

# Analyse Numérique I\*

Sup'Galilée, Ingénieurs MACS, 1ère année / L3 MIM

François Cuvelier

Laboratoire d'Analyse Géométrie et Applications  
Institut Galilée  
Université Paris XIII.

2025/11/25

# Chapitre IV

## Résolution de systèmes linéaires

- 1 Conditionnement
- 2 Méthodes directes
- 3 Méthodes itératives**
  - Principe et résultats généraux
  - Notations
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Méthode de relaxation

- Etude de la convergence
- Algorithmes scalaires
  - Principe de base
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Jeux algorithmiques
  - Méthode S.O.R.
- Algorithmes matriciels
- Autres méthodes

Méthodes itératives pour la résolution du système linéaire

$$\mathbb{A}\mathbf{x} = \mathbf{b}.$$

Trouver une **matrice d'itération**  $\mathbb{B}$  et d'un vecteur  $\mathbf{c}$  telles que

$$\mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}, \quad k \geq 0, \quad \mathbf{x}^{[0]} \text{ arbitraire}$$

vérifie

$$\lim_{k \rightarrow \infty} \mathbf{x}^{[k]} = \tilde{\mathbf{x}} \quad \text{avec} \quad \tilde{\mathbf{x}} = \mathbb{A}^{-1}\mathbf{b}$$

### Exercice 1



Soit  $\mathbb{A}$  une matrice inversible décomposée sous la forme  $\mathbb{A} = \mathbb{M} - \mathbb{N}$  avec  $\mathbb{M}$  inversible. On pose

$$\mathbb{B} = \mathbb{M}^{-1}\mathbb{N} \text{ et } \mathbf{c} = \mathbb{M}^{-1}\mathbf{b}.$$

Montrer que la suite définie par

$$\mathbf{x}^{[0]} \in \mathbb{K}^n \text{ et } \mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}$$

converge vers  $\bar{\mathbf{x}} = \mathbb{A}^{-1}\mathbf{b}$  quelque soit  $\mathbf{x}^{[0]}$  si et seulement si  $\rho(\mathbb{B}) < 1$ .

### Théorème 3.1

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  une matrice inversible décomposée sous la forme  $\mathbb{A} = \mathbb{M} - \mathbb{N}$  avec  $\mathbb{M}$  inversible. On pose

$$\mathbb{B} = \mathbb{M}^{-1}\mathbb{N} \text{ et } \mathbf{c} = \mathbb{M}^{-1}\mathbf{b}.$$

Alors la suite définie par

$$\mathbf{x}^{[0]} \in \mathbb{K}^n \text{ et } \mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}$$

converge vers  $\bar{\mathbf{x}} = \mathbb{A}^{-1}\mathbf{b}$  quelque soit  $\mathbf{x}^{[0]}$  si et seulement si  $\rho(\mathbb{B}) < 1$ .

## Illustration de la convergence

Avec les notations du Théorème et, en supposant  $\rho(\mathbb{B}) < 1$ , on pose

$$\mathbf{e}^{[k]} = \mathbf{x}^{[k]} - \bar{\mathbf{x}}$$

Si  $\mathbb{B}$  normale (pour simplifier) alors  $\exists \mathbb{U}$  unitaire et  $\mathbb{D}$  diagonale t.q.

$$\mathbb{B} = \mathbb{U}\mathbb{D}\mathbb{U}^*.$$

On a alors

$$\|\mathbb{D}\|_p = \rho(\mathbb{D}) = \rho(\mathbb{B})$$

et

$$\mathbf{e}^{[k]} = \mathbb{B}^k \mathbf{e}^{[0]} = (\mathbb{U}\mathbb{D}\mathbb{U}^*)^k \mathbf{e}^{[0]} = \mathbb{U}\mathbb{D}^k \mathbb{U}^* \mathbf{e}^{[0]} \quad \text{car } \mathbb{U} \text{ unitaire}$$

En posant  $\mathbf{E}^{[k]} = \mathbb{U}^* \mathbf{e}^{[k]}$ , on a

$$\mathbf{E}^{[k]} = \mathbb{D}^k \mathbf{E}^{[0]}$$

et on obtient

$$\|\mathbf{E}^{[k]}\|_p \leq \|\mathbb{D}\|_p^k \|\mathbf{E}^{[0]}\|_p = \rho(\mathbb{B})^k \|\mathbf{E}^{[0]}\|_p.$$

Le **facteur asymptotique de convergence** est  $\rho(\mathbb{B})$ .

⇒ Plus le rayon spectral de  $\mathbb{B}$  est proche de 0, plus rapide est la convergence

## Exercice 2

Soit  $A \in \mathcal{M}_{n,n}(\mathbb{C})$  une matrice hermitienne inversible décomposée en  $A = M - N$  où  $M$  est inversible. On note  $B = I - M^{-1}A$ .

Q. 1

Montrer que la matrice  $M^* + N$  est hermitienne.

On suppose maintenant que  $M^* + N$  est définie positive.

Q. 2

Soit  $x$  un vecteur quelconque de  $\mathbb{C}^n$  et  $y = Bx$ .

① Montrer que

$$\langle x, Ax \rangle - \langle y, Ay \rangle = \langle x, AM^{-1}Ax \rangle + \langle M^{-1}Ax, Ax \rangle - \langle M^{-1}Ax, AM^{-1}Ax \rangle \quad (1)$$

et

$$x - y = M^{-1}Ax. \quad (2)$$

② En déduire que

$$\langle x, Ax \rangle - \langle y, Ay \rangle = \langle (x - y), (M^* + N)(x - y) \rangle. \quad (3)$$

Q. 3

Montrer que si  $A$  est définie positive alors  $\rho(B) < 1$ .

On suppose  $\rho(B) < 1$  et on va démontrer, par l'absurde, que  $A$  est définie positive.

Q. 4

On suppose qu'il existe  $x^{[0]} \in \mathbb{C}^n \setminus \{0\}$  tel que  $\alpha_0 \stackrel{\text{def}}{=} \langle x^{[0]}, Ax^{[0]} \rangle \in \mathbb{C} \setminus ]0, +\infty[$ . On définit alors les suites

$$\forall k \in \mathbb{N}^*, x^{[k]} = Bx^{[k-1]} \quad \text{et} \quad \alpha_k = \langle x^{[k]}, Ax^{[k]} \rangle.$$

① Montrer que

$$\lim_{k \rightarrow +\infty} x^{[k]} = 0 \quad \text{et} \quad \lim_{k \rightarrow +\infty} \alpha_k = 0.$$

② Montrer que  $\alpha_0 \in ]-\infty, 0]$ .

③ Démontrer par récurrence sur  $k \in \mathbb{N}^*$  que

$$(\mathcal{P}_k) : x^{[k]} \neq 0, \quad x^{[k]} - x^{[k-1]} \neq 0, \quad \text{et} \quad 0 \geq \alpha_{k-1} > \alpha_k.$$

④ Conclure.

### Théorème 3.2

Soient  $\mathbb{A}$  une matrice hermitienne inversible décomposée en  $\mathbb{A} = \mathbb{M} - \mathbb{N}$  avec  $\mathbb{M}$  inversible et  $\mathbb{M}^* + \mathbb{N}$  hermitienne définie positive. On a alors

$$\rho(\mathbb{M}^{-1}\mathbb{N}) < 1 \text{ si et seulement si } \mathbb{A} \text{ est définie positive.}$$



- 1 Conditionnement
- 2 Méthodes directes
- 3 **Méthodes itératives**
  - Principe et résultats généraux
  - **Notations**
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Méthode de relaxation
- Etude de la convergence
- Algorithmes scalaires
  - Principe de base
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Jeux algorithmiques
  - Méthode S.O.R.
- Algorithmes matriciels
- Autres méthodes

Soit  $A \in \mathcal{M}_n(\mathbb{K})$  une matrice régulière, avec  $\forall i \in \llbracket 1, n \rrbracket, A_{i,i} \neq 0$

$$\begin{aligned}
 A &= \begin{pmatrix} A_{1,1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_{n,n} \end{pmatrix} + \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ A_{2,1} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ A_{n,1} & \cdots & A_{n,n-1} & 0 \end{pmatrix} + \begin{pmatrix} 0 & A_{1,2} & \cdots & A_{1,n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & A_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{pmatrix} \\
 &= D - E - F = \begin{pmatrix} \ddots & & -F \\ & D & \\ -E & & \ddots \end{pmatrix}
 \end{aligned}$$

