Analyse Numérique I

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

François Cuvelier

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

2023/10/24

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

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

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** :

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

ou encore

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

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

La méthode itérative de **Jacobi** :

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

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

ou encore

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

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

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** :

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

ou encore

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

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

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

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

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

ou encore

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

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

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)

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

Exercice 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 $\boldsymbol{b}$.

Proposition:

Soit $\mathbb{A}$ une matrice régulière 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 $\mathbb{A} = \mathbb{D} - \mathbb{E} - \mathbb{F}$. 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

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

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

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

et elle vérifie

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

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

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

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 $\mathbb{N} = \mathbb{E} + \mathbb{F}$ car

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

- Gauss-Seidel : $\mathbb{M} = \mathbb{D} - \mathbb{E}$ et $\mathbb{N} = \mathbb{F}$ car

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

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

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

Théorème:

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

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

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 $\rho(\mathbb{B}) < 1$.

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 $\rho(\mathbb{B})$.

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

Proposition:

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$ une matrice tridiagonale, i.e. $\mathrm{A}_{i,j} = 0$, si $|i - j| > 1$. Avec les notations de la Proposition 3.39, on a

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

La méthode de Gauss-Seidel converge donc plus vite que la méthode de Jacobi.

Proposition:

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

Proposition: 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, i.e. $\mathrm{A}_{i,j} = 0$, si $|i-j| > 1$. Avec les notations de la Proposition 3.39, 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. tel que

$$\rho(\mathcal{L}_{w_0}) = \min \left(\rho(\mathcal{L}_w), \; w \in]0,2[\right)$$

est donné par

$$w_0 = \frac{2}{1 + \sqrt{1 - \rho(\mathbb{J})^2}} > 1, \tag{5}$$

et, on a $\rho(\mathcal{L}_{w_0}) = w_0 - 1$.

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

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

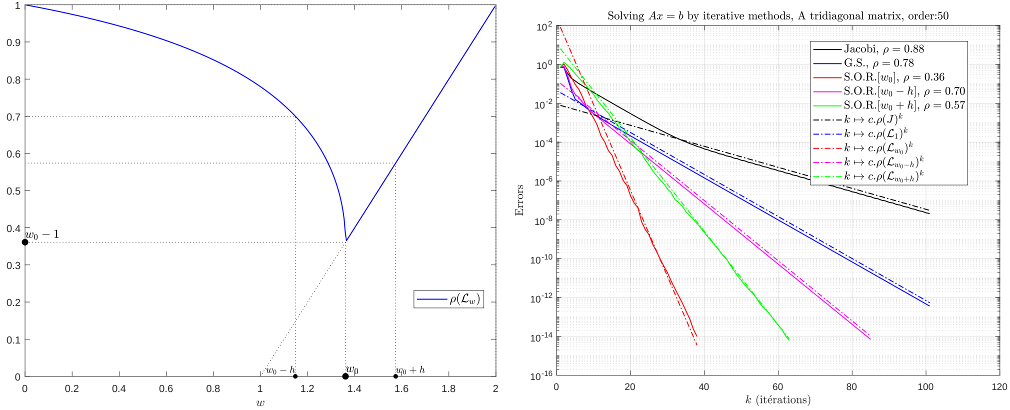

Figure: $\mathbb{A}$ est d'ordre $n = 50$.

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

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

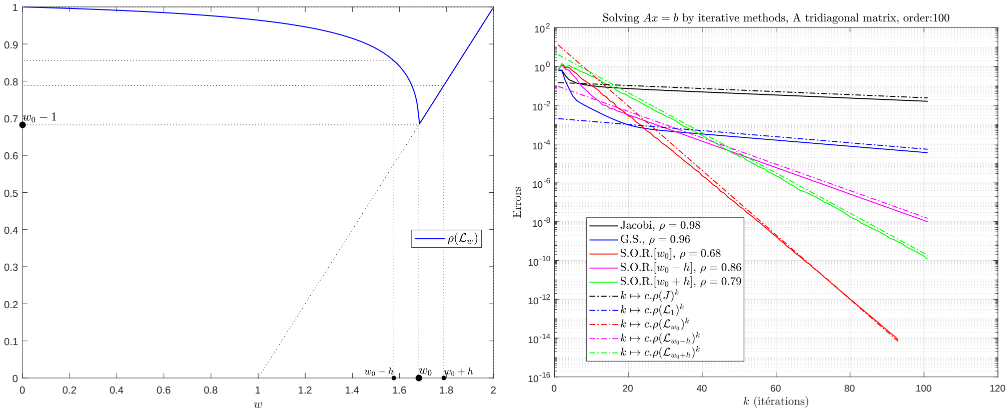

Figure: $\mathbb{A}$ est d'ordre $n = 100$.

