Analyse Numérique I*

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

François Cuvelier

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

2024/10/26

*Compilé le 2024/10/26 à 12:15:30

Chapitre IV

Résolution de systèmes linéaires

Plan

Méthodes itératives pour la résolution du système linéaire

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

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

$$\boldsymbol{x}^{[k+1]} = \mathbb{B}\boldsymbol{x}^{[k]} + \boldsymbol{c},\ k \geqslant 0,\ \boldsymbol{x}^{[0]} \text{ arbitraire}$$

vérifie

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

Exercice 1

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

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

Montrer que la suite définie par

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

converge vers $\bar{\boldsymbol{x}} = \mathbb{A}^{-1}\boldsymbol{b}$ quelque soit $\boldsymbol{x}^{[0]}$ 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

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

Alors la suite définie par

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

converge vers $\bar{\boldsymbol{x}} = \mathbb{A}^{-1}\boldsymbol{b}$ quelque soit $\boldsymbol{x}^{[0]}$ 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

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

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

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

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

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

et on obtient

$$\left\|\boldsymbol{E}^{[k]}\right\|_p \leqslant \|\mathbb{D}\|_p^k \left\|\boldsymbol{E}^{[0]}\right\|_p = \rho(\mathbb{B})^k \left\|\boldsymbol{E}^{[0]}\right\|_p.$$

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

Q. 1

Montrer que la matrice $\mathbb{M}^* + \mathbb{N}$ est hermitienne.

On suppose maintenant que $\mathbb{M}^* + \mathbb{N}$ est définie positive.

Q. 2

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

① *Montrer que*

$$\langle \boldsymbol{x}, \mathbb{A}\boldsymbol{x}\rangle - \langle \boldsymbol{y}, \mathbb{A}\boldsymbol{y}\rangle = \langle \boldsymbol{x}, \mathbb{A}\mathbb{M}^{-1}\mathbb{A}\boldsymbol{x}\rangle + \langle \mathbb{M}^{-1}\mathbb{A}\boldsymbol{x}, \mathbb{A}\boldsymbol{x}\rangle - \langle \mathbb{M}^{-1}\mathbb{A}\boldsymbol{x}, \mathbb{A}\mathbb{M}^{-1}\mathbb{A}\boldsymbol{x}\rangle \tag{1}$$

et

$$\boldsymbol{x} - \boldsymbol{y} = \mathbb{M}^{-1}\mathbb{A}\boldsymbol{x}. \tag{2}$$

② *En déduire que*

$$\langle \boldsymbol{x}, \mathbb{A}\boldsymbol{x}\rangle - \langle \boldsymbol{y}, \mathbb{A}\boldsymbol{y}\rangle = \langle (\boldsymbol{x} - \boldsymbol{y}), (\mathbb{M}^* + \mathbb{N})(\boldsymbol{x} - \boldsymbol{y})\rangle. \tag{3}$$

Q. 3

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

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

Q. 4

On suppose qu'il existe $\boldsymbol{x}^{[0]} \in \mathbb{C}^n \backslash \{0\}$ tel que $\alpha_0 \overset{\text{def}}{=} \langle \boldsymbol{x}^{[0]}, \mathbb{A}\boldsymbol{x}^{[0]}\rangle \in \mathbb{C}\backslash]0, +\infty[$. On défini alors les suites

$$\forall k \in \mathbb{N}^*, \ \boldsymbol{x}^{[k]} = \mathbb{B}\boldsymbol{x}^{[k-1]} \ \text{ et } \ \alpha_k = \left\langle \boldsymbol{x}^{[k]}, \mathbb{A}\boldsymbol{x}^{[k]}\right\rangle.$$

① *Montrer que*

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

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

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

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

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

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

Plan

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ une matrice régulière, avec $\forall i \in [\![1,n]\!],\ \mathrm{A}_{i,i} \neq 0$

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

Plan

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

La méthode itérative de **Jacobi** :

$$\forall i \in [\![1,n]\!],\quad b_i = \sum_{j=1}^{i-1} \mathrm{A}_{i,j} x_j^{[k]} + \mathrm{A}_{i,i} x_i^{[k+1]} + \sum_{j=i+1}^{n} \mathrm{A}_{i,j} x_j^{[k]}$$

ou encore

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

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

La méthode itérative de **Jacobi** :

$$\forall i \in [\![1,n]\!],\quad b_i = \sum_{j=1}^{i-1} \mathrm{A}_{i,j}x_j^{[k]} + \mathrm{A}_{i,i}x_i^{[k+1]} + \sum_{j=i+1}^{n} \mathrm{A}_{i,j}x_j^{[k]}$$

