# Université Gaston Berger de Saint-Louis

# Résolution numérique de systèmes linéaires
# Méthodes itératives

**Pr. Ousmane THIARE**

http://www.ousmanethiare.com

April 16, 2020

# Contenu

# Contenu

# Suite de vecteurs et de matrices

## Définition 4.1

Soit V un espace vectoriel muni d'une norme $\|\cdot\|$, on dit qu'une suite $(v_k)$ d'éléments de V converge vers un élément $v \in V$ , si

$$\lim_{k\to\infty} \|v_k - v\| = 0$$

et on écrit

$$v = \lim_{k\to\infty} v_k$$

## Remarque 4.1

Sur un espace vectoriel de dimension finie, toutes les normes sont équivalentes. Donc $v_k$ tend vers v si et seulement si $\|v_k - v\|$ tend vers 0 pour une norme.

# Suite de vecteurs et de matrices

## Théorème 4.1

1. Soit $\| \cdot \|$ une norme matricielle subordonnée, et $\mathbb{B}$ une matrice vérifiant
$$\|\mathbb{B}\| < 1$$
Alors la matrice $(\mathbb{I} + \mathbb{B})$ est inversible, et
$$\|(\mathbb{I} + \mathbb{B})^{-1}\| \leq \frac{1}{1 - \|\mathbb{B}\|}$$
2. Si une matrice de la forme $(\mathbb{I} + \mathbb{B})$ est singulière, alors nécessairement
$$\|\mathbb{B}\| \geq 1$$
pour toute norme matricielle, subordonnée ou non.

# Suite de vecteurs et de matrices

## Théorème 4.2

Soit $\mathbb{B}$ une matrice carrée. Les conditions suivantes sont équivalentes :

1. $\lim\limits_{k\to\infty} \mathbb{B}^k = 0$
2. $\lim\limits_{k\to\infty} \mathbb{B}^k v = 0$ pour tout vecteur v
3. $\rho(\mathbb{B}) < 1$
4. $\|\mathbb{B}\| < 1$ pour au moins une norme matricielle subordonnée $\|\cdot\|$

## Théorème 4.3

Soit $\mathbb{B}$ une matrice carrée, et $\|\cdot\|$ une norme matricielle quelconque. Alors

$$\lim_{k\to\infty} \|\mathbb{B}^k\|^{\frac{1}{k}} = \rho(\mathbb{B})$$

# Contenu

# Méthode de Jacobi, Gauss-Seidel, S.O.R.

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ une matrice régulière et $\mathbf{b} \in \mathbb{K}^n$. Il s'agit de résoudre le système $\mathbb{A}\mathbf{x} = \mathbf{b}$ par une méthode itérative, c'est-à-dire de créer une suite $\mathbf{x}^k$ qui converge vers $\mathbf{x}$. On note $\mathbb{D} = diag(\mathbb{A}), \mathbb{E}$ la matrice triangulaire inférieure vérifiant

$$\left\{ \begin{array}{ccll} e_{ij} & = & 0, & i \leq j \\ e_{ij} & = & -a_{ij}, & i > j \end{array} \right.$$

et $\mathbb{F}$ la matrice triangulaire supérieure vérifiant

$$\left\{ \begin{array}{ccll} f_{ij} & = & 0, & i \geq j \\ f_{ij} & = & -a_{ij}, & i > j \end{array} \right.$$

On a alors

$$\mathbb{A} = \begin{pmatrix} \ddots & & -\mathbb{F} \\ & \mathbb{D} & \\ -\mathbb{E} & & \ddots \end{pmatrix} = \mathbb{D} - \mathbb{E} - \mathbb{F}$$

# Méthode de Jacobi, Gauss-Seidel, S.O.R.

## Méthode de Jacobi

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k)} \right) \forall i \in \{1, \cdots, n\}$$

## Méthode de Gauss-Seidel

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{n} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \forall i \in \{1, \cdots, n\}$$

## Méthode de relaxation

$$x_i^{(k+1)} = \omega \hat{x}_i^{(k+1)} + (1-\omega) x_i^{(k)}$$

où $\hat{x}_i^{(k+1)}$ est obtenu à partir de $x^{(k)}$ par l'une des deux méthodes précédentes.

# Méthode de Jacobi, Gauss-Seidel, S.O.R.

Avec la méthode de Jacobi

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left( b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k)} \right) + (1-\omega) x_i^{(k)} \forall i \in \{1, \cdots, n\}$$

Avec la méthode de Gauss-Seidel

$$x_i^{(k+1)} = \frac{w}{a_{ii}} \left( b_i - \sum_{j=1}^{n} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) + (1-\omega) x_i^{(k)} \forall i \in \{1, \cdots ,$$

Cette méthode de relaxation est appelée méthode S.O.R. (successive over relaxation) Toutes ces méthodes se mettent sous la forme

$$Mx^{k+1} = Nx^{k} + b$$

# Méthode de Jacobi, Gauss-Seidel, S.O.R.

avec

| | | |
|---|---|---|
| Jacobi | $M = D$ | $N = E + F$ |
| Gauss-Seidel | $M = D - E$ | $N = F$ |
| SOR | $M = \frac{1}{\omega}D - E$ | $N = \frac{1-\omega}{\omega}D + F$ |

Programmation d'une étape de l'algorithme de Jacobi :

```
Pour i=1:N
   S:=B(i)
      Pour j=1:I-1
      S=S-A(i,j)*X(j)
      Pour j=i+1:N
      S=S-A(i,j)*X(j)
   Y(i)=S/A(i,i)
Pour i=1:N
   X(i):=Y(i)
```

<u>Test d'arrêt</u> : on définit le résidu à l'étape k comme $r^{(k)} = b - Ax^{(k)}$.
Le test s'écrit : tant que $\|r^{(k)}\| > eps$, on continue.