- 1 Conditionnement
- 2 Méthodes directes
- 3 **Méthodes itératives**
  - Principe et résultats généraux
  - Notations
  - **Méthode de Jacobi**
  - Méthode de Gauss-Seidel
  - Méthode de relaxation
- Etude de la convergence
- Algorithmes scalaires
  - Principe de base
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Jeux algorithmiques
  - Méthode S.O.R.
- Algorithmes matriciels
- Autres méthodes

$$\mathbb{A}\mathbf{x} = \mathbf{b} \iff \forall i, b_i = \sum_{j=1}^n A_{i,j}x_j = \sum_{j=1}^{i-1} A_{i,j}x_j + A_{i,i}x_i + \sum_{j=i+1}^n A_{i,j}x_j$$

La méthode itérative de **Jacobi** :

$$\forall i \in \llbracket 1, n \rrbracket, b_i = \sum_{j=1}^{i-1} A_{i,j}x_j^{[k]} + A_{i,i}x_i^{[k+1]} + \sum_{j=i+1}^n A_{i,j}x_j^{[k]}$$

ou encore

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

$$\mathbb{A}\mathbf{x} = \mathbf{b} \iff \forall i, b_i = \sum_{j=1}^n A_{i,j}x_j = \sum_{j=1}^{i-1} A_{i,j}x_j + A_{i,i}x_i + \sum_{j=i+1}^n A_{i,j}x_j$$

La méthode itérative de **Jacobi** :

$$\forall i \in \llbracket 1, n \rrbracket, b_i = \sum_{j=1}^{i-1} A_{i,j}x_j^{[k]} + A_{i,i}x_i^{[k+1]} + \sum_{j=i+1}^n A_{i,j}x_j^{[k]}$$

$$\iff \mathbf{b} = -\mathbb{E}\mathbf{x}^{[k]} + \mathbb{D}\mathbf{x}^{[k+1]} - \mathbb{F}\mathbf{x}^{[k]}$$

ou encore

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

$$\iff \mathbf{x}^{[k+1]} = \mathbb{D}^{-1}(\mathbb{E} + \mathbb{F})\mathbf{x}^{[k]} + \mathbb{D}^{-1}\mathbf{b}$$

On note  $\mathbb{J}$  sa matrice d'itération

$$\mathbb{J} \stackrel{\text{def}}{=} \mathbb{D}^{-1}(\mathbb{E} + \mathbb{F}).$$

- 1 Conditionnement
- 2 Méthodes directes
- 3 **Méthodes itératives**
  - Principe et résultats généraux
  - Notations
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Méthode de relaxation
- Etude de la convergence
- Algorithmes scalaires
  - Principe de base
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Jeux algorithmiques
  - Méthode S.O.R.
- Algorithmes matriciels
- Autres méthodes

$$A\mathbf{x} = \mathbf{b} \iff \forall i, b_i = \sum_{j=1}^n A_{i,j}x_j = \sum_{j=1}^{i-1} A_{i,j}x_j + A_{i,i}x_i + \sum_{j=i+1}^n A_{i,j}x_j$$

La méthode itérative de **Gauss-Seidel** :

$$\forall i \in \llbracket 1, n \rrbracket, b_i = A_{i,i}x_i^{[k+1]} + \sum_{j=1}^{i-1} A_{i,j}x_j^{[k+1]} + \sum_{j=i+1}^n A_{i,j}x_j^{[k]} \quad \forall i \in \llbracket 1, n \rrbracket$$

ou encore

$$\forall i \in \llbracket 1, n \rrbracket, x_i^{[k+1]} = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij}x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij}x_j^{[k]} \right)$$

$$\mathbb{A}\mathbf{x} = \mathbf{b} \iff \forall i, b_i = \sum_{j=1}^n A_{i,j}x_j = \sum_{j=1}^{i-1} A_{i,j}x_j + A_{i,i}x_i + \sum_{j=i+1}^n A_{i,j}x_j$$

La méthode itérative de **Gauss-Seidel** :

$$\forall i \in \llbracket 1, n \rrbracket, b_i = A_{i,i}x_i^{[k+1]} + \sum_{j=1}^{i-1} A_{i,j}x_j^{[k+1]} + \sum_{j=i+1}^n A_{i,j}x_j^{[k]} \quad \forall i \in \llbracket 1, n \rrbracket$$

$$\iff \mathbf{b} = -\mathbb{E}\mathbf{x}^{[k+1]} + \mathbb{D}\mathbf{x}^{[k+1]} - \mathbb{F}\mathbf{x}^{[k]}$$

ou encore

$$\forall i \in \llbracket 1, n \rrbracket, x_i^{[k+1]} = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij}x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij}x_j^{[k]} \right)$$

$$\iff \mathbf{x}^{[k+1]} = (\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}\mathbf{x}^{[k]} + (\mathbb{D} - \mathbb{E})^{-1}\mathbf{b}$$

On note  $\mathbb{G}$  sa matrice d'itération

$$\mathbb{G} \stackrel{\text{def}}{=} (\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}$$



- 1 Conditionnement
- 2 Méthodes directes
- 3 **Méthodes itératives**
  - Principe et résultats généraux
  - Notations
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Méthode de relaxation
- Etude de la convergence
- Algorithmes scalaires
  - Principe de base
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Jeux algorithmiques
  - Méthode S.O.R.
- Algorithmes matriciels
- Autres méthodes

Soit  $w \in \mathbb{R}^*$ .

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

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

Avec la méthode de Gauss-Seidel : méthode S.O.R. (successive over relaxation)

### Exercice 3

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$  une matrice inversible dont les éléments diagonaux sont non nuls. On note  $A_{ij}$  la composante  $(i, j)$  de la matrice  $\mathbb{A}$ . On décompose la matrice  $\mathbb{A}$  sous la forme  $\mathbb{A} = \mathbb{D} - \mathbb{E} - \mathbb{F}$ , où  $\mathbb{D}$  représente la diagonale de  $\mathbb{A}$ ,  $-\mathbb{E}$  la partie triangulaire inférieure stricte et  $-\mathbb{F}$  la partie triangulaire supérieure stricte.

La méthode S.O.R. (successive over relaxation) est donnée par

$$x_i^{[k+1]} = \frac{w}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij}x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij}x_j^{[k]} \right) + (1-w)x_i^{[k]}, \quad \forall i \in \llbracket 1, n \rrbracket$$

Q. 1

Déterminer la matrice d'itération  $\mathbb{B}$  et le vecteur  $\mathbf{c}$  tels que

$$\mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}$$

en fonction de  $\mathbb{D}$ ,  $\mathbb{E}$ ,  $\mathbb{F}$ , et  $\mathbf{b}$ .

La méthode itérative **S.O.R.** de paramètre  $w \in \mathbb{R}$

$$\forall i \in \llbracket 1, n \rrbracket, \quad x_i^{[k+1]} = \frac{w}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij} x_j^{[k]} \right) + (1-w)x_i^{[k]}$$

ou vectoriellement

$$\mathbf{x}^{[k+1]} = \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \left( \frac{1-w}{w} \mathbb{D} + \mathbb{F} \right) \mathbf{x}^{[k]} + \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \mathbf{b}.$$

Sa matrice d'itération est

$$\mathcal{L}_w \stackrel{\text{def}}{=} \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \left( \frac{1-w}{w} \mathbb{D} + \mathbb{F} \right). \quad (1)$$

En particulier, on a  $\mathcal{L}_1 = \mathbb{G}$  matrice d'itération de Gauss-Seidel.

- 1 Conditionnement
- 2 Méthodes directes
- 3 Méthodes itératives
  - Principe et résultats généraux
  - Notations
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Méthode de relaxation

- Etude de la convergence
- Algorithmes scalaires
  - Principe de base
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Jeux algorithmiques
  - Méthode S.O.R.
- Algorithmes matriciels
- Autres méthodes

Avec les notations du Théorème 3.1, on a

- Jacobi :  $\mathbb{J} \stackrel{\text{def}}{=} \mathbb{D}^{-1}(\mathbb{E} + \mathbb{F})$  avec  $\mathbb{M} = \mathbb{D}$  et  $\mathbb{N} = \mathbb{E} + \mathbb{F}$

convergence si et seulement si  $\rho(\mathbb{J}) < 1$ .

- Gauss-Seidel :  $\mathbb{G} \stackrel{\text{def}}{=} (\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}$  avec  $\mathbb{M} = \mathbb{D} - \mathbb{E}$  et  $\mathbb{N} = \mathbb{F}$

convergence si et seulement si  $\rho(\mathbb{G}) < 1$ .