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

ou encore

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

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

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

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

Plan

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

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

$$\forall i \in [\![1,n]\!],\quad b_i = \mathrm{A}_{i,i}x_i^{[k+1]} + \sum_{j=1}^{i-1} \mathrm{A}_{i,j}x_j^{[k+1]} + \sum_{j=i+1}^{n} \mathrm{A}_{i,j}x_j^{[k]} \quad \forall i \in [\![1,n]\!]$$

ou encore

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

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

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

$$\forall i \in [\![1,n]\!],\quad b_i = \mathrm{A}_{i,i}x_i^{[k+1]} + \sum_{j=1}^{i-1} \mathrm{A}_{i,j}x_j^{[k+1]} + \sum_{j=i+1}^{n} \mathrm{A}_{i,j}x_j^{[k]} \quad \forall i \in [\![1,n]\!]$$

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

ou encore

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

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

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

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

Plan

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 $\mathrm{A}_{i,j}$ 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 méthode S.O.R. (successive over relaxation) est donnée par

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

Q. 1 *Déterminer la matrice d'itération $\mathbb{B}$ et le vecteur $\boldsymbol{c}$ tels que*

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

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

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

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

ou vectoriellement

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

Sa matrice d'itération est

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

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

Plan

Avec les notations du Théorème 3.1, on a

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

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

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

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

- S.O.R. : $\mathcal{L}_w \overset{\text{def}}{=} \left(\frac{\mathbb{D}}{w}-\mathbb{E}\right)^{-1}\left(\frac{1-w}{w}\mathbb{D}+\mathbb{F}\right)$ 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 $\mathrm{A}_{i,j}$ 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

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

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*

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

Proposition 3.1

Soit $\mathbb{A}$ une matrice inversible telle que tous ses éléments diagonaux soient non nuls. On note $\mathbb{D}=\mathrm{diag}(\mathbb{A})$ 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

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

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

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

Q. 1
- *Soit $\mu\in\mathbb{C}^*$. On note $\mathbb{Q}(\mu)\in\mathcal{M}_n(\mathbb{C})$ la matrice diagonale de diagonale $(\mu,\mu^2,\dots,\mu^n)$.*
- *Expliciter la matrice $\mathbb{T}(\mu)\overset{\text{def}}{=}\mathbb{Q}(\mu)\mathbb{T}\mathbb{Q}^{-1}(\mu)$ en fonction des coefficients tridiagonaux de la matrice $\mathbb{T}$ et de μ.*
- *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 $\mathrm{A}_{i,j}$ 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 $\mathbb{J}\overset{\text{def}}{=}\mathbb{D}^{-1}(\mathbb{E}+\mathbb{F})$ et $\mathcal{L}_1\overset{\text{def}}{=}(\mathbb{D}-\mathbb{E})^{-1}\mathbb{F}$ les matrices d'itérations des méthodes de Jacobi et de Gauss-Seidel.
Soit $\boldsymbol{y}\in\mathbb{R}^n$. On souhaite résoudre le système $\mathbb{A}\boldsymbol{x}=\boldsymbol{y}$ 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

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

et que ses éléments diagonaux sont non nuls.

Q. 2
- *Montrer que les valeurs propres de $\mathbb{J}$ sont les racines du polynôme*

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

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

Q. 3
- *Montrer que les valeurs propres de $\mathcal{L}_1$ sont les racines du polynôme*

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

- *En déduire que*

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

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

Proposition 3.2 :

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

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

et donné par

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

On a alors

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

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

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

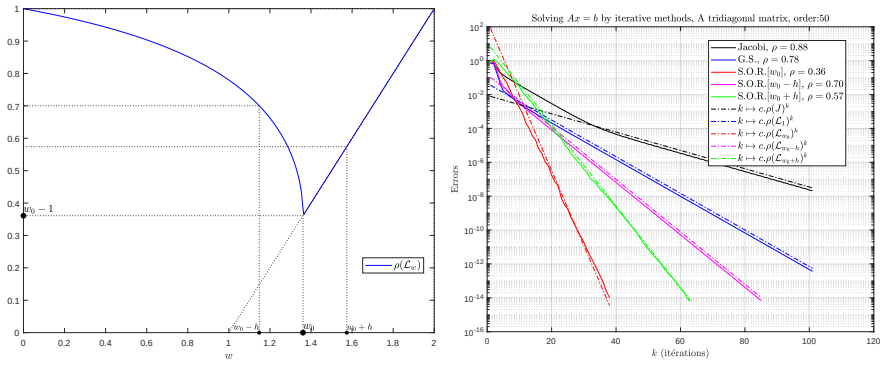

Figure: $\mathbb{A}$ est d'ordre $n = 50$.

