TP3- Ecrit

Nom : . . . .

Prénom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Donne le résultat qui sera affiché par Matlab pour chaque commande :

`triu(tril(A))` ***(1^{pt})***

```
        17     0     0     0     0
         0     5     0     0     0
ans =    0     0    13     0     0
         0     0     0    21     0
         0     0     0     0     9
```

Le résultat est sauvegardé dans ans.

`c=A\b` ***(1^{pt})***

```
          1.0000
          3.0000
c =       5.0000
          7.0000
          9.0000
```

`eig(ans)` ***(1^{pt})***

```
         5
         9
ans =   13
        17
        21
```

ans est le résultat de la 1ère commande ; c'est une matrice diagonale ; les valeurs propres d'une matrice diagonale sont les valeurs de la diagonale (propriétés des matrices diagonale. chapitre : calcul matriciel). L'ordre des valeurs n'est pas pris en considération.

1

Soit la fonction Matlab suivante :

```
function [ A ] = GT( A,b )
A=[A,b];
n=size(A,1);
for k=1:n-1
    % Insert your code here
    for i=k+1:n
        b=A(i,k);
        for j=k:n+1
            A(i,j)=A(i,j)-b*A(k,j)/A(k,k);
        end
    end
end
end
```

b) Quel est le rôle de cette fonction ? **(1pt)**

Rôle : triangularisation d'une matrice (triangularisation : créer une matrice triangulaire supérieure) selon la méthode de Gauss. Elle ne fait pas la résolution d'un système linéaire.

c) On veut insérer la ligne de code *A=PPN(A);* à la place du commentaire dans la fonction GT. Sachant que PPN est une fonction Matlab qui applique la stratégie du 1er pivot non nul. Ecris le code de cette fonction. Qu'est-ce qu'on doit ajouter aussi à la fonction GT ? **(2pts)**

```
function [ A ] = PPN( A,k )
N=size(A,1);
i=k;
while (i<=N)&&(A(i,k)==0)
    i=i+1;
end
if i<=N
    A([i,k],:)=A([k,i],:);
end
end
```

Il faut ajouter dans la fonction GT la ligne

```
if A(k,k)==0
    A = PPN( A,k )
end
```

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

Exercice 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Donne le résultat qui sera affiché par Matlab pour chaque commande :

`B = A(end:-1:1,:)` *(1pt)*

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

`triu(tril(B))` *(1pt)*

$$ans = \begin{matrix} 11 & 0 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 \\ 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 0 & 0 & 15 \end{matrix}$$

Le résultat est sauvegardé dans ans.

`eig(ans)` *(1pt)*

$$ans = \begin{matrix} 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{matrix}$$

Les valeurs propres d'une matrice diagonale sont les valeurs de la diagonale (ordre n'est pas important).

Soit la fonction Matlab suivante :

```
function [ A ] = GT( A,b )
A=[A,b];
n=size(A,1);
for k=1:n-1
    % Insert your code here
    for i=k+1:n
        b=A(i,k);
        for j=k:n+1
            A(i,j)=A(i,j)-b*A(k,j)/A(k,k);
        end
    end
end
end
```

b) Quel est le rôle de cette fonction ? *(1pt)*

Le rôle : triangularisation d'une matrice (triangularisation : créer une matrice triangulaire supérieure) selon la méthode de Gauss. Elle ne fait pas la résolution d'un système linéaire.

c) On veut insérer la ligne de code *A=MPL(A);* à la place du commentaire dans la fonction GT. Sachant que MPL est une fonction Matlab qui applique la stratégie du meilleur pivot local. Ecris le code de cette fonction. *(2pts)*

```
function [ A ] = MPL( A,k )
N=size(A,1);
MaxVal=abs(A(k,k));
MaxInd=k;
for i=k+1:N
    if abs(A(i,k))>MaxVal
        MaxVal=abs(A(i,k));
        MaxInd=i;
    end
end
if MaxInd~=k
    A([MaxInd,k],:)=A([k,MaxInd],:);
end
end
```

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Donne le résultat qui sera affiché par Matlab pour chaque commande :

triu(A) *(1pt)*

$$ans = \begin{matrix} 17 & 24 & 1 & 8 & 15 \\ 0 & 5 & 7 & 14 & 16 \\ 0 & 0 & 13 & 20 & 22 \\ 0 & 0 & 0 & 21 & 3 \\ 0 & 0 & 0 & 0 & 9 \end{matrix}$$