- S.O.R. :  $\mathcal{L}_w \stackrel{\text{def}}{=} \left(\frac{\mathbb{D}}{w} - \mathbb{E}\right)^{-1} \left(\frac{1-w}{w}\mathbb{D} + \mathbb{F}\right)$  avec  $\mathbb{M} = \frac{\mathbb{D}}{w} - \mathbb{E}$  et  $\mathbb{N} = \frac{1-w}{w}\mathbb{D} + \mathbb{F}$

convergence si et seulement si  $\rho(\mathcal{L}_w) < 1$ .

Le Théorème 3.2 peut aussi être utilisé:

$\mathbb{A}$  et  $(\mathbb{M}^* + \mathbb{N})$  hermitiennes définies positives alors convergence.

#### Exercice 4



Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$  une matrice inversible dont les éléments diagonaux sont non nuls. On note  $A_{i,j}$  la composante  $(i,j)$  de la matrice  $\mathbb{A}$ . On décompose la matrice  $\mathbb{A}$  sous la forme  $\mathbb{A} = \mathbb{D} - \mathbb{E} - \mathbb{F}$ , où  $\mathbb{D}$  représente la diagonale de  $\mathbb{A}$ ,  $-\mathbb{E}$  la partie triangulaire inférieure stricte et  $-\mathbb{F}$  la partie triangulaire supérieure stricte.

La matrice d'itération de la méthode S.O.R., notée  $\mathcal{L}_w$ , est donnée par

$$\mathcal{L}_w = \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \left( \frac{1-w}{w} \mathbb{D} + \mathbb{F} \right). \quad (1)$$

On pose  $\mathbb{L} = \mathbb{D}^{-1}\mathbb{E}$  et  $\mathbb{U} = \mathbb{D}^{-1}\mathbb{F}$ .

Q. 1

*Montrer que*

$$\mathcal{L}_w = (\mathbb{I} - w\mathbb{L})^{-1} ((1-w)\mathbb{I} + w\mathbb{U}).$$

Q. 2

*En déduire que*

$$\rho(\mathcal{L}_w) \geq |w - 1|. \quad (2)$$

### Proposition 3.1

Soit  $\mathbb{A}$  une matrice inversible telle que tous ses éléments diagonaux soient non nuls. On note  $\mathbb{D} = \text{diag}(\mathbb{A})$  et  $\mathbb{E}$ ,  $\mathbb{F}$ , les matrices à diagonales nulles respectivement triangulaire inférieure et supérieure telles que  $\mathbb{A} = \mathbb{D} - \mathbb{E} - \mathbb{F}$ .

La matrice d'itération de la méthode S.O.R., notée  $\mathcal{L}_w$ , donnée par


$$\mathcal{L}_w = \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \left( \frac{1-w}{w} \mathbb{D} + \mathbb{F} \right)$$

vérifie

$$\rho(\mathcal{L}_w) \geq |w - 1|. \quad (2)$$

La méthode S.O.R. diverge si  $w \in ]-\infty, 0] \cup [2, +\infty[$ .

Une condition nécessaire de convergence de la méthode S.O.R. est  $0 < w < 2$ .

 La condition  $0 < w < 2$  est nécessaire mais **non suffisante** pour avoir convergence!

### Exercice 5

On note  $\mathbb{T} \in \mathcal{M}_n(\mathbb{C})$  la matrice tridiagonale

$$\mathbb{T} = \begin{pmatrix} a_1 & c_1 & 0 & \dots & 0 \\ b_2 & a_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & \dots & 0 & b_n & a_n \end{pmatrix}. \quad (1)$$

Q. 1

Soit  $\mu \in \mathbb{C}^*$ . On note  $\mathbb{Q}(\mu) \in \mathcal{M}_n(\mathbb{C})$  la matrice diagonale de diagonale  $(\mu, \mu^2, \dots, \mu^n)$ .

- ④ Expliciter la matrice  $\mathbb{T}(\mu) \stackrel{\text{def}}{=} \mathbb{Q}(\mu)\mathbb{T}\mathbb{Q}^{-1}(\mu)$  en fonction des coefficients tridiagonaux de la matrice  $\mathbb{T}$  et de  $\mu$ .
- ⑤ Déterminer  $\det(\mathbb{T}(\mu))$  en fonction de  $\det(\mathbb{T})$ .

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$  une matrice inversible dont les éléments diagonaux sont non nuls. On note  $\mathbb{A}_{i,j}$  la composante  $(i,j)$  de la matrice  $\mathbb{A}$ . On décompose la matrice  $\mathbb{A}$  sous la forme  $\mathbb{A} = \mathbb{D} - \mathbb{E} - \mathbb{F}$ , où  $\mathbb{D}$  représente la diagonale de  $\mathbb{A}$ ,  $-\mathbb{E}$  la partie triangulaire inférieure stricte et  $-\mathbb{F}$  la partie triangulaire supérieure stricte.

On note respectivement  $\mathbb{J} \stackrel{\text{def}}{=} \mathbb{D}^{-1}(\mathbb{E} + \mathbb{F})$  et  $\mathcal{L}_1 \stackrel{\text{def}}{=} (\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}$  les matrices d'itérations des méthodes de Jacobi et de Gauss-Seidel.

Soit  $\mathbf{y} \in \mathbb{R}^n$ . On souhaite résoudre le système  $\mathbb{A}\mathbf{x} = \mathbf{y}$  par la méthode de Gauss-Seidel ou par la méthode de Jacobi.

On suppose dans la suite que la matrice inversible  $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$  s'écrit sous la forme

$$\mathbb{A} = \begin{pmatrix} \alpha_1 & \nu_1 & 0 & \dots & 0 \\ \beta_2 & \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \nu_{n-1} \\ 0 & \dots & 0 & \beta_n & \alpha_n \end{pmatrix} \quad (2)$$

et que ses éléments diagonaux sont non nuls.

Q. 2

- ④ Montrer que les valeurs propres de  $\mathbb{J}$  sont les racines du polynôme

$$q_{\mathbb{J}}(\lambda) \stackrel{\text{def}}{=} \det(\lambda\mathbb{D} - \mathbb{E} - \mathbb{F}).$$

- ⑤ En utilisant la question 1, montrer que  $q_{\mathbb{J}}(\lambda) = \det(\lambda\mathbb{D} - \lambda\mathbb{E} - \frac{1}{\lambda}\mathbb{F})$ .
- ⑥ En déduire que si  $\lambda \in \mathbb{C}$  est valeur propre de  $\mathbb{J}$  alors  $-\lambda$  l'est aussi.

Q. 3

- ④ Montrer que les valeurs propres de  $\mathcal{L}_1$  sont les racines du polynôme

$$q_{\mathcal{L}_1}(\lambda) \stackrel{\text{def}}{=} \det(\lambda\mathbb{D} - \lambda\mathbb{E} - \mathbb{F}).$$

- ⑤ En déduire que

$$\forall \lambda \in \mathbb{C}^*, \quad q_{\mathcal{L}_1}(\lambda^2) = \lambda^n q_{\mathbb{J}}(\lambda). \quad (3)$$

Q. 4

- ④ Comparer les valeurs propres de  $\mathbb{J}$  à celles de  $\mathcal{L}_1$ .
- ⑤ Une des deux méthodes est-elle à privilégier dans ce cas?



### Proposition 3.2

Soit  $A \in \mathcal{M}_n(\mathbb{R})$  une matrice tridiagonale ( i.e.  $A_{i,j} = 0$ , si  $|i-j| > 1$ ) d'éléments diagonaux non nuls.

Alors les rayons spectraux des matrices d'itération de Jacobi,  $\mathbb{J}$ , et de Gauss-Seidel,  $\mathcal{L}_1$ , vérifient

$$\rho(\mathcal{L}_1) = \rho(\mathbb{J})^2.$$

Dans ce cas,

- les méthodes de Jacobi et de Gauss-Seidel convergent ou divergent simultanément.
- Si elles convergent, alors la méthode de Gauss-Seidel converge plus rapidement.



**Proposition 3.3 :** voir Ciarlet[2006], Introduction à l'analyse numérique matricielle et à l'optimisation, Théorème 5.3-5, pages 106 à 109.

Soit  $A \in \mathcal{M}_n(\mathbb{R})$  une matrice tridiagonale dont les éléments diagonaux sont non nuls. On suppose que les valeurs propres de la matrice d'itération de Jacobi  $J$  sont réelles et que  $\rho(J) < 1$ . On note  $w_0$  le paramètre optimal de la méthode S.O.R. vérifiant

$$\rho(\mathcal{L}_{w_0}) = \min_{w \in ]0,2[} (\rho(\mathcal{L}_w)).$$

et donné par

$$w_0 = \frac{2}{1 + \sqrt{1 - \rho(J)^2}} > 1. \quad (3)$$

On a alors

$$\rho(\mathcal{L}_{w_0}) = w_0 - 1 \quad \text{et} \quad \rho(\mathcal{L}_{w_0}) \leq \rho(\mathcal{L}_1) = \rho(J)^2 < \rho(J).$$

Ici,  $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$ , matrice tridiagonale aléatoire inversible à éléments diagonaux non nuls t.q. les valeurs propres de  $\mathbb{J}$  sont réelles et  $\rho(\mathbb{J}) < 1$ .