Si $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$ matrice pleine (très peu de termes nuls).

- Une itération : environ $2n^2$ opérations élémentaires
- n itérations : environ $2n^3$ opérations élémentaires
- Méthode de Gauss: environ $2n^3/3$!

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

- Si la matrice est pleine: méthodes directes à privilégier,
- Si la matrice est creuse: méthodes itératives à privilégier.

Théorème: 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:

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

Théorème: 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.

$$\Longrightarrow \text{ boucle } \textbf{Tantque}$$

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?

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.

$$\Longrightarrow \text{ boucle } \textbf{Tantque}$$

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.

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

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.

$$\Longrightarrow \text{ boucle } \textbf{Tantque}$$

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.

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

Car dans ce cas, on a avec $\boldsymbol{e}^{[k]} = \bar{\boldsymbol{x}} - \boldsymbol{x}^{[k]}$

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

Algorithm 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, $\mathrm{kmax} \in \mathbb{N}^*$

Résultat :

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

$k \leftarrow 0$, $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \varnothing$
$\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
$\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **faire**
    $k \leftarrow k + 1$
    $\boldsymbol{p} \leftarrow \boldsymbol{x}$ ▷ $\boldsymbol{p}$ contient le vecteur précédent
    $\boldsymbol{x} \leftarrow$ calcul de l'itérée suivante en fonction de $\boldsymbol{p}$, $\mathbb{A}$, $\boldsymbol{b}$, ...
    $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
Fin Tantque
Si $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **alors** ▷ Convergence
    $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \boldsymbol{x}$
Fin Si

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

Algorithme 2 $\boxed{\mathcal{R}_0}$

$k \leftarrow 0$, $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \varnothing$
$\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
$\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **faire**
  $k \leftarrow k + 1$
  $\boldsymbol{p} \leftarrow \boldsymbol{x}$
  $\boxed{\boldsymbol{x} \leftarrow \text{calcul par Jacobi}}$
  $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
Fin Tantque
Si $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **alors** ▷ Convergence
  $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \boldsymbol{x}$
Fin Si

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

$k \leftarrow 0$, $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \varnothing$
$\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
$\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **faire**
  $k \leftarrow k + 1$
  $\boldsymbol{p} \leftarrow \boldsymbol{x}$
  Pour $i \leftarrow 1$ **à** n **faire**
    $$x_i \leftarrow \frac{1}{\mathrm{A}_{ii}}\left(b_i - \sum_{j=1,j\neq i}^{n} \mathrm{A}_{ij}\boldsymbol{p}_j\right)$$
  Fin Pour
  $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
Fin Tantque
Si $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **alors** ▷ Convergence
  $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \boldsymbol{x}$
Fin Si

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

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

$k \leftarrow 0$, $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \varnothing$
$\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
$\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **faire**
  $k \leftarrow k + 1$
  $\boldsymbol{p} \leftarrow \boldsymbol{x}$
  Pour $i \leftarrow 1$ **à** n **faire**
    $$\boxed{x_i \leftarrow \frac{1}{\mathrm{A}_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} \mathrm{A}_{ij} \boldsymbol{p}_j\right)}$$
  Fin Pour
  $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
Fin Tantque
Si $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **alors**
  $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \boldsymbol{x}$
Fin Si

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

$k \leftarrow 0$, $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \varnothing$
$\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
$\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **faire**
  $k \leftarrow k + 1$
  $\boldsymbol{p} \leftarrow \boldsymbol{x}$
  Pour $i \leftarrow 1$ **à** n **faire**
    $\mathrm{S} \leftarrow 0$
    Pour $j \leftarrow 1$ **à** n $(j \neq i)$ **faire**
      $\mathrm{S} \leftarrow \mathrm{S} + \mathrm{A}_{i,j} p_j$
    Fin Pour
    $x_i \leftarrow \frac{1}{\mathrm{A}_{ii}}(b_i - \mathrm{S})$
  Fin Pour
  $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
Fin Tantque
Si $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **alors**
  $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \boldsymbol{x}$
Fin Si

