et de l'Informatique.
Département d'Informatique.
Année universitaire 2020-2021.

Cours de Méthodes Numériques

TP 01: opérations matricielles

Mme OTSMANE Sarah

1. Création rapide :

Certaines commandes permettent de créer plus rapidement des vecteurs précis :

>> l1=1:10 (Un vecteur contenant les entiers de 1 à 10)

>> l2=1:1:10

>> l3=10:-1:1

>> l4=1:0.3:pi

>> l1(2)=l3(3)

>>l4(3:5)=[1,2,3]

>>l4(3:5)=[]

>> l5=linspace(1,5,5)

>> help linspace

>>who

>>whos

>>clear l1 l2 l3 l5

>>who

>>clc (efface le contenu de la fenêtre de commande)

>>clear

NB : Une ligne de commande commençant par le caractère % n'est pas exécutée par « **Matlab** ». Cela permet d'insérer des lignes de commentaires. Et il faut commenter ses programmes, ... surtout ceux de l'examen !

Exercice 01:

Entrez les différents vecteurs et matrices et donnez la longueur et la taille de chacun (Utilisez help pour trouver les fonctions qui donnent longueur et taille).

Exercice 02:

Construire: (1) une suite partant de -8 et allant à -5 par pas de 0,25.

(2) une suite décroissante d'entiers de 15 à 3.

(3) une suite de longueur 100 de $-\pi$ à π.

2. Opérations vectorielles :

Les tableaux suivants résument certaines commandes couramment utilisées.

Vecteurs :

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

Matrices :

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

Matrice particulières :

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

Exercice 03 :

[Extraction de composantes] : Entrez la matrice

>> A=[1 2 3 ; 2 3 1 ; 3 1 2]

Quels sont les résultats des commandes suivantes ?

>>A([2 3],[1 3])

>>A([2 3],1:2)

>>A([2 3],:)

>>A([2 3],end)

>>A(:)

>>A(5)

>>reshape(A(:),size(A))

Exercice 04 :

Créez des matrices particulières. Exemple de création d'une matrice par blocs :

>> C=[A, zeros(3,2); zeros(2,3), eye(2)]

Exercice 05 :

Ecrire la matrice carrée M d'ordre 12 contenant les entiers de 1 à 144 rangés par ligne. Extraire de cette matrice les matrices suivantes :

- ✓ La sous-matrice formée par les coefficients $a_{i,j}$ pour i = 1,…, 6 et j = 7,…,12;
- ✓ Celles des coefficients $a_{i,j}$ pour $(i,j) \in (1,2,5,6,9)^2$;
- ✓ Celle des coefficients $a_{i,j}$ pour i + j pair.

Les fonctions :

sqrt	exp	log
sin	cos	tan
asin	acos	atan
round	floor	ceil
abs	angle	conj

Les opérations matricielles et les fonctions :

`A'`	transposée de `A`
`rank(A)`	rang de `A`
`inv(A)`	inverse de `A`
`expm(A)`	exponentielle de `A`
`det(A)`	déterminant de `A`
`trace(A)`	trace de `A`
`poly(A)`	polynôme caractéristique de `A`
`eig(A)`	valeurs propres de `A`
`[U,D]=eig(A)`	vecteurs propres et valeurs propres de `A`
`+ -`	addition, soustraction
`* ^`	multiplication, puissance (matricielles)
`.* .^`	multiplication, puissance terme à terme
`A\b`	solution de $Ax = b$
`b/A`	solution de $xA = b$
`./`	division terme à terme

NB : A/b est équivalent à $inv(A) * b$ si A est inversible.