B = tril(ans); *(1pt)*

Matlab calcule la matrice B (qui est une matrice diagonale), mais elle n'est pas affichée

B\eye(5) *(1pt)*

Cette commande calcule l'inverse de la matrice B et le met dans ans ; le fait que B est une matrice diagonale son inverse est aussi une matrice diagonale dont les valeurs sont les inverses des valeurs de B (facile à calculer)

$$ans = \begin{matrix} 0.0588 & 0 & 0 & 0 & 0 \\ 0 & 0.2000 & 0 & 0 & 0 \\ 0 & 0 & 0.0769 & 0 & 0 \\ 0 & 0 & 0 & 0.0476 & 0 \\ 0 & 0 & 0 & 0 & 0.1111 \end{matrix} \qquad ans = \begin{matrix} 1/17 & 0 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 & 0 \\ 0 & 0 & 1/13 & 0 & 0 \\ 0 & 0 & 0 & 1/21 & 0 \\ 0 & 0 & 0 & 0 & 1/9 \end{matrix}$$

eig(ans) *(1pt)*

Les valeurs propres sont les valeurs de la diagonale (ordre n'est pas important)

$$ans = \begin{matrix} 0.0476 \\ 0.0588 \\ 0.0769 \\ 0.1111 \\ 0.2000 \end{matrix} \qquad ans = \begin{matrix} 1/21 \\ 1/17 \\ 1/13 \\ 1/9 \\ 1/5 \end{matrix}$$

Soit la fonction Matlab suivante :

```
function [ A ] = GT( A,b )
A=[A,b];
n=size(A,1);
for k=1:n-1
    for i=k+1:n
        b=A(i,k);
        for j=k:n+1
            A(i,j)=A(i,j)-b*A(k,j)/A(k,k);
        end
    end
end
end
```

b) Quel est le rôle de cette fonction ?

Le rôle : triangularisation d'une matrice (triangularisation : créer une matrice triangulaire supérieure) selon la méthode de Gauss. Elle ne fait pas la résolution d'un système linéaire. ***(1pt)***

c) Réécris cette fonction en éliminant la boucle interne (*for j=k : n+1*) sans que la fonction perd son rôle. ***(1pt)***

```
function [ A ] = GT( A,b )
A=[A,b];
n=size(A,1);
for k=1:n-1
    for i=k+1:n
        b=A(i,k);

        A(i,:)=A(i,:)-b*A(k,:)/A(k,k);

    end
end
end
```

4

TP3- Ecrit

Nom : . . . .

Prénom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Donne le résultat qui sera affiché par Matlab pour chaque commande :

n = rank(A) *(1pt)*

La commande rank donne le rang de la matrice

```
n =   5
```

[v d] = eig(A); *(1pt)*

Cette commande calcule dans v les vecteurs propres de cette matrice sous forme d'une matrice (chaque colonne est un vecteur propre) et dans d, elle donne les valeurs propres sous forme d'une matrice diagonale. Mais le résultat n'est pas affiché (donc on n'est pas obligé de calculer)

A - v*d*inv(v) *(1pt)*

*v*d*inv(v) = A (propriété de la décomposition d'une matrice A en $A=P*D*P^{-1}$; donc la commande précédente donne une matrice nulle (malgré que les valeurs données par Matlab sont presque nul ($*10^{-13}$))*

```
ans =   0   0   0   0   0
        0   0   0   0   0
        0   0   0   0   0
        0   0   0   0   0
        0   0   0   0   0
```

Soit la fonction Matlab suivante :

```
function [ X ] = GJT(A)
n=size(A,1);
for k=1:n
    %Insert your code here
    A(k,:)=A(k,:)/A(k,k);
    for i=1:n
        if i~=k
            b=A(i,k);
            for j=1:n+1
                A(i,j)=A(i,j)-b*A(k,j);
            end
        end
    end
end
X=A(:,end);
end
```

b) Quel est le rôle de cette fonction ? ***(1pt)***

Le rôle : Résolution d'un système linéaire écrit sous forme d'une matrice augmentée par la méthode de Gauss-Jordan

c) On veut insérer la ligne de code *A=PPN(A);* à la place du commentaire dans la fonction GJT. Sachant que PPN est une fonction Matlab qui applique la stratégie du 1er pivot non nul. Ecris le code de cette fonction. Qu'est-ce qu'on doit ajouter aussi à la fonction GJT ? ***(2pts)***

```
function [ A ] = PPN( A,k )
N=size(A,1);
i=k+1;
while (i<=N)&&(A(i,k)==0)
    i=i+1;
end
if i<=N
    A([i,k],:)=A([k,i],:);
end
end
```