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

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

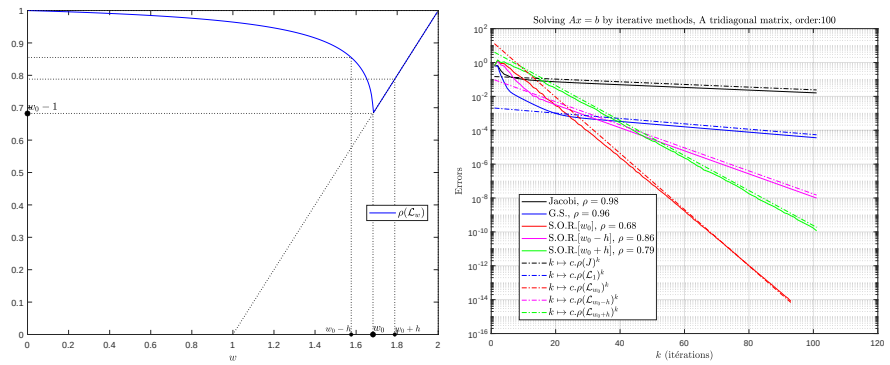

Figure: $\mathbb{A}$ est d'ordre $n = 100$.

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 $w \in]0,2[$.

Plan

Principe de base

Résoudre :

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

Méthodes itératives :

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

Algorithme :

$\boldsymbol{x}^{[0]}$ donné
Pour $k = 0,\ 1, \cdots$ **faire**
  $\boldsymbol{x}^{[k+1]} \leftarrow \mathbb{B}\boldsymbol{x}^{[k]} + \boldsymbol{c}$
Fin Pour

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

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.

$\Longrightarrow$ boucle **Tantque**

Critères d'arrêt :

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

Comment choisir le critère d'arrêt pour la convergence?

Exemple de critère d'arrêt pour la convergence :
Soit $\boldsymbol{r}^{[k]} = \boldsymbol{b} - \mathbb{A}\boldsymbol{x}^{[k]}$ le résidu.

$$\frac{\|\boldsymbol{r}^{[k]}\|}{\|\boldsymbol{b}\|} \leqslant \varepsilon$$

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

$$\frac{\|\boldsymbol{e}^{[k]}\|}{\|\bar{\boldsymbol{x}}\|} \leqslant \varepsilon\, \mathrm{cond}(\mathbb{A})$$

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

$$\frac{\|\boldsymbol{r}^{[k]}\|}{\|\boldsymbol{b}\| + 1} \leqslant \varepsilon$$

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

Données :

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

Résultat :

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

1: $k \leftarrow 0$, $\boldsymbol{x}^{\text{tol}} \leftarrow \varnothing$
2: $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
3: $\text{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
4: **Tantque** $\|\boldsymbol{r}\| > \text{tol}$ et $k \leqslant \text{kmax}$ **faire**
5: $k \leftarrow k + 1$
6: $\boldsymbol{p} \leftarrow \boldsymbol{x}$ ▷ $\boldsymbol{p}$ contient le vecteur précédent
7: $\boldsymbol{x} \leftarrow$ calcul de l'itérée suivante en fonction de $\boldsymbol{p}$, $\mathbb{A}$, $\boldsymbol{b}$, ...
8: $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
9: **Fin Tantque**
10: **Si** $\|\boldsymbol{r}\| \leqslant \text{tol}$ **alors** ▷ Convergence
11: $\boldsymbol{x}^{\text{tol}} \leftarrow \boldsymbol{x}$
12: **Fin Si**

Jacobi: $x_i^{[k+1]} = \dfrac{1}{\mathrm{A}_{ii}}\left(b_i - \displaystyle\sum_{j=1, j\neq i}^{n} \mathrm{A}_{ij} x_j^{[k]}\right), \quad \forall i \in [\![1, n]\!].$