Fonctions vectorielles :

`max(x)`	maximum
`min(x)`	minimum
`sort(x)`	tri par ordre croissant
`[y, I] = sort(x)`	retourne en plus les indices des élements de `x`
`find(x)`	retourne les indices non nuls de `x`
`[y, I] = find(x)`	retourne des lignes (dans le vecteur `I`) et des colonnes (dans le vecteur `J`) des élements non nuls du `x`
`sum(x)`	somme des éléments de `x`
`cumsum(x)`	vecteur contenant la somme cumulée des éléments de `x`
`prod(x)`	produit des éléments de `x`
`cumprod(x)`	vecteur contenant le produit cumulé des éléments de `x`
`diff(x)`	vecteur des différences entre deux éléments consécutifs de `x`
`mean(x)`	moyenne des éléments de `x`
`median(x)`	médiane
`std(x)`	écart type

Exemple 01 :

Regardez l'effet des instructions suivantes.

```
>> x=rand(1,5)
>>mean(x)
>>std(x)
>>median(x)
>> sort(x)
>> A=rand(3)
>> sort(A)
>> [B, I]=sort(A)
>> sort(A')
>> max(A)
>> max(A')
>> max(max(A))
>>sum(A)
>>cumsum(A)
>>prod(A)
>>diff(A)
>> D=A([1,2],1:3)
>>sum(D,1)
>>sum(D,2)
```

Opérateurs relationnels et logiques :

`Opérateurs relationnels`	`<`, `<=`, `>=`, `==` (égalité), `~=` (différent)
`Opérateurs logiques`	`&` (et), `|` (ou), `~` ou `not` (non)

NB : Attention de ne pas confondre = qui sert à affecter une valeur à une variable et == qui sert à tester l'égalité.

Les opérateurs relationnels peuvent être utilisés avec des scalaires ou des matrices. Le résultat d'évaluation d'une expression relationnelle est 1 (vrai) ou 0 (faux). Appliqués à une matrice, ils rendent une matrice de même dimension, formée de 1 et de 0.

3. Représentation graphique des résultats :

➢ Représentations de points dans le plan :

Il existe plusieurs possibilités pour représenter un ensemble de points (x(i); y(i)). Les plus utilisées sont énumérées ci-dessous.

`plot(x,y,'s')`	tracé d'une courbe ou d'un nuage de points
`bar(x,y,'s')`	tracé sous forme d'un histogramme
`stem(x,y,'s')`	diagramme en bâtons
`stairs(x,y)`	tracé en escalier des valeurs discrètes
`fplot`	représente des fonctions
`hist`	trace des histogrammes

➢ Gestion de la fenêtre graphique :

`hold on`	les prochains tracés se superposeront aux tracés déjà effectués
`hold off`	le contenu de la fenêtre graphique active sera effacé lors du prochain tracé
`clf`	efface le contenu de la fenêtre graphique active
`figure(n)`	affiche ou rend active la fenêtre graphique numéro `n`
`close`	ferme la fenêtre graphique active
`close all`	ferme toutes les fenêtres graphiques
`subplot(n,m,p)`	partage la fenêtre graphique active en `m` × `n` espaces graphiques et sélectionne le `p`-ième.

Exemple 02 :

Sauvez dans le répertoire courant les lignes suivantes sous le nom losange.m :

```
>>x=[0 -1 0 1 ; -1 0 1 0]
>>y=[-1 0 1 0 ; 0 1 0 -1]
>>plot(x,y)
```

4. Les instructions conditionnelles :

La manière la plus brute de procéder est d'utiliser un bloc **'if ... then ... else ... end'**. La syntaxe en est la suivante :

>>if conditions

>>instructions

>>end

Les 'instructions' ne sont exécutées que si les 'conditions' sont vérifiées, plus précisément si 'conditions' à une valeur différente de 0. Une variante plus élaborée est :

>>if conditions

>>instructions (exécutées si les 'conditions' sont vérifiées)

>>else

>>instructions (exécutées si les 'conditions' ne sont pas vérifiées)

>>end

Ou encore (avec des emboitements) :

>>if conditions 1

>>instructions (exécutées si les 'conditions1' sont vérifiées)

>>elseif conditions 2

>>instructions (exécutées si les 'conditions1' ne sont pas vérifiées mais si les 'conditions2' le sont)

>>else

>>instructions (exécutées si ni les 'conditions1' ni les 'conditions2' ne sont vérifiées)

>>end

NB : « **Matlab »** gère les commandes classiques switch et case.

L'instruction while :

Ce format de boucle permet de s'arrêter conditionnellement (et non plus à rang fixé, comme dans une boucle for). La syntaxe en est la suivante :

>>while conditions

>>instructions

>>end

Les 'instructions' sont exécutées tant que les 'conditions' sont vérifiées.