Il faut ajouter dans la fonction GJT la ligne.

```
if A(k,k)==0
    A = PPN( A,k )
end
```

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Donne le résultat qui sera affiché par Matlab pour chaque commande :

`B = A(end:-1:1,:)` ***(1^{pt})***

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

`triu(tril(B))` ***(1^{pt})***

$$ans = \begin{matrix} 11 & 0 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 \\ 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 0 & 0 & 15 \end{matrix}$$

Le résultat est sauvegardé dans ans.

`eig(ans)` ***(1^{pt})***

$$ans = \begin{matrix} 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{matrix}$$

Les valeurs propres d'une matrice diagonale sont les valeurs de la diagonale.

Soit la fonction Matlab suivante :

```
function [ X ] = GJR(A,b)
A=[A b];
n=size(A,1);
for k=1:n
    % Insert your code here
    A(k,:)=A(k,:)/A(k,k);
    for i=1:n
        if i~=k
            A(i,:)=A(i,:)-A(i,k)*A(k,:);
        end
    end
end
X=A(:,end);
end
```

b) Quel est le rôle de cette fonction ? ***(1^{pt})***

Le rôle : Résolution d'un système linéaire défini par sa matrice associée A et son vecteur résultat b, par la méthode de Gauss-Jordan.

c) On veut insérer la ligne de code *A=MPL(A);* à la place du commentaire dans la fonction GJR. Sachant que MPL est une fonction Matlab qui applique la stratégie du meilleur pivot local. Ecris le code de cette fonction. ***(2^{pts})***

```
function [ A ] = MPL( A,k )
N=size(A,1);
MaxVal=abs(A(k,k));
MaxInd=k;
for i=k+1:N
    if abs(A(i,k))>MaxVal
        MaxVal=abs(A(i,k));
        MaxInd=i;
    end
end
if MaxInd~=k
    A([MaxInd,k],:)=A([k, MaxInd],:);
end
end
```

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Pour chaque commande, donne le résultat affiché par Matlab :

```
n = rank(A)
```
(1pt)

La commande rank donne le rang de la matrice

```
n =  5
```

```
[v d] = eig(A);
```
(1pt)

Cette commande calcule dans v les vecteurs propres de cette matrice sous forme d'une matrice (chaque colonne est un vecteur propre) et dans d, elle donne les valeurs propres sous forme d'une matrice diagonale. Mais le résultat n'est pas affiché (donc on n'est pas obligé de calculer)

```
v*d*inv(v)
```
(1pt)

*v*d*inv(v) = A (propriété de la décomposition d'une matrice A en $A=P*D*P^{-1}$; donc la commande précédente donne la matrice A*

```
ans =
    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
```

```
det(ans([1,3],[2,5]))
```
(1pt)

ans([1,3],[2,5]) donne une matrice de 2x2 éléments.

```
ans =  438
```

```
    24    15
     6    22
```

Soit la fonction Matlab suivante :

```
function [ X ] = GJR(A,b)
A=[A b];
n=size(A,1);
for k=1:n
    A(k,:)=A(k,:)/A(k,k);
    for i=1:n
        if i~=k
            b=A(i,k);
            for j=1:n+1
                A(i,j)=A(i,j)-b*A(k,j);
            end
        end
    end
end
X=A(:,end);
end
```

b) Quel est le rôle de cette fonction ? *(1pt)*

Le rôle : Résolution d'un système linéaire défini par sa matrice associée A et son vecteur résultat b, par la méthode de Gauss-Jordan.

c) Réécris cette fonction en éliminant la boucle interne (*for j=1 : n+1*) sans que la fonction perd son rôle. *(1pt)*

```
function [ X ] = GJR(A,b)
A=[A b];
n=size(A,1);
for k=1:n
    A(k,:)=A(k,:)/A(k,k);
    for i=1:n
        %b=A(i,k);
        if i~=k
            A(i,:)=A(i,:)-A(i,k)*A(k,:);
        end
    end
end
X=A(:,end);
end
```

Remarque : l'instruction b=A(i,k) n'est plus importante lorsqu'on remplace la boucle par une seule ligne de commande.

