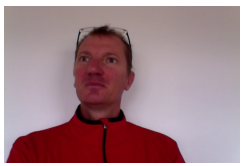
# Systèmes linéaires : Méthodes itératives et relaxation

*Exposé par Jérôme MONNIER*



*Professeur à l'INSA de Toulouse*
*Département de Génie Mathématique et Modélisation*

***

**Plan**

- Méthodes de décomposition (splitting),
- Méthodes de Jacobi et de Gauss-Siedel,
- Méthode de relaxation (SOR),
- Méthode de relaxation pour des matrices définies par blocs.

# Méthodes itératives

## Principe

▷ *Objectif*

Résoudre le système linéaire $Ax = b$ via méthode *itérative*.

Cas visé : $n >>$ et $A$ *creuse*.

# Méthodes itératives

## Principe

▷ *Objectif*

Résoudre le système linéaire $Ax = b$ via méthode *itérative*.

Cas visé : $n >>$ et $A$ *creuse*.

▷ *Principe des méthodes itératives*

Construction d'une suite $(x^{(n)})_{n\geq 0}$ tq $\lim_{n\to\infty} x^{(n)} = x^*$.

où $x^*$ : l'unique solution du système.

# Méthodes itératives

## Principe

▷ *Objectif*

Résoudre le système linéaire $Ax = b$ via méthode *itérative*.
Cas visé : $n >>$ et $A$ *creuse*.

▷ *Principe des méthodes itératives*

Construction d'une suite $(x^{(n)})_{n\geq 0}$ tq $\lim_{n\to\infty} x^{(n)} = x^*$.
où $x^*$ : l'unique solution du système.

▷ *Tests d'arrêt*. $\|x^{(n)} - x^*\| \leq \varepsilon$. Problème : $x^*$ inconnu...

# Méthodes itératives

## Principe

▷ *Objectif*

Résoudre le système linéaire $Ax = b$ via méthode *itérative*.
Cas visé : $n >>$ et $A$ *creuse*.

▷ *Principe des méthodes itératives*

Construction d'une suite $(x^{(n)})_{n\geq 0}$ tq $\lim_{n\to\infty} x^{(n)} = x^*$.
où $x^*$ : l'unique solution du système.

▷ *Tests d'arrêt*. $\|x^{(n)} - x^*\| \leq \varepsilon$. Problème : $x^*$ inconnu...

→ Test de stationnarité :

$$\frac{\|x^{(n)} - x^{(n-1)}\|}{\|x^{(n)}\|} \leq \varepsilon$$

# Méthodes itératives

## Principe

▷ *Objectif*

Résoudre le système linéaire $Ax = b$ via méthode *itérative*.
Cas visé : $n >>$ et $A$ *creuse*.

▷ *Principe des méthodes itératives*

Construction d'une suite $(x^{(n)})_{n\geq 0}$ tq $\lim_{n\to\infty} x^{(n)} = x^*$.
où $x^*$ : l'unique solution du système.

▷ *Tests d'arrêt*. $\|x^{(n)} - x^*\| \leq \varepsilon$. Problème : $x^*$ inconnu...

→ Test de stationnarité :

$$\frac{\|x^{(n)} - x^{(n-1)}\|}{\|x^{(n)}\|} \leq \varepsilon$$

▷ Algorithmes étudiés : Jacobi, Gauss-Siedel,
puis Relaxation - SOR (Successive Over Relaxation).

# Méthodes de décomposition

## Principe

▷ On décompose $A$ sous la forme $A = (M - N)$, d'où l'équation :

$$Mx = Nx + b$$

# Méthodes de décomposition

## Principe

▷ On décompose $A$ sous la forme $A = (M - N)$, d'où l'équation :

$$Mx = Nx + b$$

▷ Algorithme

$$\boxed{x_0 \text{ donné }, Mx^{(n+1)} = Nx^{(n)} + b}$$

# Méthodes de décomposition

## Principe

▷ On décompose $A$ sous la forme $A = (M - N)$, d'où l'équation :

$$Mx = Nx + b$$

▷ Algorithme

$$\boxed{x_0 \text{ donné }, Mx^{(n+1)} = Nx^{(n)} + b}$$

▷ Propriétés requises

- $M$ peu cher à "inverser" à chaque itération,
- $M$ tel que l'algorithme CV.

# Méthodes de décomposition

## Principe

▷ On décompose $A$ sous la forme $A = (M - N)$, d'où l'équation :

$$Mx = Nx + b$$

▷ Algorithme

$$\boxed{x_0 \text{ donné }, Mx^{(n+1)} = Nx^{(n)} + b}$$

▷ Propriétés requises

- $M$ peu cher à "inverser" à chaque itération,
- $M$ tel que l'algorithme CV.

