

Analyse numérique matricielle

TP4

Vernet Elliot, Courtois Thibault, Imperatore Valentin

January 3, 2024

Dans ce TP, nous allons étudier la méthode d'itération de Jacobi, de Gauss-Seidel et la méthode de relaxation, qui sont trois méthodes itératives qui permettent de résoudre des systèmes linéaires de type $Ax = b$ de façon approximative.

Méthode itérative

Pour mettre en oeuvre une méthode itérative, on décompose la matrice A du système en $A = M - N$

On obtient alors :

$$Ax = b \iff (M - N)x = b \iff x = M^{-1}(Nx + b)$$

On définit alors une suite telle que :

$$x_0 \in R \text{ et } x_{k+1} = M^{-1}(Nx_k + b)$$

Cette suite x_k converge pour tout x_0 si le rayon spectral de la matrice M est strictement plus petit que 1. La convergence est alors d'autant plus rapide que le rayon est petit.

Méthode de Jacobi

On s'intéresse à la matrice :

$$A = \begin{pmatrix} 1 & a & 0 \\ a & 1 & a \\ 0 & a & 1 \end{pmatrix} \text{ et au vecteur } b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

La méthode consiste à décomposer la matrice A du système en la somme de deux matrices : sa diagonale et son triangle inférieure et supérieure.

On a $A = D-E-F$ avec :

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad E = \begin{pmatrix} 0 & 0 & 0 \\ -a & 0 & 0 \\ 0 & -a & 0 \end{pmatrix} \quad \text{et} \quad F = \begin{pmatrix} 0 & -a & 0 \\ 0 & 0 & -a \\ 0 & 0 & 0 \end{pmatrix}$$

On choisit $M = D$ et $N = E + F$ (voir section méthode itérative). On construit alors la suite :

$$x_{k+1} = M^{-1}(Nx_k + b) = D^{-1}((E + F)x_k + b) = (I_3 - D^{-1}A)x_k + D^{-1}b$$

On pose alors $J = (I_3 - D^{-1}A)$

On obtient alors :

$$x_{k+1} = Jx_k + D^{-1}b \text{ avec } J \text{ la matrice de Jacobi (matrice itérative du système).}$$

Pour la matrice A choisit, la matrice de Jacobi est :

$$\begin{pmatrix} 0 & -a & 0 \\ -a & 0 & -a \\ 0 & -a & 0 \end{pmatrix}$$

On calcul le rayon spectrale de la matrice J à l'aide de la fonction `eig` de Matlab, pour différente valeur de a .

On obtient, pour $a = 0.1$: $\text{Sp}(J) = -0.1414; 0; 0.1414$ donc $\rho(J) = 0.1414$

Pour $a = 0.7$: $\text{Sp}(J) = -0.9899; 0; 0.9899$ donc $\rho(J) = 0.9899$

Pour $a = 2$: $\text{Sp}(J) = -2.8284; 0; 2.8284$ donc $\rho(J) = 2.8284$

On effectue dans un premier temps 10 itérations avec la méthode de Jacobi pour les 3 valeurs de a différentes, et on observe les différences de précisions.

On utilise le programme Matlab suivant :

```

clc
clear all
a = 2;
A = [1 a 0; a 1 a; 0 a 1];
b = [1; 2; 3];
J = [1 0 0; 0 1 0; 0 0 1] - A;
x = zeros(3, 1);
for i = 1:10
    x = J*x+b
    norm(A*x-b, 2)
end

```

On obtient les résultats suivants :

a	0.1	0.7	2
x^{10}	$\begin{pmatrix} 0.8367 \\ 1.6327 \\ 2.8367 \end{pmatrix}$	$\begin{pmatrix} 1.8824 \\ -3.8432 \\ 3.8824 \end{pmatrix}$	$\begin{pmatrix} -9363 \\ -28086 \\ -9361 \end{pmatrix}$
erreur	1.1085e-08	3.1313	1.135e05

On observe que la solution converge pour $a = 0.1$ et $a = 0.7$ (plus lentement) et qu'elle diverge dans le cas $a = 2$. Ceci est normale car le rayon spectrale doit être plus petit que 1 pour que la méthode converge et plus ce rayon est petit, plus la convergence est rapide.

Les résultats trouvés sont donc cohérents.

On veut maintenant observer le nombre d'itérations nécessaires dans chacun des cas pour arriver à une précision de $10e-06$.
 Pour cela, on utilise le programme suivant :

```

clc
a = 2;
A = [1 a 0; a 1 a; 0 a 1];
b = [1; 2; 3];
J = [1 0 0; 0 1 0; 0 0 1] - A

x=zeros(3,1);
n=0;
eps = 1;
while eps > 10^-6
    eps = norm(A*x -b, 2);
    x = J*x + b;
    n = n + 1;
end
n
eps
x
    
```

On obtient les résultats suivants :

	a = 0,1	a = 0,7	a = 2
Nombre d'itérations avec la méthode Jacobi	9	1492	685

On trouve un nombre d'itération inférieure pour $a = 2$ que pour $a = 0.7$ mais aucune solution n'est trouvée. **Les résultats sont donc cohérents.**

Méthode de Gauss-Seidel

On va effectuer les mêmes observation que pour la méthode de Jacobi avec la même matrice A. Le changement dans la méthode de Gauss-seidel est la matrice d'itération. En effet, on prend ici : $M = D - E$ et $N = F$.