7

TP3- Ecrit

Nom : . . . .

Prénom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Pour chaque commande, tu donnes le résultat affiché par Matlab :

`triu(tril(fliplr(A)))` *(1^{pt})*

```
        15     0     0     0     0
         0    14     0     0     0
ans =    0     0    13     0     0
         0     0     0    12     0
         0     0     0     0    11
```

`c=eig(ans)` *(1^{pt})*

L'ordre des éléments n'est pris en considération

```
      11
      12
c =   13
      14
      15
```

`B=power(ans,2)` *(1^{pt})*

La puissance d'une matrice diagonale est une matrice diagonale dont ses valeurs sont celles de la matrice initiale élevées à la puissance désirée.

```
      225     0     0     0     0
        0   196     0     0     0
B =     0     0   169     0     0
        0     0     0   144     0
        0     0     0     0   121
```

b) Ecris une fonction Matlab « DiagDominante » qui vérifie si une matrice M est à diagonale dominante ou non. *(1.5pt)*

```
function [ d ] = DiagDominante( A )
d=true;
n=size(A,1);
for i=1:n
    if sum(abs(A(i,:)))-abs(A(i,i))>abs(A(i,i))
        d=false;
        return
    end
end
end
```

c) En utilisant la fonction précédente écris une fonction Matlab « Converge » qui vérifie si la méthode de Jacobi converge pour un système linéaire Ax=b.

(1.5pts)

```
function [ Conv ] = Converge(A,b)
if DiagonaleDominante(A)
    Conv=true;
else
    D=diag(diag(A));
    E=-tril(A,-1);
    F=-triu(A,1);
    B=inv(D)*(E+F);
    Rho=vrho(B);
    if Rho<1
        Conv=true;
    else
        Conv=false;
    end
end
end
```

8

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Pour chaque commande, donne le résultat affiché par Matlab :

`B = [A X b]` *(1pt)*

```
B =
    17    24     1     8    15     1   285
    23     5     7    14    16     3   315
     4     6    13    20    22     5   425
    10    12    19    21     3     7   315
    11    18    25     2     9     9   285
```

`C = [B ;b'];` *(1pt)*

Erreur de concaténation: le nombre de colonnes de B est 7 par contre le nombre des éléments de b' est 5.

`[n,n]=size(B)` *(1pt)*

```
n =              n =

     7                7
```

b) Ecris une fonction Matlab « DiagDominante » qui vérifie si une matrice M est à diagonale dominante ou non. *(1.5pts)*

```
function [ d ] = DiagDominante( A )
d=true;
n=size(A,1);
for i=1:n
    if sum(abs(A(i,:)))-abs(A(i,i))>abs(A(i,i))
        d=false;
        return
    end
end
end
```

c) En utilisant la fonction précédente écris une fonction Matlab « Converge » qui vérifie si la méthode de Gauss-Seidel converge pour un système linéaire Ax=b.

(1.5pts)

```
function [ Conv ] = Converge(A,b)
if DiagonaleDominante(A)
    Conv=true;
else
    D=diag(diag(A));
    E=-tril(A,-1);
    F=-triu(A,1);
    B=inv(D-E)*F;
    Rho=vrho(B);
    if Rho<1
        Conv=true;
    else
        Conv=false;
    end
end
```

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

Exercice 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Pour chaque commande, donne le résultat affiché par Matlab :

`fliplr(A)` *(1pt)*

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

`B= A-(tril(A,-1)+ triu(A,1))` *(1pt)*

$$B = \begin{matrix} 17 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 \\ 0 & 0 & 0 & 21 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{matrix}$$