Algorithme 2 $\mathcal{R}_0$

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

Algorithme 2 $\mathcal{R}_1$

1: $k \leftarrow 0$, $\boldsymbol{x}^{\text{tol}} \leftarrow \varnothing$
2: $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
3: $\text{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
4: **Tantque** $\|\boldsymbol{r}\| > \text{tol}$ et $k \leqslant \text{kmax}$ **faire**
5: $k \leftarrow k + 1$
6: $\boldsymbol{p} \leftarrow \boldsymbol{x}$
7: **Pour** $i \leftarrow 1$ **à** n **faire**
8: $x_i \leftarrow \dfrac{1}{\mathrm{A}_{ii}}\left(b_i - \displaystyle\sum_{j=1, j\neq i}^{n} \mathrm{A}_{ij}\boldsymbol{p}_j\right)$
9: **Fin Pour**
10: $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
11: **Fin Tantque**
12: **Si** $\|\boldsymbol{r}\| \leqslant \text{tol}$ **alors** ▷ Convergence
13: $\boldsymbol{x}^{\text{tol}} \leftarrow \boldsymbol{x}$
14: **Fin Si**

Jacobi: $x_i^{[k+1]} = \dfrac{1}{\mathrm{A}_{ii}}\left(b_i - \displaystyle\sum_{j=1, j\neq i}^{n} \mathrm{A}_{ij} x_j^{[k]}\right), \quad \forall i \in [\![1, n]\!].$

Algorithme 2 $\mathcal{R}_1$

1: $k \leftarrow 0$, $\boldsymbol{x}^{\text{tol}} \leftarrow \varnothing$
2: $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
3: $\text{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
4: **Tantque** $\|\boldsymbol{r}\| > \text{tol}$ et $k \leqslant \text{kmax}$ **faire**
5: $k \leftarrow k + 1$
6: $\boldsymbol{p} \leftarrow \boldsymbol{x}$
7: **Pour** $i \leftarrow 1$ **à** n **faire**
8: $x_i \leftarrow \dfrac{1}{\mathrm{A}_{ii}}\left(b_i - \displaystyle\sum_{j=1, j\neq i}^{n} \mathrm{A}_{ij}\boldsymbol{p}_j\right)$
9: **Fin Pour**
10: $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
11: **Fin Tantque**
12: **Si** $\|\boldsymbol{r}\| \leqslant \text{tol}$ **alors**
13: $\boldsymbol{x}^{\text{tol}} \leftarrow \boldsymbol{x}$
14: **Fin Si**

Algorithme 2 $\mathcal{R}_2$

1: $k \leftarrow 0$, $\boldsymbol{x}^{\text{tol}} \leftarrow \varnothing$
2: $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
3: $\text{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
4: **Tantque** $\|\boldsymbol{r}\| > \text{tol}$ et $k \leqslant \text{kmax}$ **faire**
5: $k \leftarrow k + 1$
6: $\boldsymbol{p} \leftarrow \boldsymbol{x}$
7: **Pour** $i \leftarrow 1$ **à** n **faire**
8: $\mathrm{S} \leftarrow 0$
9: **Pour** $j \leftarrow 1$ **à** n $(j \neq i)$ **faire**
10: $\mathrm{S} \leftarrow \mathrm{S} + \mathrm{A}_{i,j}\boldsymbol{p}_j$
11: **Fin Pour**
12: $x_i \leftarrow \dfrac{1}{\mathrm{A}_{ii}}(b_i - \mathrm{S})$
13: **Fin Pour**
14: $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
15: **Fin Tantque**
16: **Si** $\|\boldsymbol{r}\| \leqslant \text{tol}$ **alors**
17: $\boldsymbol{x}^{\text{tol}} \leftarrow \boldsymbol{x}$
18: **Fin Si**

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

Données :

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

Résultat :

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

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

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

Algorithme 3 $\mathcal{R}_0$

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

Algorithme 3 $\mathcal{R}_1$

1: $k \leftarrow 0$, $\boldsymbol{x}^{\text{tol}} \leftarrow \emptyset$
2: $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
3: $\text{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
4: **Tantque** $\|\boldsymbol{r}\| > \text{tol}$ et $k \leqslant \text{kmax}$ **faire**
5: $k \leftarrow k + 1$
6: $\boldsymbol{p} \leftarrow \boldsymbol{x}$
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: $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
11: **Fin Tantque**
12: **Si** $\|\boldsymbol{r}\| \leqslant \text{tol}$ **alors**
13: $\boldsymbol{x}^{\text{tol}} \leftarrow \boldsymbol{x}$
14: **Fin Si**