D'après la proposition précédente

$$w_0 = \frac{2}{1 + \sqrt{1 - \rho(\mathbb{J})^2}} > 1, \quad \rho(\mathcal{L}_{w_0}) = \min(\rho(\mathcal{L}_w), w \in ]0, 2[), \quad \rho(\mathcal{L}_{w_0}) = w_0 - 1.$$

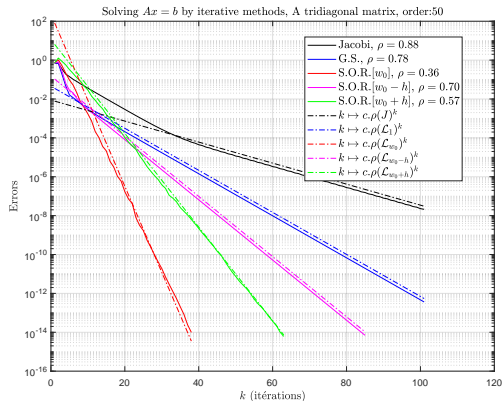
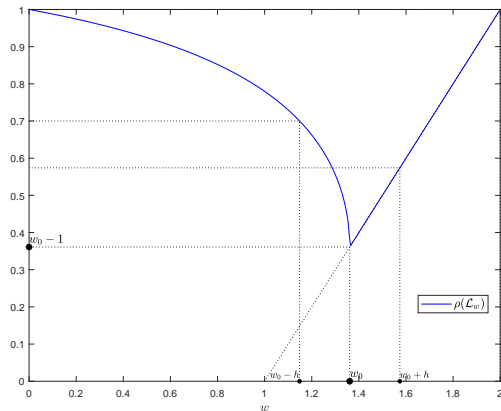


Figure:  $\mathbb{A}$  est d'ordre  $n = 50$ .

Ici,  $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$ , matrice tridiagonale aléatoire inversible à éléments diagonaux non nuls t.q. les valeurs propres de  $\mathbb{J}$  sont réelles et  $\rho(\mathbb{J}) < 1$ .

D'après la proposition précédente

$$w_0 = \frac{2}{1 + \sqrt{1 - \rho(\mathbb{J})^2}} > 1, \quad \rho(\mathcal{L}_{w_0}) = \min(\rho(\mathcal{L}_w), w \in ]0, 2[), \quad \rho(\mathcal{L}_{w_0}) = w_0 - 1.$$

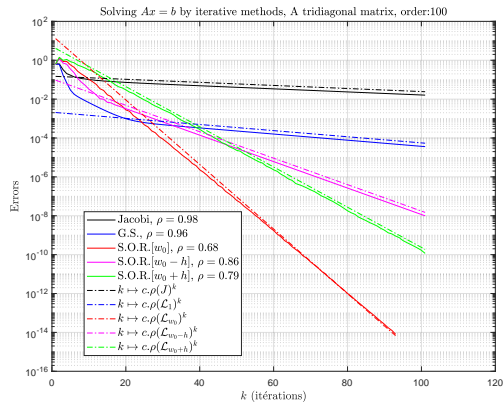
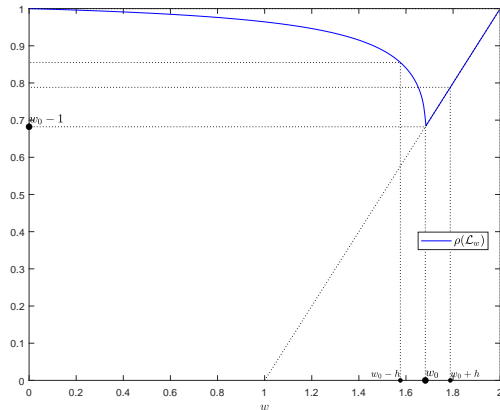


Figure:  $\mathbb{A}$  est d'ordre  $n = 100$ .

**Théorème 3.3 :** voir Lascaux-Théodor, vol.2, Théorème 19 et 20, pages 346 à 349

Soit  $\mathbb{A}$  une matrice à diagonale strictement dominante ou une matrice inversible à diagonale fortement dominante alors

- la méthode de Jacobi est convergente,
- si  $w \in ]0, 1]$  la méthode S.O.R. est convergente.

**Théorème 3.4 :** voir Lascaux-Théodor, vol.2, Corollaire 24, page 351

Soit  $\mathbb{A}$  une matrice hermitienne définie positive, alors la méthode S.O.R. converge si et seulement si  $w \in ]0, 2[$ .

*Preuve :* on utilise le Théorème 3.2 du résumé ou Théorème 3.4.3 p.115 du polycopié...

- 1 Conditionnement
  - 2 Méthodes directes
  - 3 **Méthodes itératives**
    - Principe et résultats généraux
    - Notations
    - Méthode de Jacobi
    - Méthode de Gauss-Seidel
    - Méthode de relaxation
- Etude de la convergence
  - **Algorithmes scalaires**
    - Principe de base
    - Méthode de Jacobi
    - Méthode de Gauss-Seidel
    - Jeux algorithmiques
    - Méthode S.O.R.
  - Algorithmes matriciels
  - Autres méthodes

# Principe de base

Résoudre :

$$\mathbb{A}\mathbf{x} = \mathbf{b}$$

Méthodes itératives :

$$\mathbf{x}^{[0]} \in \mathbb{K}^n \text{ et } \mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}$$

Algorithme :

```
 $\mathbf{x}^{[0]}$  donné  
Pour  $k = 0, 1, \dots$  faire  
     $\mathbf{x}^{[k+1]} \leftarrow \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}$   
Fin Pour
```

Critère d'arrêt? Stockage de tous les  $\mathbf{x}^{[k]}$ ?

La convergence de ces méthodes n'est pas assurée et si il y a convergence le nombre d'itération nécessaire n'est (à priori) pas connu.

⇒ boucle **Tantque**

**Critères d'arrêt :**

- nombre maximum d'itérations
- $\varepsilon > 0$  permet l'arrêt des calculs si  $\mathbf{x}^{[k]}$  suffisamment proche de  $\bar{\mathbf{x}} = \mathbb{A}^{-1}\mathbf{b}$

Comment choisir le critère d'arrêt pour la convergence?

Exemple de critère d'arrêt pour la convergence :

Soit  $\mathbf{r}^{[k]} = \mathbf{b} - \mathbb{A}\mathbf{x}^{[k]}$  le résidu.

$$\frac{\|\mathbf{r}^{[k]}\|}{\|\mathbf{b}\|} \leq \varepsilon$$

Car dans ce cas, on a avec  $\mathbf{e}^{[k]} \stackrel{\text{def}}{=} \bar{\mathbf{x}} - \mathbf{x}^{[k]} = \mathbb{A}^{-1}\mathbf{r}^{[k]}$

$$\frac{\|\mathbf{e}^{[k]}\|}{\|\bar{\mathbf{x}}\|} \leq \varepsilon \text{cond}(\mathbb{A})$$

Pour éviter des problèmes avec  $\mathbf{b}$  proche de  $\mathbf{0}$  :

$$\frac{\|\mathbf{r}^{[k]}\|}{\|\mathbf{b}\| + 1} \leq \varepsilon$$



---

**Algorithme** Méthode itérative pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$ 

---

**Données :**

$\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  ,  
 $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$  ,  
 $\mathbf{x}^0$  : vecteur initial de  $\mathbb{K}^n$  ,  
 $\varepsilon$  : la tolérance,  $\varepsilon \in \mathbb{R}^+$  ,  
kmax : nombre maximum d'itérations, kmax  $\in \mathbb{N}^*$

**Résultat :**

$\mathbf{x}^{\text{tol}}$  : un vecteur de  $\mathbb{K}^n$  si convergence, sinon  $\emptyset$

```
1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:    $\mathbf{x} \leftarrow$  calcul de l'itérée suivante en fonction de  $\mathbf{p}, \mathbb{A}, \mathbf{b}, \dots$ 
8:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
9: Fin Tantque
10: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
11:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
12: Fin Si
```