`B\eye(5)` *(1pt)*

Cette commande calcule l'inverse de la matrice B et le met dans ans ; le fait que B est une matrice diagonale son inverse est aussi une matrice diagonale dont les valeurs sont les inverses des valeurs de B (facile à calculer)

$$ans = \begin{matrix} 0.0588 & 0 & 0 & 0 & 0 \\ 0 & 0.2000 & 0 & 0 & 0 \\ 0 & 0 & 0.0769 & 0 & 0 \\ 0 & 0 & 0 & 0.0476 & 0 \\ 0 & 0 & 0 & 0 & 0.1111 \end{matrix} \quad ans = \begin{matrix} 1/17 & 0 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 & 0 \\ 0 & 0 & 1/13 & 0 & 0 \\ 0 & 0 & 0 & 1/21 & 0 \\ 0 & 0 & 0 & 0 & 1/9 \end{matrix}$$

b) Ecris une fonction Matlab « PDP » qui permet de décomposer une matrice carrée A en un produit de trois matrices de la même dimension que A et $A=P*D*P^{-1}$, tels que P^{-1} est l'inverse de la matrice P et D est une matrice diagonale. *(1.5pts)*

```
function [ P D ] = PDP(A)
[n,m]=size(A);
if n==m
    [P,D]=eig(A);
else
    disp('La matrice n''est pas carrée');
end
end
```

c) Ecris une fonction « LireMatTriSup » qui permet de lire une matrice triangulaire supérieure de dimension n. *(1.5pts)*

```
function [ A] = LireMatTriSup( n )
if n>0
    A=zeros(n,n);
    for i=1:n
        for j=i:n
            A(i,j)=input(sprintf('A[%d,%d]=',i,j));
        end
    end
else
    disp('Dimension de matrice non valable');
end
end
```

10

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Pour chaque commande, donne le résultat affiché par Matlab :

fliplr(A) ***(1pt)***

```
        15     8     1    24    17
        16    14     7     5    23
ans =   22    20    13     6     4
         3    21    19    12    10
         9     2    25    18    11
```

A\b ***(1pt)***

```
        1
        3
ans =   5
        7
        9
```

M=[A b] ***(1pt)***

```
      17    24     1     8    15   285
      23     5     7    14    16   315
M =    4     6    13    20    22   425
      10    12    19    21     3   315
      11    18    25     2     9   285
```

10

b) Ecris une fonction Matlab « FactorisationLU » qui décompose une matrice carrée de dimension n en un produit de 2 matrices selon le principe de la méthode de résolution directe LU. *(2pts)*

```
function [ L U ] = FactorisationLU( A )
n=size(A,1);
L=eye(n);
U=A;
for k=1:n-1
    L(k+1:n,k)= U(k+1:n,k)/U(k,k);
    for i=k+1:n
        U(i,:)=U(i,:)-L(i,k)*U(k,:);
    end
end
end
```

c) Ecris une fonction Matlab « Forward » qui permet de résoudre un système linéaire AX=b par descente (lorsque la matrice A est triangulaire inférieure).

(1pt)

```
function [ X ] = Forward( A ,b)
if tril(A)==A
    n=size(A,1);
    X=zeros(n,1);
    for i=1:n
        X(i)=b(i);
        for j=1:i-1
            X(i)=X(i)-A(i,j)*X(j);
        end
        X(i)=X(i)/A(i,i);
    end
else
    fprintf('LA MATRICE N''EST PAS TRIANGULAIRE SUPERIEURE!!!\n');
    X=[];
end
end
```

TP3- Ecrit

Nom : . . . .

Prénom : . . . .

Groupe : . . . .

Exercice 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Pour chaque commande, donne le résultat affiché par Matlab :

`A=flipud(A)` *(1pt)*

```
A =
    11    18    25     2     9
    10    12    19    21     3
     4     6    13    20    22
    23     5     7    14    16
    17    24     1     8    15
```

`A*X` *(1pt)*

```
ans =
   285
   315
   425
   315
   285
```

`round(ans-b)` *(1pt)*

```
ans =
     0
     0
     0
     0
     0
```

b) Ecris une fonction Matlab « Conditionnement » qui permet de calculer selon la valeur d'un paramètre K (K peut prendre la valeur 1, 2 , 'inf' ou 'fro') le conditionnement pour une matrice carré A de dimension n. L'utilisation de la fonction Matlab 'norm' est autorisée. *(2pts)*

```
function [ Cond ] = Conditionnement( A,k)
if size(A,1)==size(A,2)&& det(A)~=0
    InvA=inv(A);
    switch k
        case 1, Cond=norm(A,1)*norm(InvA,1);
        case 2, Cond=norm(A)*norm(InvA);
        case 'inf', Cond=norm(A,inf)*norm(InvA,inf);
        case 'fro', Cond=norm(A,'fro')*norm(InvA,'fro');
        otherwise error('valeur non accéptée')
    end
else
    disp('Matrice non valable');
end
end
```