▷ *Idée de décomposition ("splitting") classique en analyse numérique.*

# Méthodes de décomposition

## Convergence

### Proposition

*Les 4 propriétés suivantes sont équivalentes :*

*i)* $\lim_{n\to\infty} \|Mx\| = 0, \forall x \in \mathbb{R}^n,$

*ii)* $\lim_{n\to\infty} \|M\| = 0,$

*iii) Le rayon spectral de* $M$, $\rho(M) = \max_k \|\lambda_k(M)| < 1,$

*iv)* $(I - M)$ *est inversible et* $(I - M)^{-1} = \sum_{k=0}^{\infty} M^k.$

# Méthodes de décomposition

## Convergence

### Proposition

*Les 4 propriétés suivantes sont équivalentes :*

*i)* $\lim_{n\to\infty} \|Mx\| = 0, \forall x \in \mathbb{R}^n$,

*ii)* $\lim_{n\to\infty} \|M\| = 0$,

*iii) Le rayon spectral de* $M$, $\rho(M) = \max_k \|\lambda_k(M)| < 1$,

*iv)* $(I - M)$ *est inversible et* $(I - M)^{-1} = \sum_{k=0}^{\infty} M^k$.

▷ **Application à la décomposition** $M - N$,

Equation : $Mx = Nx + b$.

En passant à la limite dans l'équation, on obtient :

$$(I - M^{-1}N)x^{(\infty)} = M^{-1}b$$

# Méthodes de décomposition

## Convergence

### Proposition

*Les 4 propriétés suivantes sont équivalentes :*

*i)* $\lim_{n\to\infty} \|Mx\| = 0, \forall x \in \mathbb{R}^n$,

*ii)* $\lim_{n\to\infty} \|M\| = 0$,

*iii) Le rayon spectral de* $M$, $\rho(M) = \max_k \|\lambda_k(M)| < 1$,

*iv)* $(I - M)$ *est inversible et* $(I - M)^{-1} = \sum_{k=0}^{\infty} M^k$.

▷ **Application à la décomposition** $M - N$,

Equation : $Mx = Nx + b$.

En passant à la limite dans l'équation, on obtient :

$$(I - M^{-1}N)x^{(\infty)} = M^{-1}b$$

$\Rightarrow$ C.N.S. de l'algorithme : $\rho(M^{-1}N) < 1$.

# Décomposition $D - E - F$

## Méthode de Jacobi

▷ Base de toutes les décompositions à venir : $\boxed{A = D - E - F}$
avec $D$ diag. de $A$, $(-E)$ partie inf. stricte, $(-F)$ partie sup. stricte.

$$A = \begin{bmatrix} D & -F \\ -E & D \end{bmatrix}$$

# Décomposition $D - E - F$

## Méthode de Jacobi

▷ Base de toutes les décompositions à venir : $\boxed{A = D - E - F}$
avec $D$ diag. de $A$, $(-E)$ partie inf. stricte, $(-F)$ partie sup. stricte.

$$A = \begin{bmatrix} D & -F \\ -E & D \end{bmatrix}$$

▷ **Méthode de Jacobi** : $M = D$, $N = E + F$.
$\Rightarrow$ Itérations : $Dx^{(n+1)} = (E + F)x^{(n)} + b$

$$\boxed{x_i^{(n+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j \neq i} a_{ij} x_j^{(n)}) \ , \quad a_{ii} \neq 0}$$

# Décomposition $D - E - F$

## Méthode de Jacobi

▷ Base de toutes les décompositions à venir : $\boxed{A = D - E - F}$
avec $D$ diag. de $A$, $(-E)$ partie inf. stricte, $(-F)$ partie sup. stricte.

$$A = \begin{bmatrix} D & -F \\ -E & D \end{bmatrix}$$

▷ **Méthode de Jacobi** : $M = D$, $N = E + F$.
$\Rightarrow$ Itérations : $Dx^{(n+1)} = (E + F)x^{(n)} + b$

$$\boxed{x_i^{(n+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j \neq i} a_{ij} x_j^{(n)}) \ , \quad a_{ii} \neq 0}$$

$\oplus$ : Méthode triviale à implémenter, pas de système à inverser à chaque itération.

# Décomposition $D - E - F$

## Méthode de Jacobi

▷ Base de toutes les décompositions à venir : $\boxed{A = D - E - F}$
avec $D$ diag. de $A$, $(-E)$ partie inf. stricte, $(-F)$ partie sup. stricte.

$$A = \begin{bmatrix} D & -F \\ -E & D \end{bmatrix}$$

▷ **Méthode de Jacobi** : $M = D$, $N = E + F$.
$\Rightarrow$ Itérations : $Dx^{(n+1)} = (E + F)x^{(n)} + b$

$$\boxed{x_i^{(n+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j \neq i} a_{ij} x_j^{(n)}) \ , \quad a_{ii} \neq 0}$$

$\oplus$ : Méthode triviale à implémenter, pas de système à inverser à chaque itération.
$\ominus$ : CV très, trop, lente...