$\triangleright \mathbf{p}$  contient le vecteur précédent

$\triangleright$  Convergence



$$\text{Jacobi: } x_i^{[k+1]} = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij} x_j^{[k]} \right), \quad \forall i \in \llbracket 1, n \rrbracket.$$

### Algorithme 2 $\mathcal{R}_1$

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:   Pour  $i \leftarrow 1$  à  $n$  faire
     
$$x_i \leftarrow \frac{1}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij} p_j \right)$$

8:   Fin Pour
9:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
10: Fin Tantque
11: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
12:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
13: Fin Si
```

### Algorithme 2 $\mathcal{R}_2$

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:   Pour  $i \leftarrow 1$  à  $n$  faire
     8:      $S \leftarrow 0$ 
     9:     Pour  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) faire
        10:        $S \leftarrow S + A_{ij} p_j$ 
        11:     Fin Pour
        12:      $x_i \leftarrow \frac{1}{A_{ii}} (b_i - S)$ 
     13:   Fin Pour
     14:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
     15: Fin Tantque
     16: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
     17:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
     18: Fin Si
```

---

**Algorithme** Méthode itérative de Jacobi pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$ 

---

**Données :**

$\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,  
 $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ ,  
 $\mathbf{x}^0$  : vecteur initial de  $\mathbb{K}^n$ ,  
 $\varepsilon$  : la tolérance,  $\varepsilon \in \mathbb{R}^+$ ,  
kmax : nombre maximum d'itérations, kmax  $\in \mathbb{N}^*$

**Résultat :**

$\mathbf{X}$  : un vecteur de  $\mathbb{K}^n$

```
1: Fonction  $\mathbf{X} \leftarrow \text{RSLJacobi}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
2:    $k \leftarrow 0$ ,  $\mathbf{X} \leftarrow \emptyset$ 
3:    $\mathbf{x} \leftarrow \mathbf{x}^0$ ,  $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x}$ ,
4:    $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
5:   Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
6:      $k \leftarrow k + 1$ 
7:      $\mathbf{p} \leftarrow \mathbf{x}$ 
8:     Pour  $i \leftarrow 1$  à  $n$  faire
9:        $S \leftarrow 0$ 
10:      Pour  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) faire
11:         $S \leftarrow S + A(i, j) * p(j)$ 
12:      Fin Pour
13:       $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
14:    Fin Pour
15:     $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x}$ ,
16:  Fin Tantque
17:  Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
18:     $\mathbf{X} \leftarrow \mathbf{x}$ 
19:  Fin Si
20: Fin Fonction
```

---

$$\text{Gauss-Seidel: } x_i^{(k+1)} = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij} x_j^{[k]} \right) \quad \forall i \in \llbracket 1, n \rrbracket$$

### Algorithme 3 $\mathcal{R}_0$

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:
8:    $\mathbf{x} \leftarrow \text{calcul par Gauss-Seidel}$ 
9:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
10: Fin Tantque
11: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
12:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
13: Fin Si
```

### Algorithme 3 $\mathcal{R}_1$

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:
8:   Pour  $i \leftarrow 1$  à  $n$  faire
9:      $x_i \leftarrow \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j - \sum_{j=i+1}^n A_{ij} p_j \right)$ 
10:   Fin Pour
11:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
12: Fin Tantque
13: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
14:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
15: Fin Si
```

$$\text{Gauss-Seidel: } x_i^{(k+1)} = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij} x_j^{[k]} \right) \quad \forall i \in \llbracket 1, n \rrbracket$$

### Algorithme 3 $\mathcal{R}_1$

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:   Pour  $i \leftarrow 1$  à  $n$  faire
8:      $x_i \leftarrow \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j - \sum_{j=i+1}^n A_{ij} p_j \right)$ 
9:   Fin Pour
10:   $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
11: Fin Tantque
12: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
13:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
14: Fin Si
```

### Algorithme 3 $\mathcal{R}_2$

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:   Pour  $i \leftarrow 1$  à  $n$  faire
8:      $S \leftarrow 0$ 
9:     Pour  $j \leftarrow 1$  à  $i - 1$  faire
10:       $S \leftarrow S + A_{i,j} x_j$ 
11:   Fin Pour
12:   Pour  $j \leftarrow i + 1$  à  $n$  faire
13:      $S \leftarrow S + A_{i,j} p_j$ 
14:   Fin Pour
15:    $x_i \leftarrow \frac{1}{A_{ii}} (b_i - S)$ 
16: Fin Pour
17:   $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
18: Fin Tantque
19: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
20:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
21: Fin Si
```

---

**Algorithme** Méthode itérative de Gauss-Seidel pour la résolution d'un système linéaire  $Ax = b$ 

---

**Données :**

- $A$  : matrice de  $M_n(\mathbb{K})$ ,
- $b$  : vecteur de  $\mathbb{K}^n$ ,
- $x^0$  : vecteur initial de  $\mathbb{K}^n$ ,
- $\varepsilon$  : la tolérance,  $\varepsilon \in \mathbb{R}^+$ ,
- kmax : nombre maximum d'itérations, kmax  $\in \mathbb{N}^*$

**Résultat :** $X$  : un vecteur de  $\mathbb{K}^n$ 

```
1: Fonction  $X \leftarrow \text{RSLGaussSeidel}(A, b, x^0, \varepsilon, \text{kmax})$ 
2:    $k \leftarrow 0, X \leftarrow \emptyset$ 
3:    $x \leftarrow x^0, r \leftarrow b - A * x,$ 
4:    $\text{tol} \leftarrow \varepsilon(\|b\| + 1)$ 
5:   Tantque  $\|r\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
6:      $k \leftarrow k + 1$ 
7:      $p \leftarrow x$ 
8:     Pour  $i \leftarrow 1$  à  $n$  faire
9:        $S \leftarrow 0$ 
10:      Pour  $j \leftarrow 1$  à  $i - 1$  faire
11:         $S \leftarrow S + A(i, j) * x(j)$ 
12:      Fin Pour
13:      Pour  $j \leftarrow i + 1$  à  $n$  faire
14:         $S \leftarrow S + A(i, j) * p(j)$ 
15:      Fin Pour
16:       $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
17:    Fin Pour
18:     $r \leftarrow b - A * x,$ 
19:  Fin Tantque
20:  Si  $\|r\| \leq \text{tol}$  alors
21:     $X \leftarrow x$ 
22:  Fin Si
23: Fin Fonction
```

```

Fonction  $X \leftarrow \text{RSLJacobi}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
 $k \leftarrow 0, X \leftarrow \emptyset$ 
 $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
 $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
     $k \leftarrow k + 1$ 
     $\mathbf{p} \leftarrow \mathbf{x}$ 
    Pour  $i \leftarrow 1$  à  $n$  faire
         $S \leftarrow 0$ 
        Pour  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) faire
             $S \leftarrow S + A(i, j) * p(j)$ 
        Fin Pour
         $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
    Fin Pour
     $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
Fin Tantque
Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
     $X \leftarrow \mathbf{x}$ 
Fin Si
Fin Fonction

```

```

Fonction  $X \leftarrow \text{RSLGaussSeidel}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
 $k \leftarrow 0, X \leftarrow \emptyset$ 
 $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
 $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
     $k \leftarrow k + 1$ 
     $\mathbf{p} \leftarrow \mathbf{x}$ 
    Pour  $i \leftarrow 1$  à  $n$  faire
         $S \leftarrow 0$ 
        Pour  $j \leftarrow 1$  à  $i - 1$  faire
             $S \leftarrow S + A(i, j) * x(j)$ 
        Fin Pour
        Pour  $j \leftarrow i + 1$  à  $n$  faire
             $S \leftarrow S + A(i, j) * p(j)$ 
        Fin Pour
         $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
    Fin Pour
     $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
Fin Tantque
Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
     $X \leftarrow \mathbf{x}$ 
Fin Si
Fin Fonction

```



```

Fonction  $X \leftarrow \text{RSLJacobi}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, k_{\max})$ 
 $k \leftarrow 0, X \leftarrow \emptyset$ 
 $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
 $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  faire
   $k \leftarrow k + 1$ 
   $\mathbf{p} \leftarrow \mathbf{x}$ 
  Pour  $i \leftarrow 1$  à  $n$  faire
     $S \leftarrow 0$ 
    Pour  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) faire
       $S \leftarrow S + A(i, j) * p(j)$ 
    Fin Pour
     $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
  Fin Pour
   $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
Fin Tantque
Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
   $X \leftarrow \mathbf{x}$ 
Fin Si
Fin Fonction

```

```

Fonction  $X \leftarrow \text{RSLGaussSeidel}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, k_{\max})$ 
 $k \leftarrow 0, X \leftarrow \emptyset$ 
 $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
 $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  faire
   $k \leftarrow k + 1$ 
   $\mathbf{p} \leftarrow \mathbf{x}$ 
  Pour  $i \leftarrow 1$  à  $n$  faire
     $S \leftarrow 0$ 
    Pour  $j \leftarrow 1$  à  $i - 1$  faire
       $S \leftarrow S + A(i, j) * x(j)$ 
    Fin Pour
    Pour  $j \leftarrow i + 1$  à  $n$  faire
       $S \leftarrow S + A(i, j) * p(j)$ 
    Fin Pour
     $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
  Fin Pour
   $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
Fin Tantque
Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
   $X \leftarrow \mathbf{x}$ 
Fin Si
Fin Fonction

```

Même ossature puisque toutes deux basées sur l'Algorithme générique

Peut-on simplifier, clarifier et raccourcir les codes?