c) Ecris une fonction Matlab « BackSubstitution » qui permet de résoudre un système linéaire AX=b par remontée (lorsque la matrice A est triangulaire supérieure). *(1pt)*

```
function [ X ] = BackSubstitution( A ,b)
if triu(A)==A
    n=size(A,1);
    X=zeros(n,1);
    for i=n:-1:1
        X(i)=b(i);
        for j=i+1:n
            X(i)=X(i)-A(i,j)*X(j);
        end
        X(i)=X(i)/A(i,i);
    end
else
    fprintf('LA MATRICE N''EST PAS TRIANGULAIRE SUPERIEURE!!!\n');
    X=[];
end
end
```

12

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Pour chaque commande, donne le résultat affiché par Matlab :

`B =flipud(fliplr(A))` *(1pt)*

```
        9     2    25    18    11
        3    21    19    12    10
B =    22    20    13     6     4
       16    14     7     5    23
       15     8     1    24    17
```

`C=B([1,3,4],[2,5,4])` *(1pt)*

```
        2    11    18
C =    20     4     6
       14    23     5
```

`C(end,end)=[]` *1pt)*

Erreur : on ne peut pas supprimer un seul élément sauf si la matrice ne contient déjà qu'un seul.

b) Ecris une fonction Matlab « GaussJordan » qui permet de résoudre un système linéaire Ax=b (A matrice carrée de dimension n et b un vecteur colonne de n élément) selon la méthode de Gauss-Jordan. *(1.5pts)*

```
function [ X ] = GaussJordan(A,b)
A=[A b];
n=size(A,1);
for k=1:n
    A(k,:)=A(k,:)/A(k,k);
    for i=1:n
        if i~=k
            A(i,:)=A(i,:)-A(i,k)*A(k,:);
        end
    end
end
X=A(:,end);
end
```

c) Ecris une fonction Matlab « MPG » qui permet de localiser dans une matrice carrée le plus grand élément par valeur absolue. Cette fonction retourne la valeur exacte de cet élément et sa position. *(1.5pts)*

```
function [ Valeur Ligne Colonne ] = MPG( A )
n=size(A,1);
Valeur=A(1,1);
Ligne=1;
Colonne=1;
for i=1:n
    for j=1:n
        if abs(val)<abs(A(i,j))
            Valeur=A(i,j);
            Ligne=i;
            Colonne=j;
        end
    end
end
end
```

13

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Pour chaque commande, donne le résultat affiché par Matlab :

A*ones(5,1) *(1pt)*

```
        65
        65
ans =   65
        65
        65
```

ans-sum(A,2) *(1pt)*

```
        0
        0
ans =   0
        0
        0
```

trace(A) *(1pt)*

```
ans =  65
```

b) Ecris une fonction Matlab « JacobiResolution » qui permet de résoudre un système linéaire Ax=b par la méthode de Jacobi (on suppose que la convergence est satisfaite). *(2pts)*

```
function [ X ] = JacobiResolution( A,b,Epsilon,MaxIteration,X0 )
DeltaX=Epsilon+1;
Iteration=1;
n=size(A,1);
X=zeros(n,1);
while Iteration<=MaxIteration && DeltaX>Epsilon
    for Line=1:n
        X(Line)=b(Line);
        for Col=1:n
            if Col~=Line
                X(Line)=X(Line)-A(Line,Col)*X0(Col);
            end
        end
        X(Line)=X(Line)/A(Line,Line);
    end
    DeltaX=max(abs(X-X0));
    X0=X;
    Iteration=Iteration+1;
end
disp(X)
end
```

c) Ecris une fonction Matlab « ReadPositiveNumber » qui oblige l'utilisateur de faire entrer un nombre strictement positif. *(1pt)*

```
function [ N ] = ReadPositiveNumber( )
N=-1;
while N<=0
    N=input(sprintf('Entrer un nombre positif '));
end
end
```

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Pour chaque commande, donne le résultat affiché par Matlab :

`diag(diag(A))*diag(X)` *(1^{pt})*

$$ans = \begin{matrix} 17 & 0 & 0 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 65 & 0 & 0 \\ 0 & 0 & 0 & 147 & 0 \\ 0 & 0 & 0 & 0 & 81 \end{matrix}$$