Exercice 06:

Donnez le code « **Matlab »** qui permet de :

1. Définir le vecteur ***V = [0, 1, 2, 3,…49, 50].***
2. Quelle est la taille de vecteur ***V*** ?
3. Définir le vecteur ***W*** contenant les cinq premiers éléments de ***V***.
4. Définir le vecteur ***X*** contenant les cinq premiers et les cinq derniers éléments de ***V***.
5. Définir ensuite le vecteur ***Z = [0, 2, 4,…48, 50]*** à partir de ***V.***

Exercice 07:

Donnez le code « **Matlab »** qui permet de :

1. Créez un vecteur colonne ***vec*** de 5 éléments linéairement espacés entre 2 et 3.
2. Ajoutez deux lignes à la fin de ce vecteur avec la valeur 0.
3. Ajoutez 1 aux deuxièmes et sixièmes éléments de ce vecteur.
4. Créez un second vecteur ***vec2*** colonne de même dimension que ***vec*** contenant les entiers pairs supérieurs ou égaux à 6.
5. Définir un vecteur ***sumvec*** comme la somme des deux vecteurs ***vec*** et ***vec2***.
6. Définir un vecteur ***prodvec*** comme le produit termes à termes des deux vecteurs ***vec*** et ***vec2***.
7. Quel est la somme des éléments de ***prodvec*** ?
8. Quel est le plus grand élément du vecteur ***prodvec*** ?

9. Renverse l'ordre des éléments du vecteur ***prodvec***.

10. Quel est le plus petit élément du vecteur ***sumvec*** ?

11. Quel est la moyenne des éléments de ***sumvec*** ?

12. Ordonne les éléments du ***sumvec*** vecteur par ordre croissant.

13. Quel est le plus grand élément du vecteur $\boldsymbol{v3} = \frac{\boldsymbol{vec^2+\sqrt{vec2+1}}}{\boldsymbol{vec}.(\boldsymbol{vec2+1})}$?

Exercice 08 :

Donnez le code « **Matlab »** qui permet de :

1. Définir la matrice $A = \begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix}$.
2. Quelle est la taille de la matrice *A*.
3. Définir la matrice identité d'ordre 10.
4. Définir une matrice à 10 linges et 15 colonnes dans tous les éléments valent 1.
5. Définir une matrice à 5 linges et 2 colonnes dans tous les éléments valent 0.
6. Définir une matrice à 6 linges et 10 colonnes dont les éléments sont générés de manière aléatoire entre 0 et 1.
7. Construire la matrice $M = \begin{pmatrix} 12 & \pi \\ 3i+4 & 1 \end{pmatrix}$.
8. Créez les matrices

$$a = \begin{pmatrix} 1 & 5 \\ -3 & 2 \end{pmatrix}, b = \begin{pmatrix} 4 & 2 \\ 0 & -4 \end{pmatrix}, c = \begin{pmatrix} 1 & 2 & 3 \\ 5 & 4 & 6 \end{pmatrix}$$

9. Construire les matrices $a + b, a.* b, a * c \; et \; a.\hat{}2$
10. Définir par blocs la matrice D :

$$D = \left(\begin{array}{ccc|cc|c} 35 & 1 & 6 & 26 & 19 & 24 \\ 3 & 32 & 7 & 21 & 23 & 25 \\ 31 & 9 & 2 & 22 & 27 & 20 \\ \hline 8 & 28 & 33 & 17 & 10 & 15 \\ 30 & 5 & 34 & 12 & 14 & 16 \\ \hline 4 & 36 & 29 & 13 & 18 & 11 \end{array}\right).$$

Exercice 09 :

a) Construire la matrice T triangle de la commande diag() utilisée 3 fois

$$T = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

b) Extraire de T les deux premières colonnes.

c) Extraire de T les éléments des colonnes et des linges 2 à 4.

d) Créer une matrice T_2 ou linge 1 est échangée avec linge 3 puis la colonne 2 est remplacée par les valeurs de la colonne 4.

e) Construire la matrice triangulaire inférieure de la matrice T_2.

f) Construire la matrice triangulaire supérieure de la matrice T_2.

g) Quel est la transposée et l'inverse de la matrice T_2.