# Contenu

# Résultats généraux de convergence

Soit donc l'algorithme

$$Mx^{k+1} = Nx^k + b$$

avec M-N=A. Si la suite converge, elle converge vers la solution x de Ax=b, et l'erreur $e^{(k)} = x^{(k)} - x$ est solution de $Me^{(k+1)} = Ne^{(k)}$. On note $B = M^{-1}N$. D'après le théorème 4.2, on a

**Théorème 4.4**

La suite $x^{(k)}$ converge pour toute donnée initiale $x^0$ si et seulement si $\rho(\mathbb{B}) < 1$, si et seulement si $\|\mathbb{B}\| < 1$ pour au moins une norme matricielle subordonnée $\|\cdot\|$.

# Résultats généraux de convergence

Il est d'usage d'affecter les noms suivants aux matrices des méthodes précédentes

| | |
|---|---|
| Jacobi | $J = D^{-1}(E+F)$ |
| SOR | $\mathcal{L}_\omega = (\frac{1}{\omega}D - E)^{-1}(\frac{1-\omega}{\omega}D + F)$ |

**Lemme 4.1**

Pour tout $\omega \neq 0$, on a $\rho(\mathcal{L}_\omega) \geq |\omega - 1|$.

# Résultats généraux de convergence

On en déduit par le théorème 4.4,

**Théorème 4.5**

Si la méthode de relaxation converge pour toute donnée initiale, on a

$$0 < \omega < 2$$

On définit le taux de convergence asymptotique par

$$R(B) = -ln\rho(B)$$

**Théorème 4.6**

Le nombre d'itérations nécessaires pour réduire l'erreur d'un facteur $\epsilon$ est au moins égal à $K = \dfrac{-ln\epsilon}{R(B)}$

# Contenu

# Cas des matrices hermitiennes

## Théorème 4.7

Soit A une matrice hermitienne définie positive, A=M-N , où M est inversible. Si $M+N^*$ (qui est toujours hermitienne), est définie positive, la méthode itérative converge pour toute donnée initiale.

## Corollaire 4.1

Soit A une matrice hermitienne définie positive. Si $\omega \in ]0,2[$, la méthode de relaxation converge pour toute donnée initiale.

# Contenu

# Cas des matrices tridiagonales

## Théorème 4.8

Soit A une matrice tridiagonale. Alors $\rho(\mathcal{L}_1) = (\rho(J))^2$ : les méthodes de Jacobi et Gauss-Seidel convergent ou divergent simultanément. Si elles convergent, la méthode de Gauss-Seidel est la plus rapide.

## Théorème 4.9

Soit A une matrice tridiagonale telles que les valeurs propres de J soient réelles. Alors les méthodes de Jacobi et de relaxation convergent ou divergent simultanément pour $\omega \in ]0, 2[$. Si elles convergent, la fonction $\omega \mapsto \rho(\mathcal{L}_\omega)$ a l'allure suivante : avec $\omega^* = \dfrac{2}{1 + \sqrt{1 - (\rho(J))^2}}$

# Cas des matrices tridiagonales

## Remarque 4.2

On ne connaît pas précisément ce $\omega^*$ si on ne connaît pas $\rho(J)$. Dans ce cas, le graphe ci-dessus montre que qu'il vaut mieux choisir $\omega$ trop grand que trop petit.

# Contenu

# Matrices à diagonale dominante

## Théorème 4.10

Soit A une matrice à diagonale strictement dominante ou irréductible à diagonale dominante. Alors la méthode de Jacobi converge.

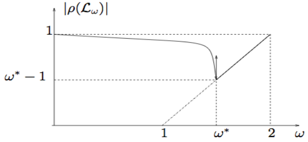


Figure: variation de $\mathcal{L}_\omega$ en fonction de $\omega$

## Théorème 4.11

Soit A une matrice à diagonale strictement dominante ou irréductible à diagonale dominante. Si $0 < \omega \leq 1$, la méthode de relaxation converge.

# Contenu

# La matrices du laplacien

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

On a

$$\rho(J) = 1 - \frac{\pi^2}{2n^2} + \mathcal{O}(n^{-4}), \rho(\mathcal{L}_1) = 1 - \frac{\pi^2}{n^2} + \mathcal{O}(n^{-4}),$$

$$\omega^* = 2(1 - \frac{\pi}{n} + \mathcal{O}(n^{-2})), \rho(\mathcal{L}_{\omega^*}) = \omega^* - 1 == 1 - \frac{2\pi}{n} + \mathcal{O}(n^{-2}).$$

# La matrices du laplacien

Pour n=100, pour obtenir une erreur de $\epsilon = 10^{-1}$, on doit faire

- 9342 itérations de l'algorithme de Jacobi,
- 4671 itérations de l'algorithme de Gauss-Seidel,
- 75 itérations de l'algorithme de l'algorithme de relaxation optimale.

# Contenu

# Complexité

Supposons la matrice A pleine. La complexité d'une itération est d'environ $2n^2$. Si l'on fait au moins n itérations, on a donc une complexité totale de $2n^3$, à comparer aux $2n^3/3$ de la méthode de Gauss.

*Pour résoudre un système linéaire, on préférera les méthodes directes dans le cas des matrices pleines, et les méthodes itératives dans le cas des matrices creuses.*