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

Algorithme 3 $\mathcal{R}_1$

1: $k \leftarrow 0$, $\boldsymbol{x}^{\text{tol}} \leftarrow \emptyset$
2: $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
3: $\text{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
4: **Tantque** $\|\boldsymbol{r}\| > \text{tol}$ et $k \leqslant \text{kmax}$ **faire**
5: $k \leftarrow k + 1$
6: $\boldsymbol{p} \leftarrow \boldsymbol{x}$
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: $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
11: **Fin Tantque**
12: **Si** $\|\boldsymbol{r}\| \leqslant \text{tol}$ **alors**
13: $\boldsymbol{x}^{\text{tol}} \leftarrow \boldsymbol{x}$
14: **Fin Si**

Algorithme 3 $\mathcal{R}_2$

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

Algorithme Méthode itérative de Gauss-Seidel pour la résolution d'un système linéaire $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$

Données :
$\mathbb{A}$: matrice de $\mathcal{M}_n(\mathbb{K})$,
$\boldsymbol{b}$: vecteur de $\mathbb{K}^n$,
$\boldsymbol{x}^0$: vecteur initial de $\mathbb{K}^n$,
ε : la tolérence, $\varepsilon \in \mathbb{R}^+$,
kmax : nombre maximum d'itérations, $\text{kmax} \in \mathbb{N}^*$
Résultat :
$\boldsymbol{X}$: un vecteur de $\mathbb{K}^n$

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

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

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

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

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

Même ossature puisque toutes deux basées sur l'Algorithme générique

Peut-on simplifier, clarifier et racourcir les codes?

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

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

$$x_i = \frac{1}{\text{A}_{ii}} \left(b_i - \sum_{j=1, j \neq i}^{n} \text{A}_{ij} y_j \right), \quad \forall i \in [\![1, n]\!].$$

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 ← 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: $S \leftarrow S + \text{A}(i,j) * \text{y}(j)$
6: **Fin Pour**
7: $\text{x}(i) \leftarrow (\text{b}(i) - S)/\text{A}(i,i)$
8: **Fin Pour**
9: **Fin Fonction**

Fonction $\boldsymbol{X} \leftarrow$ RSLJacobi2($\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \text{kmax}$)
  $k \leftarrow 0$, $\boldsymbol{X} \leftarrow \varnothing$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  $\text{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
  Tantque $\|\boldsymbol{r}\| > \text{tol}$ et $k \leqslant \text{kmax}$ **faire**
    $k \leftarrow k + 1$
    $\boldsymbol{p} \leftarrow \boldsymbol{x}$
    $\boldsymbol{x} \leftarrow$ IterJacobi$(\mathbb{A}, \boldsymbol{b}, \boldsymbol{p})$
    $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  Fin Tantque
  Si $\|\boldsymbol{r}\| \leqslant \text{tol}$ **alors**
    $\boldsymbol{X} \leftarrow \boldsymbol{x}$
  Fin Si
Fin Fonction

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

$$x_i = \frac{1}{\text{A}_{ii}} \left(b_i - \sum_{j=1, j \neq i}^{n} \text{A}_{ij} y_j \right), \quad \forall i \in [\![1, n]\!].$$

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 ← 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: $S \leftarrow S + \text{A}(i,j) * \text{y}(j)$
6: **Fin Pour**
7: $\text{x}(i) \leftarrow (\text{b}(i) - S)/\text{A}(i,i)$
8: **Fin Pour**
9: **Fin Fonction**

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

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

$$x_i = \frac{1}{\text{A}_{ii}} \left(b_i - \sum_{j=1}^{i-1} \text{A}_{i,j} x_j - \sum_{j=i+1}^{n} \text{A}_{i,j} y_j \right), \quad \forall i \in [\![1, n]\!].$$

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 ← 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: $S \leftarrow S + \text{A}(i,j) * \text{x}(j)$
6: **Fin Pour**
7: **Pour** $j \leftarrow i + 1$ **à** n **faire**
8: $S \leftarrow S + \text{A}(i,j) * \text{y}(j)$
9: **Fin Pour**
10: $\text{x}(i) \leftarrow (\text{b}(i) - S)/\text{A}(i,i)$
11: **Fin Pour**
12: **Fin Fonction**

Fonction $\boldsymbol{X} \leftarrow$ RSLGaussSeidel2($\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \text{kmax}$)
  $k \leftarrow 0$, $\boldsymbol{X} \leftarrow \varnothing$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  tol $\leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
  Tantque $\|\boldsymbol{r}\| >$ tol et $k \leqslant$ kmax **faire**
    $k \leftarrow k + 1$
    $\boldsymbol{p} \leftarrow \boldsymbol{x}$
    $\boldsymbol{x} \leftarrow$ IterGaussSeidel($\mathbb{A}, \boldsymbol{b}, \boldsymbol{p}$)
    $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  Fin Tantque
  Si $\|\boldsymbol{r}\| \leqslant$ tol **alors**
    $\boldsymbol{X} \leftarrow \boldsymbol{x}$
  Fin Si
Fin Fonction