Algorithm 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, $\mathrm{kmax} \in \mathbb{N}^*$

Résultat :

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

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

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

Algorithme 3 $\mathcal{R}_0$

$k \leftarrow 0$, $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \varnothing$
$\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
$\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **faire**
    $k \leftarrow k + 1$
    $\boldsymbol{p} \leftarrow \boldsymbol{x}$
    $\boxed{\boldsymbol{x} \leftarrow \text{calcul par Gauss-Seidel}}$
    $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
Fin Tantque
Si $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **alors**
    $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \boldsymbol{x}$
Fin Si

Algorithme 3 $\mathcal{R}_1$

$k \leftarrow 0$, $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \varnothing$
$\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
$\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **faire**
    $k \leftarrow k + 1$
    $\boldsymbol{p} \leftarrow \boldsymbol{x}$
    Pour $i \leftarrow 1$ **à** n **faire**
        $$x_i \leftarrow \frac{1}{\mathrm{A}_{ii}}\left(b_i - \sum_{j=1}^{i-1} \mathrm{A}_{ij} x_j - \sum_{j=i+1}^{n} \mathrm{A}_{ij} p_j\right)$$
    Fin Pour
    $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
Fin Tantque
Si $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **alors**
    $\boldsymbol{x}^{\mathrm{tol}} \leftarrow \boldsymbol{x}$
Fin Si

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

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

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

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

$k \leftarrow 0$, $\boldsymbol{x}^{\text{tol}} \leftarrow \varnothing$
$\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
$\text{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
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**
    $x_i \leftarrow \frac{1}{\mathrm{A}_{ii}} \left(b_i - \sum_{j=1}^{i-1} \mathrm{A}_{ij} x_j - \sum_{j=i+1}^{n} \mathrm{A}_{ij} p_j \right)$
  Fin Pour
  $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A}\boldsymbol{x}$,
Fin Tantque
Si $\|\boldsymbol{r}\| \leqslant \text{tol}$ **alors**
  $\boldsymbol{x}^{\text{tol}} \leftarrow \boldsymbol{x}$
Fin Si

Algorithm 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, $\mathrm{kmax} \in \mathbb{N}^*$

Résultat :

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

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

```
Fonction X ← RSLJacobi ( 𝔸, b, x⁰, ε, kmax )
  k ← 0, X ← ∅
  x ← x⁰, r ← b − 𝔸 ∗ x,
  tol ← ε(‖b‖ + 1)
  Tantque ‖r‖ > tol et k ⩽ kmax faire
    k ← k + 1
    p ← x
    Pour i ← 1 à n faire
      S ← 0
      Pour j ← 1 à n (j ≠ i) faire
        S ← S + A(i,j) ∗ p(j)
      Fin Pour
      x(i) ← (b(i) − S)/A(i,i)
    Fin Pour
    r ← b − 𝔸 ∗ x,
  Fin Tantque
  Si ‖r‖ ⩽ tol alors
    X ← x
  Fin Si
Fin Fonction
```

```
Fonction X ← RSLGaussSeidel ( 𝔸, b, x⁰, ε, kmax )
  k ← 0, X ← ∅
  x ← x⁰, r ← b − 𝔸 ∗ x,
  tol ← ε(‖b‖ + 1)
  Tantque ‖r‖ > tol et k ⩽ kmax faire
    k ← k + 1
    p ← x
    Pour i ← 1 à n faire
      S ← 0
      Pour j ← 1 à i − 1 faire
        S ← S + A(i,j) ∗ x(j)
      Fin Pour
      Pour j ← i + 1 à n faire
        S ← S + A(i,j) ∗ p(j)
      Fin Pour
      x(i) ← (b(i) − S)/A(i,i)
    Fin Pour
    r ← b − 𝔸 ∗ x,
  Fin Tantque
  Si ‖r‖ ⩽ tol alors
    X ← x
  Fin Si
Fin Fonction
```

```
Fonction X ← RSLJacobi ( A, b, x^0, ε, kmax )
  k ← 0, X ← ∅
  x ← x^0, r ← b − A ∗ x,
  tol ← ε(‖b‖ + 1)
  Tantque ‖r‖ > tol et k ⩽ kmax faire
    k ← k + 1
    p ← x
    Pour i ← 1 à n faire
      S ← 0
      Pour j ← 1 à n (j ≠ i) faire
        S ← S + A(i,j) ∗ p(j)
      Fin Pour
      x(i) ← (b(i) − S)/A(i,i)
    Fin Pour
    r ← b − A ∗ x,
  Fin Tantque
  Si ‖r‖ ⩽ tol alors
    X ← x
  Fin Si
Fin Fonction
```

