

TP1 : Méthodes de résolution de systèmes linéaires

Spécialité Master1 : G01: Commande Electrique + G02: Réseaux Electriques +  
G03: Energies Renouvelables

**But de TP :**

Durant ce TP, nous allons mettre en œuvre les algorithmes des méthodes de résolution des équations linéaires étudiées pendant le cours : la **Méthode de Jacobi** et la méthode de **Gauss Seidel**.

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ 2x_1 - x_2 + 3x_3 = 4 \\ 3x_1 + 2x_2 - 2x_3 = -2 \end{cases}$$

et cela pour une tolérance

$$\delta(i) = |x_i^{(k+1)} - x_i^{(k)}| < \varepsilon = 10^{-10}$$

Partie I: Méthode de Jacobi

1) Créer un fichier M-File nommé Jacobi et l'enregistrer.

```
function X=jacobi(A,B,P,delta,max1)
% Input -A is an N x N nonsingular matrix
% -B is an N x 1 matrix
% -P is an N x 1 matrix; the initial guess
% -delta is the tolerance for P
% -max1 is the maximum number of iterations
% Output-X is an N x 1 matrix : the Jacobi approximation to
% the solutions of AX= B
N= length(B);
for k=1:max1
for j=1:N
X(j)=(B(j)-A(j,[1:j-1,j+1:N])*P([1:j-1,j+1:N]))/A(j,j);
end
err=abs(norm(X'-P));
relerr=err/(norm(X)+eps);
P=X';
if (err<delta)|(relerr<delta)
break
end
end
X=X';
```

1) Créer un autre fichier contenant le programme ci-dessous puis l'exécuter:

```
A=[1 1 1;2 -1 3;3 2 -2];
B=[1 4 -2]';
P=[0 0 0]';
delta=10^-20
max1=10000
X=jacobi(A,B,P,delta,max1)% appel
```

2) Afficher la solution finale. Le programme converge-il? Justifier votre réponse.

Partie I: Méthode de Gauss-Seidel

1) Créer un fichier M-File nommé gseid et l'enregistrer.

```
function X=gseid(A,B,P,delta,max1)
% Input -A is an N x N nonsingular matrix
% -B is an N x 1 matrix
% -P is an N x 1 matrix; the initial guess
% -delta is the tolerance for P
% -max1 is the maximum number of iterations
% Output-X is an N x 1 matrix : the jacobi approximation to
% the solutions of AX= B
N= length(B);
for k=1:max1
for j=1:N
if j==1
X(1)=(B(1)-A(1,2:N)*P(2:N))/A(1,1);
elseif j==N
X(N)=(B(N)-A(N,1:N-1)*(X(1:N-1)))/A(N,N);
else
% X contains th kth approximations and P th (k-1)st
X(j)=(B(j)-A(j,1:j-1)*X(1:j-1)-A(j,j+1:N)*P(j+1:N))/A(j,j);
end
end
err=abs(norm(X'-P));
```

1) Créer un autre fichier contenant le programme ci-dessous puis l'exécuter:

```
A=[10 1 1;2 6 3;3 2 -12];
B=[1 4 -2]';
P=[1 1 10]';
delta=10^-5
max1=10
X=gseid(A,B,P,delta,max1)
```

Ce script ci-dessus permet de résoudre le système d'équation suivant

$$\begin{cases} 10x_1 + x_2 + x_3 = 1 \\ 2x_1 + 6x_2 + 3x_3 = 4 \\ 3x_1 + 2x_2 - 12x_3 = -2 \end{cases}$$

et cela pour une tolérance

$$\delta(i) = |x_i^{(k+1)} - x_i^{(k)}| < \varepsilon = 10^{-5}$$

- 2) Afficher la solution finale. Le programme converge-il? Justifier votre réponse.
- 3) Refaire en utilisant la méthode de Jacobi.
- 4) Comparer les résultats obtenus par les deux méthodes