`eig(ans)` *(1^{pt})*

$$ans = \begin{matrix} 15 \\ 17 \\ 65 \\ 81 \\ 147 \end{matrix}$$

`A\b` *(1^{pt})*

$$ans = \begin{matrix} 1.0000 \\ 3.0000 \\ 5.0000 \\ 7.0000 \\ 9.0000 \end{matrix}$$

b) Ecris une fonction Matlab « GaussSeidelResolution » qui permet de résoudre un système linéaire Ax=b par la méthode de Gauss-Seidel (on suppose que la convergence est satisfaite). *(2pts)*

```
function [ X ] = GaussSeidelResolution(A,b,Epsilon,MaxIte,X0 )
DeltaX=Epsilon+1;
Iteration=1;
n=size(A,1);
X=zeros(n,1);
while Iteration<=MaxIt && DeltaX>Epsilon
    for Line=1:n
        X(Line)=b(Line);
        for Col=1:n
            if Col<Line
                X(Line)=X(Line)-A(Line,Col)*X(Col);
            else
                if Col>Line
                    X(Line)=X(Line)-A(Line,Col)*X0(Col);
                end
            end
        end
        X(Line)=X(Line)/A(Line,Line);
    end
    DeltaX=max(abs(X-X0));
    X0=X;
    Iteration=Iteration+1;
end
disp(X)
end
```

c) Ecris une fonction Matlab « Erreur » qui calcule l'erreur absolue et l'erreur relative entre les valeurs de deux vecteurs X et Xbar. *(1pt)*

```
function [ Delta Rho ] = Erreur( X,Xbar )
Delta=abs(X-Xbar);
Rho=abs(Delta./Xbar);
end
```

TP3- Ecrit

Prénom : . . . .

Nom : . . . .

Groupe : . . . .

EXERCICE 1 :

Soit A une matrice de 5x5 éléments et b et X deux vecteurs colonnes de 5 éléments. Les valeurs de ces variables sont données dans la figure :

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} \quad b = \begin{pmatrix} 285 \\ 315 \\ 425 \\ 315 \\ 285 \end{pmatrix}$$

*Sachant que A * X = b ;*

a) Pour chaque commande, donne le résultat affiché par Matlab :

`nnz(A)/numel(A)` ***(1pt)***

```
ans = 1
```

`diag(fliplr(A))` ***(1pt)***

```
       15
       14
ans =  13
       12
       11
```

`diag(ans)\eye(5)` ***(1pt)***

```
        0.0667       0        0        0        0
             0  0.0714        0        0        0
ans =        0       0   0.0769        0        0
             0       0        0   0.0833        0
             0       0        0        0   0.0909
```

ou

```
        1/15    0      0      0      0
        0       1/14   0      0      0
ans =   0       0      1/13   0      0
        0       0      0      1/12   0
        0       0      0      0      1/11
```

diag(ans) donne une matrice diagonale,; la commande globale calcule l'inverse de la matrice diagonale

b) Ecris une fonction Matlab « GaussSeidelResolution » qui permet de résoudre un système linéaire Ax=b par la méthode de Gauss-Seidel (on suppose que la convergence est satisfaite). *(2pts)*

```
function [ X ] = GaussSeidelResolution(A,b,Epsilon,MaxIte,X0 )
DeltaX=Epsilon+1;
Iteration=1;
n=size(A,1);
X=zeros(n,1);
while Iteration<=MaxIt && DeltaX>Epsilon
    for Line=1:n
        X(Line)=b(Line);
        for Col=1:n
            if Col<Line
                X(Line)=X(Line)-A(Line,Col)*X(Col);
            else
                if Col>Line
                    X(Line)=X(Line)-A(Line,Col)*X0(Col);
                end
            end
        end
        X(Line)=X(Line)/A(Line,Line);
    end
    DeltaX=max(abs(X-X0));
    X0=X;
    Iteration=Iteration+1;
end
disp(X)
end
```

c) Ecris une fonction Matlab « Erreur » qui calcule l'erreur absolue et l'erreur relative entre les valeurs de deux vecteurs X et Xbar. Le résultat doit être un vecteur V de la même taille. *(1pt)*

```
function [ Delta Rho ] = Erreur( X,Xbar )
if length(X)==length(Xbar)
    Delta=abs(X-Xbar);
    Rho=abs(Delta./Xbar);
else
    error('Les deux vecteurs doivent avoir la même longueur') ;
end
end
```