TP1- Calcul matriciel

EXERCICE 1 : (1 SEANCE DE TP (30/31 OCT. 2024))

1. Créer une matrice ***A*** de 5x5 éléments en utilisant la commande '***magic***'.
2. Calculer la trace de ***A***.
3. A l'aide de la commande '***sum***' vérifier qu'il s'agit bien d'un carré magique.
4. En utilisant les commandes '***diag***', '***tril***' et '***triu***' créer les matrices ***B***, ***C***, ***D*** et ***E*** à partir de la matrice initiale ***A***.

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix}$$

$$B = \begin{pmatrix} 17 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 \\ 0 & 0 & 0 & 21 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{pmatrix} \quad C = \begin{pmatrix} 17 & 24 & 0 & 0 & 0 \\ 23 & 5 & 7 & 0 & 0 \\ 4 & 6 & 13 & 20 & 0 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix}$$

$$D = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 0 & 6 & 13 & 20 & 22 \\ 0 & 0 & 19 & 21 & 3 \\ 0 & 0 & 0 & 2 & 9 \end{pmatrix} \quad E = \begin{pmatrix} 0 & 24 & 1 & 8 & 15 \\ 23 & 0 & 7 & 14 & 16 \\ 4 & 6 & 0 & 20 & 22 \\ 10 & 12 & 19 & 0 & 3 \\ 11 & 18 & 25 & 2 & 0 \end{pmatrix}$$

5. Créer la matrice ***F*** la concaténation des 4 dernières matrices.

$$F = \begin{pmatrix} C & B \\ D & E \end{pmatrix}$$

6. Créer la matrice ***G*** la concaténation des parties encerclées de ***C***, ***D*** et ***E*** :

$$C = \begin{pmatrix} 17 & 24 & 0 & 0 & 0 \\ 23 & 5 & 7 & 0 & 0 \\ 4 & 6 & 13 & 20 & 0 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad D = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 3 & 5 & 7 & 14 & 16 \\ 0 & 6 & 13 & 20 & 22 \\ 0 & 0 & 19 & 21 & 3 \\ 0 & 0 & 0 & 2 & 9 \end{pmatrix}$$

$$E = \begin{pmatrix} 0 & 24 & 1 & 8 & 15 \\ 23 & 0 & 7 & 14 & 16 \\ 4 & 6 & 0 & 20 & 22 \\ 10 & 12 & 19 & 0 & 3 \\ 11 & 18 & 25 & 2 & 0 \end{pmatrix} \quad G = \begin{pmatrix} 17 & 24 & 8 & 0 \\ 23 & 5 & 14 & 23 \\ 4 & 6 & 20 & 4 \\ 21 & 3 & 0 & 10 \\ 2 & 9 & 2 & 11 \end{pmatrix}$$

7. Calculer la densité de la matrice ***F*** (le nombre des éléments non nuls / le nombre total des éléments).
8. Supprimer de ***F*** la *1ère*, et la dernière ligne ainsi que la *3ème*, la *5ème* et la *6ème* colonne.
9. Inverser l'ordre des lignes et l'ordre des colonnes de ***F***.
10. Permuter la *1ère* et la dernière ligne de ***F.***

<u>**Remarque :**</u> Copier la réponse sur une feuille de papier contenant vos affiliations (*Noms ; Prénoms et Groupes*).

EXERCICE 2 : (2 SEANCES DE TP (06/07 ET 13/14 NOV. 2024))

PARTIE1 :

1. Taper dans la *fenêtre de commande* Matlab les commandes suivantes en respectant leur ordre.

```
clc;
A=[3 1 0 ;-4 -1 0 ;4 8 -2]
AA=[A eye(3)]
AA(1,:)=AA(1,:)/AA(1,1)
AA(2,:)=AA(2,:)-AA(2,1)*AA(1,:)
AA(3,:)=AA(3,:)-AA(3,1)*AA(1,:)
AA(2,:)=AA(2,:)/AA(2,2)
AA(1,:)=AA(1,:)-AA(1,2)*AA(2,:)
AA(3,:)=AA(3,:)-AA(3,2)*AA(2,:)
AA(3,:)=AA(3,:)/AA(3,3)
B=AA(:,4:6)
```

2. Que fait cet ensemble de commandes ?
3. Ecrire un programme pour généraliser ce travail sur n'importe quelle matrice carrée d'ordre n.

PARTIE2 :

1. On appelle Mineur(i, j) d'une matrice A (de rang n) la matrice M (de rang n-1) obtenue en éliminant de A sa ligne i et sa colonne j. Ecrire une fonction Matlab '***Mineur***' qui calcule le mineur(i ,j) d'une matrice A.
2. Ecrire une fonction '***Determinant***' qui utilise la fonction précédente pour calculer le déterminant d'une matrice A de rang n.
3. Ecrire une fonction Matlab '***Comatrice***' qui utilise les fonctions précédentes pour calculer la matrice adjointe (comatrice) d'une matrice A de rang n.
4. Ecrire une fonction Matlab '***InverseDeterminant***' qui utilise les fonctions précédentes pour calculer l'inverse d'une matrice par la méthode du déterminant.
5. Ecrire une fonction Matlab '***InverseCofacteur***' pour calculer l'inverse d'une matrice par la méthode du cofacteur.