**Fonction**  $X \leftarrow \text{RSLJacobi}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$

$k \leftarrow 0, X \leftarrow \emptyset$

$\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$

$\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$

**Tantque**  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  **faire**

$k \leftarrow k + 1$

$\mathbf{p} \leftarrow \mathbf{x}$

**Pour**  $i \leftarrow 1$  à  $n$  **faire**

$S \leftarrow 0$

**Pour**  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) **faire**

$S \leftarrow S + A(i, j) * p(j)$

**Fin Pour**

$x(i) \leftarrow (b(i) - S) / A(i, i)$

**Fin Pour**

$\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$

**Fin Tantque**

**Si**  $\|\mathbf{r}\| \leq \text{tol}$  **alors**

$X \leftarrow \mathbf{x}$

**Fin Si**

**Fin Fonction**

**Algorithme** Itération de Jacobi : calcul de  $\mathbf{x}$  tel que

$$x_i = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij} y_j \right), \quad \forall i \in \llbracket 1, n \rrbracket.$$

**Données :**

$A$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,

$b$  : vecteur de  $\mathbb{K}^n$ ,

$y$  : vecteur de  $\mathbb{K}^n$ ,

**Résultat :**

$x$  : un vecteur de  $\mathbb{K}^n$

1: **Fonction**  $x \leftarrow \text{IterJacobi}(A, b, y)$

2: **Pour**  $i \leftarrow 1$  à  $n$  **faire**

3:  $S \leftarrow 0$

4: **Pour**  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) **faire**

5:  $S \leftarrow S + A(i, j) * y(j)$

6: **Fin Pour**

7:  $x(i) \leftarrow (b(i) - S) / A(i, i)$

8: **Fin Pour**

9: **Fin Fonction**

```

 $k \leftarrow 0, \mathbf{X} \leftarrow \emptyset$ 
 $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
 $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  faire
     $k \leftarrow k + 1$ 
     $\mathbf{p} \leftarrow \mathbf{x}$ 
     $\mathbf{x} \leftarrow \text{IterJacobi}(\mathbb{A}, \mathbf{b}, \mathbf{p})$ 
     $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
Fin Tantque
Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
     $\mathbf{X} \leftarrow \mathbf{x}$ 
Fin Si
Fin Fonction

```

$$x_i = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij} y_j \right), \quad \forall i \in \llbracket 1, n \rrbracket.$$

A : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,  
b : vecteur de  $\mathbb{K}^n$ ,  
v : vecteur de  $\mathbb{K}^n$ .

$x$  : un vecteur de  $\mathbb{K}^n$

```

1: Fonction  $x \leftarrow \text{IterJacobi}(A, b, y)$ 
2:   Pour  $i \leftarrow 1$  à  $n$  faire
3:      $S \leftarrow 0$ 
4:     Pour  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) faire
5:        $S \leftarrow S + A(i, j) * y(j)$ 
6:     Fin Pour
7:      $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
8:   Fin Pour
9: Fin Fonction

```

**Fonction**  $X \leftarrow \text{RSLGaussSeidel}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, k_{\max})$

$k \leftarrow 0, X \leftarrow \emptyset$

$\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$

$\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$

**Tantque**  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  **faire**

$k \leftarrow k + 1$

$\mathbf{p} \leftarrow \mathbf{x}$

**Pour**  $i \leftarrow 1$  à  $n$  **faire**

$S \leftarrow 0$

**Pour**  $j \leftarrow 1$  à  $i - 1$  **faire**

$S \leftarrow S + A(i, j) * x(j)$

**Fin Pour**

**Pour**  $j \leftarrow i + 1$  à  $n$  **faire**

$S \leftarrow S + A(i, j) * p(j)$

**Fin Pour**

$x(i) \leftarrow (b(i) - S) / A(i, i)$

**Fin Pour**

$\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$

**Fin Tantque**

**Si**  $\|\mathbf{r}\| \leq \text{tol}$  **alors**

$X \leftarrow \mathbf{x}$

**Fin Si**

**Fin Fonction**

**Algorithme** Itération de Gauss-Seidel : calcul de  $\mathbf{x}$  tel que

$$x_i = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j - \sum_{j=i+1}^n A_{ij} y_j \right), \quad \forall i \in \llbracket 1, n \rrbracket.$$

**Données :**

$\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,

$\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ ,

$\mathbf{y}$  : vecteur de  $\mathbb{K}^n$ ,

**Résultat :**

$\mathbf{x}$  : un vecteur de  $\mathbb{K}^n$

1: **Fonction**  $\mathbf{x} \leftarrow \text{IterGaussSeidel}(\mathbb{A}, \mathbf{b}, \mathbf{y})$

2: **Pour**  $i \leftarrow 1$  à  $n$  **faire**

3:  $S \leftarrow 0$

4: **Pour**  $j \leftarrow 1$  à  $i - 1$  **faire**

5:  $S \leftarrow S + A(i, j) * x(j)$

6: **Fin Pour**

7: **Pour**  $j \leftarrow i + 1$  à  $n$  **faire**

8:  $S \leftarrow S + A(i, j) * y(j)$

9: **Fin Pour**

10:  $x(i) \leftarrow (b(i) - S) / A(i, i)$

11: **Fin Pour**

12: **Fin Fonction**

**Fonction**  $X \leftarrow \text{RSLGaussSeidel2}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$

$k \leftarrow 0, X \leftarrow \emptyset$

$\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$

$\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$

**Tantque**  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  **faire**

$k \leftarrow k + 1$

$\mathbf{p} \leftarrow \mathbf{x}$

$\mathbf{x} \leftarrow \text{IterGaussSeidel}(\mathbb{A}, \mathbf{b}, \mathbf{p})$

$\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$

**Fin Tantque**

**Si**  $\|\mathbf{r}\| \leq \text{tol}$  **alors**

$X \leftarrow \mathbf{x}$

**Fin Si**

**Fin Fonction**

**Algorithme** Itération de Gauss-Seidel : calcul de  $\mathbf{x}$  tel que

$$x_i = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^n A_{ij}y_j \right), \quad \forall i \in \llbracket 1, n \rrbracket.$$

**Données :**

$\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,

$\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ ,

$\mathbf{y}$  : vecteur de  $\mathbb{K}^n$ ,

**Résultat :**

$\mathbf{x}$  : un vecteur de  $\mathbb{K}^n$

1: **Fonction**  $\mathbf{x} \leftarrow \text{IterGaussSeidel}(\mathbb{A}, \mathbf{b}, \mathbf{y})$

2: **Pour**  $i \leftarrow 1$  à  $n$  **faire**

3:  $S \leftarrow 0$

4: **Pour**  $j \leftarrow 1$  à  $i - 1$  **faire**

5:  $S \leftarrow S + A(i, j) * x(j)$

6: **Fin Pour**

7: **Pour**  $j \leftarrow i + 1$  à  $n$  **faire**

8:  $S \leftarrow S + A(i, j) * y(j)$

9: **Fin Pour**

10:  $x(i) \leftarrow (b(i) - S) / A(i, i)$

11: **Fin Pour**

12: **Fin Fonction**

**Fonction**  $X \leftarrow \text{RSLGaussSeidel2}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, k_{\max})$

$k \leftarrow 0, X \leftarrow \emptyset$

$\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$

$\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$

**Tantque**  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  **faire**

$k \leftarrow k + 1$

$\mathbf{p} \leftarrow \mathbf{x}$

$\mathbf{x} \leftarrow \text{IterGaussSeidel}(\mathbb{A}, \mathbf{b}, \mathbf{p})$

$\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$

**Fin Tantque**

**Si**  $\|\mathbf{r}\| \leq \text{tol}$  **alors**

$X \leftarrow \mathbf{x}$

**Fin Si**

**Fin Fonction**

**Fonction**  $X \leftarrow \text{RSLJacobi2}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, k_{\max})$

$k \leftarrow 0, X \leftarrow \emptyset$

$\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$

$\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$

**Tantque**  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq k_{\max}$  **faire**

$k \leftarrow k + 1$

$\mathbf{p} \leftarrow \mathbf{x}$

$\mathbf{x} \leftarrow \text{IterJacobi}(\mathbb{A}, \mathbf{b}, \mathbf{p})$

$\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$

**Fin Tantque**

**Si**  $\|\mathbf{r}\| \leq \text{tol}$  **alors**

$X \leftarrow \mathbf{x}$

**Fin Si**

**Fin Fonction**

Les deux codes sont fortement similaires!