On obtient alors :

$$x_{k+1} = M^{-1}(Nx_k + b) \iff x_{k+1} = (D - E)^{-1}Fx_k + (D - E)^{-1}b$$

La matrice d'itération est : $L = (D - E)^{-1}F$

On effectue 10 itérations avec le programme suivant :

```
clc
a = 0.1;
x = zeros(3,1);
A = [1 a 0; a 1 a; 0 a 1];
E = [ 0 0 0; -a 0 0; 0 -a 0];
D = [1 0 0; 0 1 0; 0 0 1];
F = [ 0 -a 0; 0 0 -a; 0 0 0];
for i = 1:10
    x = inv(D-E)*F*x + inv(D-E)*b;
    eps = norm(A*x-b);
end
x
eps
```

On obtient les résultats suivants :

a	0.1	0.7	2
x^{10}	$\begin{pmatrix} 0.8367 \\ 1.6327 \\ 2.8367 \end{pmatrix}$	$\begin{pmatrix} 4.044 \\ -5.5662 \\ 6.8694 \end{pmatrix}$	$\begin{pmatrix} 28760941 \\ -115043766 \\ 230087535 \end{pmatrix}$
erreur	8.0059e-16	0.6005	4.5018e08

Ainsi, les résultats sont plus précis dans les cas $a = 0.1$ et $a = 0.7$ mais le résultat pour $a = 2$ diverge plus rapidement qu'avec la méthode de Jacobi.

On utilise le programme suivant :

```
clc
a = 0.1;
x = zeros(3,1);
A = [1 a 0; a 1 a; 0 a 1];
E = [ 0 0 0; -a 0 0; 0 -a 0];
D = [1 0 0; 0 1 0; 0 0 1];
F = [ 0 -a 0; 0 0 -a; 0 0 0];
eps = 1;
n = 0;
while eps > 10^-6
    x = inv(D-E)*F*x + inv(D-E)*b;
    eps = norm(A*x-b);
    n = n+1;
end
x
eps
n
```

On obtient les résultats suivants :

	a = 0,1	a = 0,7	a = 2
Nombre d'itérations avec la méthode Jacobi	9	1492	685
Nombre d'itérations avec la méthode Gauss-Seidel	5	669	343

Les mêmes observations que pour la méthode de Jacobi peuvent être faite. Cependant, la méthode converge plus rapidement.

Il semble donc plus intéressant d'utiliser cette méthode de résolution itérative de système linéaires.

Méthode de relaxation

On définit la méthode de relaxation par :

$$\left(\frac{D}{w} - E\right)x_{k+1} = \left(\frac{1-w}{w}D + F\right)x_k + b \text{ avec } w \in R$$

$$\text{Soit : } x_{k+1} = \left(\frac{D}{w} - E\right)^{-1}\left(\frac{1-w}{w}D + F\right)x_k + \left(\frac{D}{w} - E\right)^{-1}b$$

On a donc une nouvelle matrice d'itération: $L_w = \left(\frac{D}{w} - E\right)^{-1}\left(\frac{1-w}{w}D + F\right)$

On effectue des essais sur les 3 valeurs de a différentes pour différentes valeurs de w avec le code suivant :

```
clc
clear all;
a = 0.1;
w = 1;
A = [1 a 0; a 1 a; 0 a 1];
b=[1; 2; 3];
E = [0 0 0; -a 0 0; 0 -a 0];
F = [0 -a 0; 0 0 -a; 0 0 0];
D = [1 0 0; 0 1 0; 0 0 1];

x=zeros(3,1);
eps = norm(A*x-b,2);
n=0;
while eps > 10^-6
    x=inv(D/w - E)*(((1-w)/w)*D+F)*x + inv(D/w-E)*b;
    eps = norm(A*x-b,2);
    n=n+1;
end
n
```

On trouve alors les résultats suivants :

	a = 0,1	a = 0,7
w = 0,1	148	12555
w = 0,3	45	3747
w = 0,5	24	1986
w = 0,7	14	1233
w = 0,9	8	815
w = 1 (revient à la méthode de Gauss-Seidel)	5	669
w = 3	1023	537
w = 5	511	259
w = 10	323	163
w = 50	182	92
w = 100	154	77
w = 1000	72	52
w = 10000	49	39
w = 100000	37	31

On observe que les résultats convergent pour $w \in]0, 2[$ dans les cas $a = 0.1$ et $a = 0.7$.

Autrement, les valeurs trouvées pour le vecteur solution x sont absurdes ou sont des erreurs.

On peut donc empiriquement dire que cette **méthode converge pour tout** $w \in]0, 2[$.

Elle converge plus rapidement que les deux autres méthodes pour un certain $w = \frac{2}{1 + \sqrt{1 - \rho(J)^2}}$ avec J la matrice d'itération de la méthode de Jacobi.