PARTIE 3 :

Ecrire un script Matlab qui affiche un menu, permettant à l'utilisateur de faire le choix entre la lecture d'une matrice de dimension n, le calcul du déterminant de cette matrice, et le calcul de l'inverse de cette matrice par les deux méthodes précédentes (utiliser les fonctions déjà écrites précédemment).

Remarque :

- Insérer un commentaire au début du programme contenant vos affiliations (Noms ; prénoms et groupes).
- Effacer la fenêtre de commande et utiliser le format d'affichage « short ».
- Afficher l'entête « ***= = = = = = = TP1 : Calcul matriciel = = = = = = =*** » en haut de la fenêtre de commande.
- L'indentation et la lisibilité du code, et l'explication des affichages sont prises en compte dans l'évaluation du travail.
- Interdit d'utiliser les commandes ***det*** , ***inv*** et ***A^-1*** du Matlab.

EXERCICE 3 : (N'A PAS ETE ATTRIBUE)

1. Créer les deux matrices ***A*** et ***δA*** suivantes (utiliser '***format long***') :

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \quad \delta A = \begin{pmatrix} -0{,}002553422381 & 0{,}004227714089 & -0{,}001061727529 & 0{,}000629732580 \\ 0{,}004227714089 & -0{,}006999847167 & 0{,}001757907531 & -0{,}001042651353 \\ -0{,}001061727529 & 0{,}001757907531 & -0{,}000441472337 & 0{,}000261846383 \\ 0{,}000629732580 & -0{,}001042651353 & 0{,}000261846383 & -0{,}000155306511 \end{pmatrix}$$

2. Créer le vecteur ***b***.

$$b = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

3. Calculer ***B = A + δA***
4. Résoudre les systèmes linéaires ***Ax = b*** et ***Bx = b***. Qu'est-ce qu'on peut remarquer à propos de cette matrice ?
5. Calculer les valeurs/vecteurs propres de ***A***.
6. Calculer les normes subordonnées de ***A***.
7. Calculer le conditionnement de ***A***.
8. Donner la décomposition de ***A*** en $A = PDP^{-1}$.
9. Evaluer l'erreur absolue et l'erreur relative dans le calcul de A^{10} entre le calcul direct (***A^10***) et le calcul par la décomposition PDP^{-1}.
10. Donner (à la main) la formule de A^n.

CORRIGE-TYPE

BAREME :

- **Exercice 1 : 10pts**
  - **Chaque question : 1pt**
- **Exercice 2 : 15pts**
  - **Partie 1 : 3pts**
  - **Partie 2 : 7pts**
  - **Partie 3 : 4pts**
  - **Lisibilité du code + Affichage des résultats clair : 1pt**

NOTE FINALE DU TP1 :

- **Somme des notes obtenues dans les deux exercices / 5.**

EXERCICE 1 : (10pts)

```
% Question 1: Création de A:
    A = magic(5);

% Question 2: Trace de A:
    trace(A)

 % Question 3: Vérification que la carré est magique :
    sum(A); sum(A,2);
  % ou bien
    isequal(sum(A),sum(A,2)')
  % ou bien
    isequal(min(sum(A)),max(sum(A)),min(sum(A,2)),max(sum(A,2)),trace(A),trace(fliplr
    (A)))

% Question 4: Création des matrice B, C, D et E:
    B = diag(diag(A));
  % ou bien
    B = A - tril(A,-1) - triu(A,1);
    % ou bien
    B = tril(triu(A));

    C = tril(A,1)
    D = triu(A,-1)
    E = tril(A-1) + triu(A,1);
  % ou bien
    E = A-B;

% Question 5: Concaténation des matrices B, C, D et E:
    F = [C B;D E]

 % Question 6: Concaténation des parties de C, D et E:
    G = [[C(1:3,1:2);D(4:5,4:5)] E(:,4) E(:,1)]

% Question 7: Densité de F:
    nnz(F)/prod(size(F));
  % ou bien
    nnz(F)/(size(F,1) * size(F,2));
  % ou bien
    nnz(F)/numel(F);

% Question 8: Suppression de la 1ère et dernière ligne puis la 3ème, la
% 5ème et la 6ème colonne:
    F(1,:) = [];
    F(end,:) = [];
    F(:,3) = [];
    F(:,5) = [];
    F(:,6) = []
  % ou bien
    F([1,end],:)=[]; F(:,[3,5,6])=[] ;
```

```
% Question 9: Inverser l'ordre des lignes puis celui des colonnes:
  F = flipud(F) ;
  F = fliplr(F);
 % ou bien
  F = flipud(fliplr(F));
 % ou bien
  F = F([end:-1:1,end:-1:1]);
% Question 10: Permutation de la 1ère et la dernière ligne de F:
  X = F(1,:);
  F(1,:) = F(end,:);
  F(end,:) = X;
 % ou bien
  F([1,end], :)= F([end,1], :)
```