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

$$x_i = \frac{1}{\mathrm{A}_{ii}} \left(b_i - \sum_{j=1}^{i-1} \mathrm{A}_{i,j} x_j - \sum_{j=i+1}^{n} \mathrm{A}_{i,j} y_j \right), \quad \forall i \in [\![1, n]\!].$$

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 $\mathbb{K}^n$

Fonction x $\leftarrow$ IterGaussSeidel(A, b, y)
  Pour $i \leftarrow 1$ **à** n **faire**
    $S \leftarrow 0$
    Pour $j \leftarrow 1$ **à** $i - 1$ **faire**
      $S \leftarrow S + \mathrm{A}(i, j) * \mathrm{x}(j)$
    Fin Pour
    Pour $j \leftarrow i + 1$ **à** n **faire**
      $S \leftarrow S + \mathrm{A}(i, j) * \mathrm{y}(j)$
    Fin Pour
    $\mathrm{x}(i) \leftarrow (\mathrm{b}(i) - S)/\mathrm{A}(i, i)$
  Fin Pour
Fin Fonction

Fonction $\boldsymbol{X} \leftarrow$ RSLGaussSeidel2($\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \text{kmax}$)
  $k \leftarrow 0$, $\boldsymbol{X} \leftarrow \varnothing$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  tol $\leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
  Tantque $\|\boldsymbol{r}\| >$ tol et $k \leqslant$ kmax **faire**
    $k \leftarrow k + 1$
    $\boldsymbol{p} \leftarrow \boldsymbol{x}$
    $\boldsymbol{x} \leftarrow$ IterGaussSeidel($\mathbb{A}, \boldsymbol{b}, \boldsymbol{p}$)
    $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  Fin Tantque
  Si $\|\boldsymbol{r}\| \leqslant$ tol **alors**
    $\boldsymbol{X} \leftarrow \boldsymbol{x}$
  Fin Si
Fin Fonction

Fonction $\boldsymbol{X} \leftarrow$ RSLJacobi2($\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \text{kmax}$)
  $k \leftarrow 0$, $\boldsymbol{X} \leftarrow \varnothing$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  tol $\leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
  Tantque $\|\boldsymbol{r}\| >$ tol et $k \leqslant$ kmax **faire**
    $k \leftarrow k + 1$
    $\boldsymbol{p} \leftarrow \boldsymbol{x}$
    $\boldsymbol{x} \leftarrow$ IterJacobi($\mathbb{A}, \boldsymbol{b}, \boldsymbol{p}$)
    $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  Fin Tantque
  Si $\|\boldsymbol{r}\| \leqslant$ tol **alors**
    $\boldsymbol{X} \leftarrow \boldsymbol{x}$
  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 :

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

Algorithme Méthode itérative pour la résolution d'un système linéaire $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$
Données :
$\mathbb{A}$: matrice de $\mathcal{M}_n(\mathbb{K})$,
$\boldsymbol{b}$: 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 $\mathbb{K}^n$.
$\boldsymbol{x}^0$: vecteur initial de $\mathbb{K}^n$,
ε : la tolérence, $\varepsilon \in \mathbb{R}^+$,
kmax : nombre maximum d'itérations, kmax $\in \mathbb{N}^*$
Résultat :
$\boldsymbol{x}^{\text{tol}}$: un vecteur de $\mathbb{K}^n$ si convergence, sinon $\varnothing$

Fonction $\boldsymbol{X} \leftarrow$ RSLMethIter($\mathbb{A}, \boldsymbol{b}$, IterFonc, $\boldsymbol{x}^0, \varepsilon$, kmax)
  $k \leftarrow 0$, $\boldsymbol{x}^{\text{tol}} \leftarrow \varnothing$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
  tol $\leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
  Tantque $\|\boldsymbol{r}\| >$ tol et $k \leqslant$ kmax **faire**
    $k \leftarrow k + 1$
    $\boldsymbol{p} \leftarrow \boldsymbol{x}$
    $\boldsymbol{x} \leftarrow$ IterFonc($\mathbb{A}, \boldsymbol{b}, \boldsymbol{p}$)
    $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
  Fin Tantque
  Si $\|\boldsymbol{r}\| \leqslant$ tol **alors**
    $\boldsymbol{x}^{\text{tol}} \leftarrow \boldsymbol{x}$
  Fin Si
Fin Fonction