Peut-on éviter les copier/coller et gagner encore en lisibilité?

Ecriture Algorithme générique sous forme d'une fonction et on ajoute aux paramètres d'entrées une fonction formelle **IterFonc** calculant une itérée :

$$\mathbf{x} \leftarrow \text{IterFonc}(\mathbb{A}, \mathbf{b}, \mathbf{y}).$$

---

**Algorithme** Méthode itérative pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$

---

**Données :**

$\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  ,  
 $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$  ,  
**IterFonc** : fonction de paramètres une matrice d'ordre  $n$ ,  
et deux vecteurs de  $\mathbb{K}^n$ . retourne un vecteur de  $\mathbb{K}^n$ .  
 $\mathbf{x}^0$  : vecteur initial de  $\mathbb{K}^n$ ,  
 $\varepsilon$  : la tolérance,  $\varepsilon \in \mathbb{R}^+$ ,  
kmax : nombre maximum d'itérations, kmax  $\in \mathbb{N}^*$

**Résultat :**

$\mathbf{x}^{\text{tol}}$  : un vecteur de  $\mathbb{K}^n$  si convergence, sinon  $\emptyset$

```
1: Fonction  $\mathbf{X} \leftarrow \text{RSLMethIter}(\mathbb{A}, \mathbf{b}, \text{IterFonc}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
2:    $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
3:    $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
4:    $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
5:   Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
6:      $k \leftarrow k + 1$ 
7:      $\mathbf{p} \leftarrow \mathbf{x}$ 
8:      $\mathbf{x} \leftarrow \text{IterFonc}(\mathbb{A}, \mathbf{b}, \mathbf{p})$ 
9:      $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
10:  Fin Tantque
11:  Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
12:     $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
13:  Fin Si
14: Fin Fonction
```

**Fonction**  $X \leftarrow \text{RSLJacobi3}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$   
 $X \leftarrow \text{RSLMethIter}(\mathbb{A}, \mathbf{b}, \text{IterJacobi}, \mathbf{x}^0, \varepsilon, \text{kmax})$   
**Fin Fonction**

**Fonction**  $X \leftarrow \text{RSLGaussSeidel3}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$   
 $X \leftarrow \text{RSLMethIter}(\mathbb{A}, \mathbf{b}, \text{IterGaussSeidel}, \mathbf{x}^0, \varepsilon, \text{kmax})$   
**Fin Fonction**

---

**Algorithme** Méthode itérative pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$

---

**Données :**

$\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,  
 $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ ,  
**IterFonc** : fonction de paramètres une matrice d'ordre  $n$ ,  
et deux vecteurs de  $\mathbb{K}^n$ . retourne un vecteur de  $\mathbb{K}^n$ .  
 $\mathbf{x}^0$  : vecteur initial de  $\mathbb{K}^n$ ,  
 $\varepsilon$  : la tolérance,  $\varepsilon \in \mathbb{R}^+$ ,  
kmax : nombre maximum d'itérations, kmax  $\in \mathbb{N}^*$

**Résultat :**

$\mathbf{x}^{\text{tol}}$  : un vecteur de  $\mathbb{K}^n$  si convergence, sinon  $\emptyset$

```

1: Fonction  $X \leftarrow \text{RSLMethIter}(\mathbb{A}, \mathbf{b}, \text{IterFonc}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
2:    $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
3:    $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
4:    $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
5:   Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
6:      $k \leftarrow k + 1$ 
7:      $\mathbf{p} \leftarrow \mathbf{x}$ 
8:      $\mathbf{x} \leftarrow \text{IterFonc}(\mathbb{A}, \mathbf{b}, \mathbf{p})$ 
9:      $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
10:  Fin Tantque
11:  Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
12:     $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
13:  Fin Si
14: Fin Fonction

```

---



Méthode de relaxation utilisant Gauss-Seidel, avec  $w \in \mathbb{R}^*$ ,

$$x_i^{[k+1]} = \frac{w}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij} x_j^{[k]} \right) + (1-w)x_i^{[k]} \quad \forall i \in \llbracket 1, n \rrbracket$$

---

**Algorithme** Itération S.O.R.

---

**Données :**

A : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,  
b : vecteur de  $\mathbb{K}^n$ ,  
y : vecteur de  $\mathbb{K}^n$ ,  
w : réel non nul.

**Résultat :**

x : un vecteur de  $\mathbb{K}^n$

```
1: Fonction x ← IterSOR( A, b, y, w )
2:   Pour i ← 1 à n faire
3:     S ← 0
4:     Pour j ← 1 à i - 1 faire
5:       S ← S - A(i, j) * x(j)
6:     Fin Pour
7:     Pour j ← i + 1 à n faire
8:       S ← S - A(i, j) * y(j)
9:     Fin Pour
10:    x(i) ← w * (b(i) - S) / A(i, i) + (1 - w) * y(i)
11:  Fin Pour
12: Fin Fonction
```

Paramètre w "en trop" dans l'appel de la fonction **IterSOR** pour pouvoir utiliser la fonction générique **RSLMethIter!**

Méthode de relaxation utilisant Gauss-Seidel, avec  $w \in \mathbb{R}^*$ ,

$$x_i^{[k+1]} = \frac{w}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij} x_j^{[k]} \right) + (1-w) x_i^{[k]} \quad \forall i \in \llbracket 1, n \rrbracket$$

---

**Algorithme** Itération S.O.R.

---

**Données :**

A : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,  
b : vecteur de  $\mathbb{K}^n$ ,  
y : vecteur de  $\mathbb{K}^n$ ,  
w : réel non nul.

**Résultat :**

x : un vecteur de  $\mathbb{K}^n$

```
1: Fonction x ← IterSOR( A, b, y, w )
2:   Pour i ← 1 à n faire
3:     S ← 0
4:     Pour j ← 1 à i - 1 faire
5:       S ← S - A(i,j) * x(j)
6:     Fin Pour
7:     Pour j ← i + 1 à n faire
8:       S ← S - A(i,j) * y(j)
9:     Fin Pour
10:    x(i) ← w * (b(i) - S) / A(i,i) + (1 - w) * y(i)
11:  Fin Pour
12: Fin Fonction
```

Paramètre w "en trop" dans l'appel de la fonction **IterSOR** pour pouvoir utiliser la fonction générique **RSLMethIter!**

**Fonction** X ← **RLSOR3**( A, b, w, x<sup>0</sup>, ε, kmax )

IterFun ← ((M, r, s) ↦ IterSOR(M, r, s, w))

X ← **RSLMethIter**(A, b, IterFun, x<sup>0</sup>, ε, kmax)

**Fin Fonction**

- 1 Conditionnement
- 2 Méthodes directes
- 3 **Méthodes itératives**
  - Principe et résultats généraux
  - Notations
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Méthode de relaxation
- Etude de la convergence
- Algorithmes scalaires
  - Principe de base
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Jeux algorithmiques
  - Méthode S.O.R.
- **Algorithmes matriciels**
- Autres méthodes

Les méthodes de Jacobi, Gauss-Seidel et S.O.R. peuvent s'écrire sous la forme

$$\mathbb{M}\mathbf{x}^{[k+1]} = \mathbb{N}\mathbf{x}^{[k]} + \mathbf{b}$$

avec  $\mathbb{A} = \mathbb{M} - \mathbb{N}$  et, dans ce cas, la matrice d'itération est  $\mathbb{B} = \mathbb{M}^{-1}\mathbb{N}$ .

- Jacobi :  $\mathbb{M} = \mathbb{D}$  et  $\mathbb{N} = \mathbb{E} + \mathbb{F}$ ,
- Gauss-Seidel :  $\mathbb{M} = \mathbb{D} - \mathbb{E}$  et  $\mathbb{N} = \mathbb{F}$ ,
- S.O.R. :  $\mathbb{M} = \frac{\mathbb{D}}{w} - \mathbb{E}$  et  $\mathbb{N} = \frac{1-w}{w}\mathbb{D} + \mathbb{F}$ .