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

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

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

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 $\mathbb{K}^n$

Fonction $\mathrm{x} \leftarrow$ IterJacobi $(\mathrm{A}, \mathrm{b}, \mathrm{y})$
  Pour $i \leftarrow 1$ **à** n **faire**
    $S \leftarrow 0$
    Pour $j \leftarrow 1$ **à** n $(j \neq i)$ **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 $\boldsymbol{X} \leftarrow$ RSLJacobi2 $(\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \mathrm{kmax})$
  $k \leftarrow 0$, $\boldsymbol{X} \leftarrow \emptyset$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  $\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
  Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **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 $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **alors**
    $\boldsymbol{X} \leftarrow \boldsymbol{x}$
  Fin Si
Fin Fonction

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

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

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 $\mathbb{K}^n$

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

Algorithm 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 $\mathrm{x} \leftarrow$ IterGaussSeidel ($\mathrm{A}, \mathrm{b}, \mathrm{y}$)
  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 $(\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \mathrm{kmax})$
  $k \leftarrow 0$, $\boldsymbol{X} \leftarrow \emptyset$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  $\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
  Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **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 $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **alors**
    $\boldsymbol{X} \leftarrow \boldsymbol{x}$
  Fin Si
Fin Fonction

Algorithm 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 $\mathrm{x} \leftarrow$ IterGaussSeidel $(\mathrm{A}, \mathrm{b}, \mathrm{y})$
  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 ($\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \mathrm{kmax}$)
  $k \leftarrow 0$, $\boldsymbol{X} \leftarrow \emptyset$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  $\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
  Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **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 $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **alors**
    $\boldsymbol{X} \leftarrow \boldsymbol{x}$
  Fin Si
Fin Fonction

Fonction $\boldsymbol{X} \leftarrow$ RSLJacobi2 ($\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \mathrm{kmax}$)
  $k \leftarrow 0$, $\boldsymbol{X} \leftarrow \emptyset$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$, $\boldsymbol{r} \leftarrow \boldsymbol{b} - \mathbb{A} * \boldsymbol{x}$,
  $\mathrm{tol} \leftarrow \varepsilon(\|\boldsymbol{b}\| + 1)$
  Tantque $\|\boldsymbol{r}\| > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **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 $\|\boldsymbol{r}\| \leqslant \mathrm{tol}$ **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 :

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

Algorithm 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, $\mathrm{kmax} \in \mathbb{N}^*$

Résultat :

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

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

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

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

Algorithm 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 $\mathbb{K}^n$,
$\textsc{IterFonc}$	:	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, $\mathrm{kmax} \in \mathbb{N}^*$

Résultat :

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

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

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

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

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 $\mathbb{K}^n$

Fonction $\mathrm{x} \leftarrow$ IterSOR ($\mathrm{A}, \mathrm{b}, \mathrm{y}, w$)
    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
        $\mathrm{x}(i) \leftarrow w * (\mathrm{b}(i) - S)/\mathrm{A}(i,i) + (1-w) * \mathrm{y}(i)$
    Fin Pour
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 $w \in \mathbb{R}^*$,

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

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 $\mathbb{K}^n$

```
Fonction x ← ITERSOR ( A, b, y, w )
   Pour i ← 1 à n faire
      S ← 0
      Pour j ← 1 à i − 1 faire
         S ← S − A(i, j) ∗ x(j)
      Fin Pour
      Pour j ← i + 1 à n faire
         S ← S − A(i, j) ∗ y(j)
      Fin Pour
      x(i) ← w ∗ (b(i) − S)/A(i, i) + (1 − w) ∗ y(i)
   Fin Pour
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, x⁰, ε, kmax )
   ITERFUN ← ((𝕄, r, s) ↦ ITERSOR(𝕄, r, s, w))
   X ← RSLMETHITER(𝔸, b, ITERFUN, x⁰, ε, kmax)
Fin Fonction
```

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

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

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

Algorithm 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, $\mathrm{tol} \in \mathbb{R}^+$,

kmax : nombre maximum d'itérations, $\mathrm{kmax} \in \mathbb{N}^*$

Résultat :