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

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

Algorithme Méthode itérative pour la résolution d'un système linéaire $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$
Données :
$\mathbb{A}$: matrice de $\mathcal{M}_n(\mathbb{K})$,
$\boldsymbol{b}$: 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 $\mathbb{K}^n$.
$\boldsymbol{x}^0$: vecteur initial de $\mathbb{K}^n$,
ε : la tolérence, $\varepsilon \in \mathbb{R}^+$,
kmax : nombre maximum d'itérations, kmax $\in \mathbb{N}^*$
Résultat :
$\boldsymbol{x}^{\text{tol}}$: un vecteur de $\mathbb{K}^n$ si convergence, sinon $\varnothing$

Fonction $\boldsymbol{X} \leftarrow$ RSLMethIter($\mathbb{A}, \boldsymbol{b}$, IterFonc, $\boldsymbol{x}^0, \varepsilon$, kmax)
  $k \leftarrow 0$, $\boldsymbol{x}^{\text{tol}} \leftarrow \varnothing$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
  tol $\leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
  Tantque $\|\boldsymbol{r}\| >$ tol et $k \leqslant$ kmax **faire**
    $k \leftarrow k + 1$
    $\boldsymbol{p} \leftarrow \boldsymbol{x}$
    $\boldsymbol{x} \leftarrow$ IterFonc($\mathbb{A}, \boldsymbol{b}, \boldsymbol{p}$)
    $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
  Fin Tantque
  Si $\|\boldsymbol{r}\| \leqslant$ tol **alors**
    $\boldsymbol{x}^{\text{tol}} \leftarrow \boldsymbol{x}$
  Fin Si
Fin Fonction