En posant,  $\Phi(\mathbf{x}) = \mathbb{M}^{-1}(\mathbb{N}\mathbf{x} + \mathbf{b})$ , on a

$$\mathbf{x}^{[k+1]} = \Phi(\mathbf{x}^{[k]}).$$

⇒ on peut utiliser l'algorithme vectoriel du point fixe

---

**Algorithme** Méthode de point fixe vectorielle

---

**Données :**

$\Phi$  :  $\Phi : \mathbb{K}^N \longrightarrow \mathbb{K}^N$ ,  
 $\mathbf{x0}$  : donnée initiale,  $\mathbf{x0} \in \mathbb{K}^N$ ,  
 $\text{tol}$  : la tolérance,  $\text{tol} \in \mathbb{R}^+$ ,  
 $\text{kmax}$  : nombre maximum d'itérations,  $\text{kmax} \in \mathbb{N}^*$

**Résultat :**

$\alpha_{\text{tol}}$  : un réel tel que  $\|\Phi(\alpha_{\text{tol}}) - \alpha_{\text{tol}}\| \leq \text{tol}$

```
1: Fonction  $\alpha_{\text{tol}} \leftarrow \text{PtFixeVec}(\Phi, \mathbf{x0}, \text{tol}, \text{kmax})$   
2:    $k \leftarrow 0, \alpha_{\text{tol}} \leftarrow \emptyset$   
3:    $\mathbf{x} \leftarrow \mathbf{x0}, \mathbf{fx} \leftarrow \Phi(\mathbf{x0})$ ,  
4:    $\text{err} \leftarrow \|\mathbf{fx} - \mathbf{x}\|$   
5:   Tantque  $\text{err} > \text{tol}$  et  $k \leq \text{kmax}$  faire  
6:      $k \leftarrow k + 1$   
7:      $\mathbf{x} \leftarrow \mathbf{fx}$   
8:      $\mathbf{fx} \leftarrow \Phi(\mathbf{x})$   
9:      $\text{err} \leftarrow \|\mathbf{fx} - \mathbf{x}\|$   
10:  Fin Tantque  
11:  Si  $\text{err} \leq \text{tol}$  alors  
12:     $\alpha_{\text{tol}} \leftarrow \mathbf{x}$   
13:  Fin Si  
14: Fin Fonction
```

---

$$\Phi(\mathbf{x}) = \mathbb{M}^{-1}(\mathbb{N}\mathbf{x} + \mathbf{b})$$

$$\mathbf{y} = \Phi(\mathbf{x}) \iff \mathbb{M}\mathbf{y} = \mathbb{N}\mathbf{x} + \mathbf{b}.$$

- Jacobi :  $\mathbb{M} = \mathbb{D}$  (diagonale) et  $\mathbb{N} = \mathbb{E} + \mathbb{F}$ ,

$$\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLMatDiag}(\mathbb{M}, \mathbb{N}\mathbf{x} + \mathbf{b}) \right)$$

- Gauss-Seidel :  $\mathbb{M} = \mathbb{D} - \mathbb{E}$  (tri. inf.) et  $\mathbb{N} = \mathbb{F}$ ,

$$\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLTriInf}(\mathbb{M}, \mathbb{N}\mathbf{x} + \mathbf{b}) \right)$$

- S.O.R. :  $\mathbb{M} = \frac{\mathbb{D}}{w} - \mathbb{E}$  (tri. inf.) et  $\mathbb{N} = \frac{1-w}{w}\mathbb{D} + \mathbb{F}$ ,

$$\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLTriInf}\left(\mathbb{M}, \mathbb{N}\mathbf{x} + \mathbf{b}\right) \right).$$

$$\triangleright \text{ ou } \frac{\|\mathbf{fx} - \mathbf{x}\|}{\|\mathbf{x}\| + 1}$$

$$\triangleright \text{ ou } \frac{\|\mathbf{fx} - \mathbf{x}\|}{\|\mathbf{x}\| + 1}$$

---

**Algorithme** Méthode de point fixe vectorielle

---

**Données :**

$\Phi$  :  $\Phi : \mathbb{K}^N \longrightarrow \mathbb{K}^N$ ,  
 $\mathbf{x0}$  : donnée initiale,  $\mathbf{x0} \in \mathbb{K}^N$ ,  
tol : la tolérance,  $\text{tol} \in \mathbb{R}^+$ ,  
kmax : nombre maximum d'itérations,  $\text{kmax} \in \mathbb{N}^*$

**Résultat :**

$\alpha_{\text{tol}}$  : un réel tel que  $\|\Phi(\alpha_{\text{tol}}) - \alpha_{\text{tol}}\| \leq \text{tol}$

```
1: Fonction  $\alpha_{\text{tol}} \leftarrow \text{PtFixeVec}(\Phi, \mathbf{x0}, \text{tol}, \text{kmax})$ 
2:    $k \leftarrow 0, \alpha_{\text{tol}} \leftarrow \emptyset$ 
3:    $\mathbf{x} \leftarrow \mathbf{x0}, \mathbf{fx} \leftarrow \Phi(\mathbf{x0})$ ,
4:    $\text{err} \leftarrow \|\mathbf{fx} - \mathbf{x}\|$ 
5:   Tantque  $\text{err} > \text{tol}$  et  $k \leq \text{kmax}$  faire
6:      $k \leftarrow k + 1$ 
7:      $\mathbf{x} \leftarrow \mathbf{fx}$ 
8:      $\mathbf{fx} \leftarrow \Phi(\mathbf{x})$ 
9:      $\text{err} \leftarrow \|\mathbf{fx} - \mathbf{x}\|$ 
10:  Fin Tantque
11:  Si  $\text{err} \leq \text{tol}$  alors
12:     $\alpha_{\text{tol}} \leftarrow \mathbf{x}$ 
13:  Fin Si
14: Fin Fonction
```

---

**Fonction**  $\mathbf{X} \leftarrow \text{RSLJacobiPF}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$

$[\mathbb{M}, \mathbb{N}] \leftarrow \text{JacobiMN}(\mathbb{A})$

$\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLMatDiag}(\mathbb{M}, \mathbb{N}\mathbf{x} + \mathbf{b}) \right)$

$\mathbf{X} \leftarrow \text{PtFixeVec}(\Phi, \mathbf{x0}, \text{tol}, \text{kmax})$

**Fin Fonction**

**Fonction**  $\mathbf{X} \leftarrow \text{RSLGaussSeidelPF}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$

$[\mathbb{M}, \mathbb{N}] \leftarrow \text{GaussSeidelMN}(\mathbb{A})$

$\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLTriInf}(\mathbb{M}, \mathbb{N}\mathbf{x} + \mathbf{b}) \right)$

$\mathbf{X} \leftarrow \text{PtFixeVec}(\Phi, \mathbf{x0}, \text{tol}, \text{kmax})$

**Fin Fonction**

**Fonction**  $\mathbf{X} \leftarrow \text{RSLSORPF}(\mathbb{A}, \mathbf{b}, w, \mathbf{x}^0, \varepsilon, \text{kmax})$

$[\mathbb{M}, \mathbb{N}] \leftarrow \text{SORmatMN}(\mathbb{A}, w)$

$\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLTriInf}(\mathbb{M}, \mathbb{N}\mathbf{x} + \mathbf{b}) \right)$

$\mathbf{X} \leftarrow \text{PtFixeVec}(\Phi, \mathbf{x0}, \text{tol}, \text{kmax})$

**Fin Fonction**

$\triangleright$  ou  $\frac{\|\mathbf{fx} - \mathbf{x}\|}{\|\mathbf{x}\| + 1}$

$\triangleright$  ou  $\frac{\|\mathbf{fx} - \mathbf{x}\|}{\|\mathbf{x}\| + 1}$

- 1 Conditionnement
- 2 Méthodes directes
- 3 **Méthodes itératives**
  - Principe et résultats généraux
  - Notations
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Méthode de relaxation
- Etude de la convergence
- Algorithmes scalaires
  - Principe de base
  - Méthode de Jacobi
  - Méthode de Gauss-Seidel
  - Jeux algorithmiques
  - Méthode S.O.R.
- Algorithmes matriciels
- **Autres méthodes**

Nous venons de voir trois méthodes itératives classiques. Nous pouvons citer d'autres méthodes

- Méthode des directions alternées (Douglas, Peaceman, Rachford 1955)
- Méthodes de Richardson (pas constant, pas variable, préconditionnées, ...)
- Méthodes de Gradient Conjugué et dérivées: CG, CGS, BICG, BICGSTABL, ...
- *Generalized Minimal Residual method* (GMRES) et dérivées, ...
- ...