$\boldsymbol{\alpha}_{\mathrm{tol}}$: un réel tel que $\|\boldsymbol{\Phi}(\boldsymbol{\alpha}_{\mathrm{tol}}) - \boldsymbol{\alpha}_{\mathrm{tol}}\| \leqslant \mathrm{tol}$

```
Fonction α_tol ← PtFixeVec ( Φ, x0, tol, kmax )
  k ← 0, α_tol ← ∅
  x ← x0, fx ← Φ(x0),
  err ← ‖fx − x‖                              ▷ ou ‖fx−x‖/(‖x‖+1)
  Tantque err > tol et k ⩽ kmax faire
    k ← k + 1
    x ← fx
    fx ← Φ(x)
    err ← ‖fx − x‖                            ▷ ou ‖fx−x‖/(‖x‖+1)
  Fin Tantque
  Si err ⩽ tol alors
    α_tol ← x
  Fin Si
Fin Fonction
```

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

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

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

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

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

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

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

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

Algorithm 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, $\mathrm{tol} \in \mathbb{R}^+$,
kmax	:	nombre maximum d'itérations, $\mathrm{kmax} \in \mathbb{N}^*$

Résultat :

$\boldsymbol{\alpha}_{\mathrm{tol}}$	:	un réel tel que $\lVert\boldsymbol{\Phi}(\boldsymbol{\alpha}_{\mathrm{tol}}) - \boldsymbol{\alpha}_{\mathrm{tol}}\rVert \leqslant \mathrm{tol}$

Fonction $\boldsymbol{\alpha}_{\mathrm{tol}} \leftarrow$ PtFixeVec ($\boldsymbol{\Phi}, \boldsymbol{x0}, \mathrm{tol}, \mathrm{kmax}$)
  $k \leftarrow 0$, $\boldsymbol{\alpha}_{\mathrm{tol}} \leftarrow \emptyset$
  $\boldsymbol{x} \leftarrow \boldsymbol{x0}$, $\boldsymbol{fx} \leftarrow \boldsymbol{\Phi}(\boldsymbol{x0})$,
  $\mathrm{err} \leftarrow \lVert\boldsymbol{fx} - \boldsymbol{x}\rVert$ ▷ ou $\frac{\lVert\boldsymbol{fx}-\boldsymbol{x}\rVert}{\lVert\boldsymbol{x}\rVert+1}$
  Tantque $\mathrm{err} > \mathrm{tol}$ et $k \leqslant \mathrm{kmax}$ **faire**
    $k \leftarrow k + 1$
    $\boldsymbol{x} \leftarrow \boldsymbol{fx}$
    $\boldsymbol{fx} \leftarrow \boldsymbol{\Phi}(\boldsymbol{x})$
    $\mathrm{err} \leftarrow \lVert\boldsymbol{fx} - \boldsymbol{x}\rVert$ ▷ ou $\frac{\lVert\boldsymbol{fx}-\boldsymbol{x}\rVert}{\lVert\boldsymbol{x}\rVert+1}$
  Fin Tantque
  Si $\mathrm{err} \leqslant \mathrm{tol}$ **alors**
    $\boldsymbol{\alpha}_{\mathrm{tol}} \leftarrow \boldsymbol{x}$
  Fin Si
Fin Fonction

Fonction $\boldsymbol{X} \leftarrow$ RSLJacobiPF ($\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \mathrm{kmax}$)
  $[\mathbb{M}, \mathbb{N}] \leftarrow$ 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 $(\boldsymbol{\Phi}, \boldsymbol{x0}, \mathrm{tol}, \mathrm{kmax})$
Fin Fonction

Fonction $\boldsymbol{X} \leftarrow$ RSLGaussSeidelPF ($\mathbb{A}, \boldsymbol{b}, \boldsymbol{x}^0, \varepsilon, \mathrm{kmax}$)
  $[\mathbb{M}, \mathbb{N}] \leftarrow$ 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 $(\boldsymbol{\Phi}, \boldsymbol{x0}, \mathrm{tol}, \mathrm{kmax})$
Fin Fonction

Fonction $\boldsymbol{X} \leftarrow$ RSLSORPF ($\mathbb{A}, \boldsymbol{b}, w, \boldsymbol{x}^0, \varepsilon, \mathrm{kmax}$)
  $[\mathbb{M}, \mathbb{N}] \leftarrow$ 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 $(\boldsymbol{\Phi}, \boldsymbol{x0}, \mathrm{tol}, \mathrm{kmax})$
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, ...
- ...