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

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

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

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

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 ← RSLSOR3( A, b, w, x^0, ε, kmax )
  IterFun ← ((M, r, s) ↦ IterSOR(M, r, s, w))
  X ← RSLMethIter(A, b, IterFun, x^0, ε, kmax)
Fin Fonction
```

Méthodes itératives | Algorithmes scalaires | 2024/10/26 | 39 / 45

Plan

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

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

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

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

$\Rightarrow$ on peut utiliser l'algorithme vectoriel du point fixe

Algorithme Méthode de point fixe vectorielle

Données :
Φ : $\boldsymbol{\Phi} : \mathbb{K}^N \longrightarrow \mathbb{K}^N$,
$\boldsymbol{x0}$: donnée initiale, $\boldsymbol{x0} \in \mathbb{K}^N$,
tol : la tolérence, $\text{tol} \in \mathbb{R}^+$,
kmax : nombre maximum d'itérations, $\text{kmax} \in \mathbb{N}^*$
Résultat :
$\boldsymbol{\alpha}_{\text{tol}}$: un réel tel que $\|\boldsymbol{\Phi}(\boldsymbol{\alpha}_{\text{tol}}) - \boldsymbol{\alpha}_{\text{tol}}\| \leqslant \text{tol}$

1: **Fonction** $\boldsymbol{\alpha}_{\text{tol}} \leftarrow$ PtFixeVec($\boldsymbol{\Phi}, \boldsymbol{x0}, \text{tol}, \text{kmax}$)
2: $k \leftarrow 0$, $\boldsymbol{\alpha}_{\text{tol}} \leftarrow \varnothing$
3: $\boldsymbol{x} \leftarrow \boldsymbol{x0}$, $\boldsymbol{fx} \leftarrow \boldsymbol{\Phi}(\boldsymbol{x0})$,
4: $\text{err} \leftarrow \|\boldsymbol{fx} - \boldsymbol{x}\|$ ▷ ou $\frac{\|\boldsymbol{fx}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|+1}$
5: **Tantque** $\text{err} > \text{tol}$ et $k \leqslant \text{kmax}$ **faire**
6: $k \leftarrow k + 1$
7: $\boldsymbol{x} \leftarrow \boldsymbol{fx}$
8: $\boldsymbol{fx} \leftarrow \boldsymbol{\Phi}(\boldsymbol{x})$
9: $\text{err} \leftarrow \|\boldsymbol{fx} - \boldsymbol{x}\|$ ▷ ou $\frac{\|\boldsymbol{fx}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|+1}$
10: **Fin Tantque**
11: **Si** $\text{err} \leqslant \text{tol}$ **alors**
12: $\boldsymbol{\alpha}_{\text{tol}} \leftarrow \boldsymbol{x}$
13: **Fin Si**
14: **Fin Fonction**

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

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

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

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

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

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

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

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

Algorithme Méthode de point fixe vectorielle

Données :
Φ : $\boldsymbol{\Phi} : \mathbb{K}^N \longrightarrow \mathbb{K}^N$,
$\boldsymbol{x0}$: donnée initiale, $\boldsymbol{x0} \in \mathbb{K}^N$,
tol : la tolérence, $\text{tol} \in \mathbb{R}^+$,
kmax : nombre maximum d'itérations, $\text{kmax} \in \mathbb{N}^*$
Résultat :
$\boldsymbol{\alpha}_{\text{tol}}$: un réel tel que $\|\boldsymbol{\Phi}(\boldsymbol{\alpha}_{\text{tol}}) - \boldsymbol{\alpha}_{\text{tol}}\| \leqslant \text{tol}$

1: **Fonction** $\boldsymbol{\alpha}_{\text{tol}} \leftarrow$ PtFixeVec($\boldsymbol{\Phi}, \boldsymbol{x0}, \text{tol}, \text{kmax}$)
2: $k \leftarrow 0$, $\boldsymbol{\alpha}_{\text{tol}} \leftarrow \varnothing$
3: $\boldsymbol{x} \leftarrow \boldsymbol{x0}$, $\boldsymbol{fx} \leftarrow \boldsymbol{\Phi}(\boldsymbol{x0})$,
4: $\text{err} \leftarrow \|\boldsymbol{fx} - \boldsymbol{x}\|$ ▷ ou $\frac{\|\boldsymbol{fx}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|+1}$
5: **Tantque** $\text{err} > \text{tol}$ et $k \leqslant \text{kmax}$ **faire**
6: $k \leftarrow k + 1$
7: $\boldsymbol{x} \leftarrow \boldsymbol{fx}$
8: $\boldsymbol{fx} \leftarrow \boldsymbol{\Phi}(\boldsymbol{x})$
9: $\text{err} \leftarrow \|\boldsymbol{fx} - \boldsymbol{x}\|$ ▷ ou $\frac{\|\boldsymbol{fx}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|+1}$
10: **Fin Tantque**
11: **Si** $\text{err} \leqslant \text{tol}$ **alors**
12: $\boldsymbol{\alpha}_{\text{tol}} \leftarrow \boldsymbol{x}$
13: **Fin Si**
14: **Fin Fonction**

Fonction $\boldsymbol{X} \leftarrow$ RSLJacobiPF($\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \text{kmax}$)
$[\mathbb{M}, \mathbb{N}] \leftarrow$ JacobiMN($\mathbb{A}$)
$\boldsymbol{\Phi} \leftarrow \left(\boldsymbol{x} \mapsto \text{RSLMatDiag}(\mathbb{M}, \mathbb{N}\boldsymbol{x} + \boldsymbol{b})\right)$
$\boldsymbol{X} \leftarrow$ PtFixeVec($\boldsymbol{\Phi}, \boldsymbol{x0}, \text{tol}, \text{kmax}$)
Fin Fonction

Fonction $\boldsymbol{X} \leftarrow$ RSLGaussSeidelPF($\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \text{kmax}$)
$[\mathbb{M}, \mathbb{N}] \leftarrow$ GaussSeidelMN($\mathbb{A}$)
$\boldsymbol{\Phi} \leftarrow \left(\boldsymbol{x} \mapsto \text{RSLTriInf}(\mathbb{M}, \mathbb{N}\boldsymbol{x} + \boldsymbol{b})\right)$
$\boldsymbol{X} \leftarrow$ PtFixeVec($\boldsymbol{\Phi}, \boldsymbol{x0}, \text{tol}, \text{kmax}$)
Fin Fonction

Fonction $\boldsymbol{X} \leftarrow$ RSLSORPF($\mathbb{A}, \boldsymbol{b}, w, \boldsymbol{x}^0, \varepsilon, \text{kmax}$)
$[\mathbb{M}, \mathbb{N}] \leftarrow$ SORmatMN($\mathbb{A}, w$)
$\boldsymbol{\Phi} \leftarrow \left(\boldsymbol{x} \mapsto \text{RSLTriInf}(\mathbb{M}, \mathbb{N}\boldsymbol{x} + \boldsymbol{b})\right)$
$\boldsymbol{X} \leftarrow$ PtFixeVec($\boldsymbol{\Phi}, \boldsymbol{x0}, \text{tol}, \text{kmax}$)
Fin Fonction

Plan

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, ...
- ...