# Méthode de Gauss-Siedel

▷ Semblable à Jacobi mais avec les valeurs de $x_j$ fraîchement calculées :

$$x_i^{(n+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(n)})$$

# Méthode de Gauss-Siedel

▷ Semblable à Jacobi mais avec les valeurs de $x_j$ fraîchement calculées :

$$\boxed{x_i^{(n+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(n)})}$$

$\Leftrightarrow M = D - E$ et $N = F$. Soit :

$$(D - E)x^{(n+1)} = Fx^{(n)} + b$$

# Méthode de Gauss-Siedel

▷ Semblable à Jacobi mais avec les valeurs de $x_j$ fraîchement calculées :

$$\boxed{x_i^{(n+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(n)})}$$

$\Leftrightarrow M = D - E$ et $N = F$. Soit :

$$(D - E)x^{(n+1)} = Fx^{(n)} + b$$

$\oplus$ : Converge plus vite que Jacobi ; et itérations toujours pas chères : $\mathcal{O}(cn)$, où $c$ nombre moyen d'éléments non nuls par lignes.

# Méthode de Gauss-Siedel

▷ Semblable à Jacobi mais avec les valeurs de $x_j$ fraîchement calculées :

$$\boxed{x_i^{(n+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(n)})}$$

$\Leftrightarrow M = D - E$ et $N = F$. Soit :

$$(D - E)x^{(n+1)} = Fx^{(n)} + b$$

$\oplus$ : Converge plus vite que Jacobi ; et itérations toujours pas chères : $\mathcal{O}(cn)$, où $c$ nombre moyen d'éléments non nuls par lignes.

$\ominus$ : Vitesse de CV non optimale...

# Méthode de relaxation

## L'algorithme

▷ Généralisation de l'idée précédente :

On introduit un "équilibre" de la décomposition diagonale entre la partie explicite et la partie implicite :

$$M = \frac{1}{\omega} D - E \text{ et } N = \frac{1-\omega}{\omega} D + F$$

# Méthode de relaxation

## L'algorithme

▷ Généralisation de l'idée précédente :

On introduit un "équilibre" de la décomposition diagonale entre la partie explicite et la partie implicite :

$$M = \frac{1}{\omega}D - E \text{ et } N = \frac{1-\omega}{\omega}D + F$$

▷ Algorithme de relaxation :

$$\boxed{(\frac{1}{\omega}D - E)x^{(n+1)} = (\frac{1-\omega}{\omega}D + F)x^{(n)} + b}$$

$$x_i^{(n+1)} = \frac{\omega}{a_{ii}}[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(n)}] + (1-\omega)x_i^{(n)}$$

# Méthode de relaxation

## L'algorithme

▷ Généralisation de l'idée précédente :

On introduit un "équilibre" de la décomposition diagonale entre la partie explicite et la partie implicite :

$$M = \frac{1}{\omega}D - E \text{ et } N = \frac{1-\omega}{\omega}D + F$$

▷ Algorithme de relaxation :

$$\boxed{(\frac{1}{\omega}D - E)x^{(n+1)} = (\frac{1-\omega}{\omega}D + F)x^{(n)} + b}$$

$$x_i^{(n+1)} = \frac{\omega}{a_{ii}}[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(n)}] + (1-\omega)x_i^{(n)}$$

▷ Gauss-Siedel = relaxation avec $\omega = 1$.

# Méthode de relaxation

## L'algorithme

▷ Généralisation de l'idée précédente :

On introduit un "équilibre" de la décomposition diagonale entre la partie explicite et la partie implicite :

$$M = \frac{1}{\omega}D - E \text{ et } N = \frac{1-\omega}{\omega}D + F$$

▷ Algorithme de relaxation :

$$\boxed{(\frac{1}{\omega}D - E)x^{(n+1)} = (\frac{1-\omega}{\omega}D + F)x^{(n)} + b}$$

$$x_i^{(n+1)} = \frac{\omega}{a_{ii}}[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(n)}] + (1-\omega)x_i^{(n)}$$

▷ Gauss-Siedel = relaxation avec $\omega = 1$.

**Exercice**. Vérifier que si l'algorithme SOR converge alors on obtient bien la solution recherchée.

# Méthode de relaxation

## Convergence

▷ Coût par itération de l'algorithme : idem, $\mathcal{O}(cn)$.

# Méthode de relaxation

## Convergence

▷ Coût par itération de l'algorithme : idem, $\mathcal{O}(cn)$.

▷ Intérêt de cette "relaxation" ?

→ Tenter d'accélerer l'algorithme... Et on y arrive !...

