

Analyse Numérique I

Sup'Galilée, Ingénieurs MACS, 1ère année

François Cuvelier

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

2022/10/28

Chapitre IV

Résolution de systèmes linéaires

- 1 Conditionnement
- 2 Méthodes directes
- 3 Méthodes itératives

- Principe
- Notations
- Méthode de Jacobi
- Méthode de Gauss-Seidel

- Méthode de relaxation
- Etude de la convergence
- Algorithmes
 - Principe de base
 - Méthode de Jacobi
 - Méthode de Gauss-Seidel
 - Jeux algorithmiques
 - Méthode S.O.R.

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

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

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

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
 - Notations
 - Méthode de Jacobi
 - Méthode de Gauss-Seidel
- Méthode de relaxation
 - Etude de la convergence
 - Algorithmes
 - Principe de base
 - Méthode de Jacobi
 - Méthode de Gauss-Seidel
 - Jeux algorithmiques
 - Méthode S.O.R.

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

La méthode itérative de **Jacobi** :

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

ou encore

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

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

La méthode itérative de **Jacobi** :

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

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

ou encore

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

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

- 1 Conditionnement
- 2 Méthodes directes
- 3 Méthodes itératives
 - Principe
 - Notations
 - Méthode de Jacobi
 - Méthode de Gauss-Seidel

- Méthode de relaxation
- Etude de la convergence
- Algorithmes
 - Principe de base
 - Méthode de Jacobi
 - Méthode de Gauss-Seidel
 - Jeux algorithmiques
 - Méthode S.O.R.

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

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

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

ou encore

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

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

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

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

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

ou encore

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

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

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

- **Méthode de relaxation**
- Etude de la convergence
- Algorithmes
 - Principe de base
 - Méthode de Jacobi
 - Méthode de Gauss-Seidel
 - Jeux algorithmiques
 - Méthode S.O.R.

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)

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



Exercice 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} .



Proposition:



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

La matrice d'itération de la méthode de Jacobi, notée \mathbb{J} , est donnée par

$$\mathbb{J} = \mathbb{D}^{-1}(\mathbb{E} + \mathbb{F}) = \mathbb{L} + \mathbb{U}, \quad (1)$$

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

$$\mathcal{L}_w = \left(\frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \left(\frac{1-w}{w} \mathbb{D} + \mathbb{F} \right) = (\mathbb{I} - w\mathbb{L})^{-1} ((1-w)\mathbb{I} + w\mathbb{U}). \quad (2)$$

et elle vérifie

$$\rho(\mathcal{L}_w) \geq |w - 1|. \quad (3)$$

La matrice d'itération de Gauss-Seidel est \mathcal{L}_1 et elle correspond à

$$\mathcal{L}_1 = (\mathbb{D} - \mathbb{E})^{-1}\mathbb{F} = (\mathbb{I} - \mathbb{L})^{-1}\mathbb{U}. \quad (4)$$

- 1 Conditionnement
- 2 Méthodes directes
- 3 Méthodes itératives

- Principe
- Notations
- Méthode de Jacobi
- Méthode de Gauss-Seidel

- Méthode de relaxation
- **Etude de la convergence**
- Algorithmes
 - Principe de base
 - Méthode de Jacobi
 - Méthode de Gauss-Seidel
 - Jeux algorithmiques
 - Méthode S.O.R.



Théorème:



Soit \mathbb{A} une matrice régulière décomposée sous la forme $\mathbb{A} = \mathbb{M} - \mathbb{N}$ avec \mathbb{M} régulière. 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 $\rho(\mathbb{B}) < 1$.

Lien avec les méthodes de Jacobi, Gauss-Seidel, S.O.R.?



Corollaire:



Soit A une matrice vérifiant $A_{i,i} \neq 0 \forall i$. Une condition nécessaire de convergence pour la méthode S.O.R. est que $0 < w < 2$.



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

Soit 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 de Relaxation est convergente.



Théorème:



Soit A une matrice hermitienne inversible en décomposée en $A = M - N$ où M est inversible. Soit $B = I - M^{-1}A$, la matrice de l'itération. Supposons que $M^* + N$ (qui est hermitienne) soit définie positive. Alors $\rho(B) < 1$ si et seulement si A est définie positive.



Théorème

Soit A une matrice hermitienne définie positive, alors la méthode de relaxation converge si et seulement si $w \in]0, 2[$.

Preuve : voir Lascaux-Théodor, *Analyse numérique matricielle appliquée à l'art de l'ingénieur*, vol.2, Corollaire 24, page 351.

- 1 Conditionnement
- 2 Méthodes directes
- 3 Méthodes itératives
 - Principe
 - Notations
 - Méthode de Jacobi
 - Méthode de Gauss-Seidel

- Méthode de relaxation
- Etude de la convergence
- **Algorithmes**
 - Principe de base
 - Méthode de Jacobi
 - Méthode de Gauss-Seidel
 - Jeux algorithmiques
 - Méthode S.O.R.

Principe de base

Résoudre :

$$Ax = b$$

Méthodes itératives :

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

Algorithme :

$x^{[0]}$ donné

Pour $k = 0, 1, \dots$ **faire**

$$x^{[k+1]} \leftarrow Bx^{[k]} + c$$

Fin Pour

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

La convergence de ces méthodes n'est pas assurées 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?

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.