EXERCICE 2 : (15 pts)

BAREME :

- **Partie 1 : 3 pts**
- **Partie 2 : 7 pts**
- **Partie 3 : 4 pts**
- **Lisibilité du code + Affichage des résultats expliqués : 1 pt**

PARTIE1 : (3 pts)

1. **Exécution des commandes :**
2. **Rôle des commandes : (1 pt)**

   Ces commandes permettent de calculer ***l'inverse*** *d'une matrice par la* ***méthode de Gauss****.*

3. **Programme Matlab : (2 pts)**

```
clc;
%A=[3 1 0 ;-4 -1 0 ;4 8 -2]
A=input(sprintf('\n\tENTRER UNE MATRICE CARREE : \n'));
n=size(A,1);
AA=[A eye(n)];
fprintf('\n\tVOICI LA MATRICE AA (la concatenation de A et I ) : \n\n');
disp(AA);
for i=1:n
    AA(i,:)=AA(i,:)/AA(i,i);
    for j=1:n
        if j~=i
            AA(j,:)=AA(j,:)-AA(j,i)*AA(i,:);
        end
    end
end
fprintf('\n\tVOICI LA MATRICE AA APRES TRAITEMENT : \n\n');
disp(AA);
B=AA(:,n+1:end);
fprintf('\n\tVOICI LA MATRICE INVERSIBLE DE A : \n\n');
disp(B);
```

PARTIE2 : (7pts)

1. Fonction Mineur : (1pt)

```
function [ M ] = Mineur( A,L,C )
M=A;
M(:,C)=[];
M(L,:)=[];
end
```

2. Fonction Déterminant : (2pts)

```
function [ D ] = Determinant( A )
n=size(A,1);
s=1;
D=0;
if n==1
    D=A(1,1);
else
    for j=1:n
        D=D+s*A(1,j)*Determinant(Mineur(A,1,j));
        s=-s;
    end
end
end
```

3. Fonction Comatrice : (2pts)

```
function [ C ] = Comatrice( A )
n=size(A,1);
for i=1:n
    for j=1:n
        C(i,j)=(-1)^(i+j)*Determinant(Mineur(A,i,j));
    end
end
end
```

4. Fonction InverseDeterminant : (1pt)

```
function [ B ] = InverseDeterminant( A )
d=Determinant(A);
if d~=0
    B=1/d*transpose(Comatrice(A));
else
    disp('Matrice n''est pas inversible');
end
end
```

5. Fonction InverseCofacteur : (1pt)

```
function [ B ] = InverseCofacteur( A )
B=A\eye(size(A,1));
end
```

PARTIE 3 : (4pts)

Script Matlab (Menu + Appel des fonctions) : (4pts)

```
clc;
format;
fprintf('= = = = = = = = = = = TP: Calcul Matriciel = = = = = = = = = = =\n\n\n');


Choix=-1;
while Choix~=0
    clc;
    fprintf('= = = = = TP: Calcul Matriciel = = = = =\n\n');
    fprintf('Entrer votre choix :\n\n');
    fprintf('\t\t[0]: Terminer le programme.\n');
    fprintf('\t\t[1]: Lire une matrice carrée.\n');
    fprintf('\t\t[2]: Calculer le déterminant.\n');
    fprintf('\t\t[3]: Calculer l''inverse par la méthode du déterminant.\n');
    fprintf('\t\t[4]: Calculer l''inverse par la méthode des cofacteurs.\n\n');
    A=[1 3 3;-2 11 -2;8 -7 6];
    Choix=input(sprintf('\t\tChoix = '));
    switch Choix
        case 0, fprintf('Exécution terminée');
        case 1, A=input(sprintf('Entrer une matrice carrée\n'));
            fprintf('Voici la matrice A :\n ');
            disp(A);
            input('Une touche pour continuer!!');
        case 2, D=Determinant(A);
            fprintf('Voici le déterminant de la matrice\n ');
            disp(D);
            input('Une touche pour continuer!!');
        case 3, B=InverseDeterminant(A);
            fprintf('Voici l''inverse de la matrice\n ');
            disp(B);
            input('Une touche pour continuer!!');
        case 4, B=InverseCofacteur(A);
            fprintf('Voici l''inverse de la matrice\n ');
            disp(B);
            input('Une touche pour continuer!!');
    end
end
```