# Méthode de relaxation

## Convergence

▷ Coût par itération de l'algorithme : idem, $\mathcal{O}(cn)$.
▷ Intérêt de cette "relaxation" ?
$\rightarrow$ Tenter d'accélerer l'algorithme... Et on y arrive !...

▷ Comment choisir $\omega$ ?

# Méthode de relaxation

## Convergence

▷ Coût par itération de l'algorithme : idem, $\mathcal{O}(cn)$.
▷ Intérêt de cette "relaxation" ?
$\rightarrow$ Tenter d'accélerer l'algorithme... Et on y arrive !...

▷ Comment choisir $\omega$ ?

### Théorème

- *Pour A carrée "quelconque", C. Nécessaire de CV :* $\omega \in ]0, 2[$

# Méthode de relaxation

## Convergence

▷ Coût par itération de l'algorithme : idem, $\mathcal{O}(cn)$.
▷ Intérêt de cette "relaxation" ?
→ Tenter d'accélerer l'algorithme... Et on y arrive !...

▷ Comment choisir $\omega$ ?

### Théorème

- *Pour A carrée "quelconque", C. Nécessaire de CV :* $\omega \in ]0, 2[$
- *Cas A sym. déf. positive, cette condition est Suffisante.*
  *Aussi, dans ce cas,* $\omega^{opt}$ *est connu.*

# Méthode de relaxation

## Convergence

▷ Coût par itération de l'algorithme : idem, $\mathcal{O}(cn)$.
▷ Intérêt de cette "relaxation" ?
$\rightarrow$ Tenter d'accélerer l'algorithme... Et on y arrive !...

▷ Comment choisir $\omega$ ?

### Théorème

- *Pour A carrée "quelconque", C. Nécessaire de CV :* $\omega \in ]0, 2[$
- *Cas A sym. déf. positive, cette condition est Suffisante.*
  *Aussi, dans ce cas,* $\omega^{opt}$ *est connu.*
- *Cas A à diagonale dominante, condition Suffisante :* $\omega \in ]0, 1]$.

# Méthode de relaxation

## Convergence

▷ Coût par itération de l'algorithme : idem, $\mathcal{O}(cn)$.
▷ Intérêt de cette "relaxation" ?
→ Tenter d'accélerer l'algorithme... Et on y arrive !...

▷ Comment choisir $\omega$ ?

### Théorème

- *Pour A carrée "quelconque", C. Nécessaire de CV :* $\omega \in ]0, 2[$
- *Cas A sym. déf. positive, cette condition est Suffisante.*
  *Aussi, dans ce cas,* $\omega^{opt}$ *est connu.*
- *Cas A à diagonale dominante, condition Suffisante :* $\omega \in ]0, 1]$.

Rem. Successive *Over Relaxation* car $\omega^{opt} \in ]1, 2[$.

# Méthode de relaxation

## Cas de matrices définies par blocs

▷ Soit $A$ de la forme suivante :

# Méthode de relaxation

## Cas de matrices définies par blocs

▷ Soit $A$ de la forme suivante :


→ Forme classique, modèles numériques d'EDP sur maillages structurés.

# Méthode de relaxation

## Cas de matrices définies par blocs

▷ Soit $A$ de la forme suivante :


→ Forme classique, modèles numériques d'EDP sur maillages structurés.

▷ Méthode de relaxation par blocs :

$$A_{ii}x_i^{(n+1)} = \omega(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(n)}) + (1-\omega)A_{ii}x_i^{(n)}$$

# Méthode de relaxation

## Cas de matrices définies par blocs

▷ Soit $A$ de la forme suivante :


→ Forme classique, modèles numériques d'EDP sur maillages structurés.

▷ Méthode de relaxation par blocs :

$$A_{ii}x_i^{(n+1)} = \omega(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(n)}) + (1-\omega)A_{ii}x_i^{(n)}$$

⇒ Système linéaire $m \times m$ à résoudre à chaque itération :

$$A_{ii}x_i^{(n+1)} = c_i^n \quad 1 \leq i \leq n$$

# Méthode de relaxation

## Cas de matrices définies par blocs

▷ Soit $A$ de la forme suivante :


→ Forme classique, modèles numériques d'EDP sur maillages structurés.

▷ Méthode de relaxation par blocs :

$$A_{ii}x_i^{(n+1)} = \omega(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(n)}) + (1-\omega)A_{ii}x_i^{(n)}$$

⇒ Système linéaire $m \times m$ à résoudre à chaque itération :

$$A_{ii}x_i^{(n+1)} = c_i^n \quad 1 \leq i \leq n$$

→ Factorisations LU des matrices $A_{ii}$.

# Méthodes itératives de base pour systèmes linéaires

## Méthodes de décomposition - relaxation


***

*That's all for now Folks !*

Jerome.Monnier@insa-toulouse.fr