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

La convergence de ces méthodes n'est pas assurées 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 $\mathbf{e}^{[k]} = \bar{\mathbf{x}} - \mathbf{x}^{[k]}$

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

Algorithm 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 ,
- ε : la tolérance, $\varepsilon \in \mathbb{R}^+$,
- kmax : nombre maximum d'itérations, kmax $\in \mathbb{N}^*$

Résultat :

- \mathbf{x}^{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**

▷ \mathbf{p} contient le vecteur précédent

▷ Convergence

Pour Jacobi , la suite des itérées est définie par

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

Algorithme 2 \mathcal{R}_0

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

Algorithme 2 \mathcal{R}_1

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

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**
 8:
 9:
$$x_i \leftarrow \frac{1}{A_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n A_{ij} p_j \right)$$

 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**

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: $S \leftarrow S + A_{i,j} p_j$
 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**

Algorithm 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 ,
- ε : 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
```

Pour Gauss-Seidel , la suite des itérées est définie par

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

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 $k \leq k_{\text{max}}$ **faire**
- 5: $k \leftarrow k + 1$
- 6: $\mathbf{p} \leftarrow \mathbf{x}$
- 7:
- 8: $\mathbf{x} \leftarrow$ 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 $k \leq k_{\text{max}}$ **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**

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 $k \leq k_{\text{max}}$ **faire**
 5: $k \leftarrow k + 1$
 6: $\mathbf{p} \leftarrow \mathbf{x}$
 7: **Pour** $i \leftarrow 1$ à n **faire**
 8:
 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**

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 $k \leq k_{\text{max}}$ **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**

Algorithm Méthode itérative de Gauss-Seidel 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 ,
 ε : 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  $\mathbf{X} \leftarrow \text{RSLGAUSSSEIDEL} (\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
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  $i - 1$  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:   $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x}$ ,
19: Fin Tantque
20: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
21:    $\mathbf{X} \leftarrow \mathbf{x}$ 
22: Fin Si
23: Fin Fonction
```

Fonction $X \leftarrow \text{RSLJACOBI}$ ($A, b, x^0, \varepsilon, k_{\max}$)

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

$x \leftarrow x^0, r \leftarrow b - A * x,$

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

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

$k \leftarrow k + 1$

$p \leftarrow x$

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

$r \leftarrow b - A * x,$

Fin Tantque

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

$X \leftarrow x$

Fin Si

Fin Fonction

Fonction $X \leftarrow \text{RSLGAUSSSEIDEL}$ ($A, b, x^0, \varepsilon, k_{\max}$)

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

$x \leftarrow x^0, r \leftarrow b - A * x,$

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

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

$k \leftarrow k + 1$

$p \leftarrow x$

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

$r \leftarrow b - A * x,$

Fin Tantque

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

$X \leftarrow x$

Fin Si

Fin Fonction

```

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

```

```

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

```

Même ossature puisque toutes deux basées sur l'Algorithme générique

Peut-on simplifier, clarifier et raccourcir les codes?

Fonction $X \leftarrow \text{RSLJACOBI}$ ($A, b, x^0, \varepsilon, k_{\max}$)

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

$x \leftarrow x^0, r \leftarrow b - A * x,$

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

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

$k \leftarrow k + 1$

$p \leftarrow x$

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

$r \leftarrow b - A * x,$

Fin Tantque

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

$X \leftarrow x$

Fin Si

Fin Fonction

Algorithm Itération de Jacobi : calcul de 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**

Fonction $X \leftarrow \text{RSLJACOBI2} (A, b, x^0, \varepsilon, k_{\max})$

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

$x \leftarrow x^0, r \leftarrow b - A * x,$

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

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

$k \leftarrow k + 1$

$p \leftarrow x$

$x \leftarrow \text{ITERJACOBI}(A, b, p)$

$r \leftarrow b - A * x,$

Fin Tantque

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

$X \leftarrow x$

Fin Si

Fin Fonction

Algorithm Itération de Jacobi : calcul de 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**

Fonction $X \leftarrow \text{RSLGAUSSSEIDEL} (A, b, x^0, \varepsilon, k_{\max})$

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

$x \leftarrow x^0, r \leftarrow b - A * x,$

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

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

$k \leftarrow k + 1$

$p \leftarrow x$

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

$r \leftarrow b - A * x,$

Fin Tantque

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

$X \leftarrow x$

Fin Si

Fin Fonction

Algorithm Itération de Gauss-Seidel : calcul de 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 :

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** $x \leftarrow \text{ITERGAUSSSEIDEL} (A, b, y)$

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 $\mathbf{X} \leftarrow \text{RSLGAUSSSEIDEL2}$ ($\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax}$)

$k \leftarrow 0, \mathbf{X} \leftarrow \emptyset$

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

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

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

$\mathbf{X} \leftarrow \mathbf{x}$

Fin Si

Fin Fonction

Algorithm 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 :

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** $\mathbf{x} \leftarrow \text{ITERGAUSSSEIDEL}$ ($\mathbb{A}, \mathbf{b}, \mathbf{y}$)
 - 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?

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

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

Algorithm 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 ,
- ε : la tolérance, $\varepsilon \in \mathbb{R}^+$,
- kmax : nombre maximum d'itérations, kmax $\in \mathbb{N}^*$

Résultat :

- \mathbf{x}^{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

Algorithm 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 ,
 ε : la tolérance, $\varepsilon \in \mathbb{R}^+$,
 kmax : nombre maximum d'itérations, kmax $\in \mathbb{N}^*$

Résultat :

\mathbf{x}^{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 $w \in \mathbb{R}^*$,

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

Algorithm 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 $w \in \mathbb{R}^*$,

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

Algorithm 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, x0, ε, kmax )
  ITERFUN ← ((M, r, s) → ITERSOR(M, r, s, w))
  X ← RSLMETHITER(A, b, ITERFUN, x0, ε, kmax)
Fin Fonction
```