

# Analyse Numérique I

Sup'Galilée, Ingénieurs MACS, 1ère année

François Cuvelier

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

2015/10/28

# Chapitre IV

## Résolution de systèmes linéaires

# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

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

Soient  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  une matrice inversible et  $\mathbf{b} \in \mathbb{K}^n$ .

Résoudre

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

Le calcul de la matrice inverse  $\mathbb{A}^{-1}$  revient à résoudre  $n$  systèmes linéaires.



Pour résoudre un système linéaire, on ne calcule pas la matrice inverse associée.

- **Méthodes directes** : On cherche  $\mathbb{M}$  inversible tel que  $\mathbb{M}\mathbb{A}$  *facilement* inversible

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

- **Méthodes itératives** : On cherche  $\mathbb{B}$  et  $\mathbf{c}$ ,

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

en espérant  $\lim_{k \rightarrow +\infty} \mathbf{x}^{[k]} = \mathbf{x}$ .

# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

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

## Definition

Une **norme** sur un espace vectoriel  $V$  est une application  $\|\bullet\| : V \rightarrow \mathbb{R}^+$  qui vérifie les propriétés suivantes

- ◇  $\|\mathbf{v}\| = 0 \iff \mathbf{v} = 0$ ,
- ◇  $\|\alpha\mathbf{v}\| = |\alpha| \|\mathbf{v}\|$ ,  $\forall \alpha \in \mathbb{K}$ ,  $\forall \mathbf{v} \in V$ ,
- ◇  $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ ,  $\forall (\mathbf{u}, \mathbf{v}) \in V^2$  (inégalité triangulaire).

Une norme sur  $V$  est également appelée **norme vectorielle**. On appelle **espace vectoriel normé** un espace vectoriel muni d'une norme.



## Proposition

Soit  $\mathbf{v} \in \mathbb{K}^n$ . Pour tout nombre réel  $p \geq 1$ , l'application  $\|\bullet\|_p$  définie par

$$\|\mathbf{v}\|_p = \left( \sum_{i=1}^n |v_i|^p \right)^{1/p}$$

est une norme sur  $\mathbb{K}^n$ .

Normes usitées :

$$\|\mathbf{v}\|_1 = \sum_{i=1}^n |v_i|, \quad \|\mathbf{v}\|_2 = \left( \sum_{i=1}^n |v_i|^2 \right)^{1/2}, \quad \|\mathbf{v}\|_\infty = \max_{i \in \llbracket 1, n \rrbracket} |v_i|.$$



## Lemme 2.1: Inégalité de Cauchy-Schwarz



$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{K}^n$$

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2. \quad (1)$$

Cette inégalité s'appelle l'**inégalité de Cauchy-Schwarz**. On a égalité si et seulement si  $\mathbf{x}$  et  $\mathbf{y}$  sont colinéaires.





## Lemme 2.2: Inégalité de Hölder



Pour  $p > 1$  et  $\frac{1}{p} + \frac{1}{q} = 1$ , on a  $\forall \mathbf{u}, \mathbf{v} \in \mathbb{K}^n$

$$\sum_{i=1}^n |u_i v_i| \leq \left( \sum_{i=1}^n |u_i|^p \right)^{1/p} \left( \sum_{i=1}^n |v_i|^q \right)^{1/q} = \|\mathbf{u}\|_p \|\mathbf{v}\|_q. \quad (2)$$

Cette inégalité s'appelle l'**inégalité de Hölder**.

## ♥ Definition 2.3

Deux **normes**  $\|\bullet\|$  et  $\|\bullet\|'$ , définies sur un même espace vectoriel  $V$ , sont **équivalentes** s'il existe deux constantes  $C$  et  $C'$  telles que

$$\|\mathbf{v}\|' \leq C \|\mathbf{v}\| \quad \text{et} \quad \|\mathbf{v}\| \leq C' \|\mathbf{v}\|' \quad \text{pour tout } \mathbf{v} \in V. \quad (3)$$

## 📖 Proposition

Sur un espace vectoriel de dimension finie toutes les normes sont équivalentes.

# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

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

## ♥ Definition 2.4

Une **norme matricielle** sur  $\mathcal{M}_n(\mathbb{K})$  est une application  $\|\bullet\| : \mathcal{M}_n(\mathbb{K}) \rightarrow \mathbb{R}^+$  vérifiant

- ①  $\|A\| = 0 \iff A = 0,$
- ②  $\|\alpha A\| = |\alpha| \|A\|, \forall \alpha \in \mathbb{K}, \forall A \in \mathcal{M}_n(\mathbb{K}),$
- ③  $\|A + B\| \leq \|A\| + \|B\|, \forall (A, B) \in \mathcal{M}_n(\mathbb{K})^2$  (inégalité triangulaire)
- ④  $\|AB\| \leq \|A\| \|B\|, \forall (A, B) \in \mathcal{M}_n(\mathbb{K})^2$

Peut-on étendre cette définition sur  $\mathcal{M}_{m,n}(\mathbb{K})$ ?

### Proposition:

Etant donné une norme vectorielle  $\|\bullet\|$  sur  $\mathbb{C}^n$ , l'application  $\|\bullet\|_s : \mathcal{M}_n(\mathbb{C}) \rightarrow \mathbb{R}^+$  définie par

$$\|A\|_s = \sup_{\substack{\mathbf{v} \in \mathbb{C}^n \\ \mathbf{v} \neq 0}} \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{C}^n \\ \|\mathbf{v}\| \leq 1}} \|A\mathbf{v}\| = \sup_{\substack{\mathbf{v} \in \mathbb{C}^n \\ \|\mathbf{v}\| = 1}} \|A\mathbf{v}\|, \quad (4)$$

est une norme matricielle, appelée **norme matricielle subordonnée** (à la norme vectorielle donnée).

De plus

$$\|A\mathbf{v}\| \leq \|A\|_s \|\mathbf{v}\| \quad \forall \mathbf{v} \in \mathbb{C}^n \quad (5)$$

et la norme  $\|A\|$  peut se définir aussi par

$$\|A\|_s = \inf \{ \alpha \in \mathbb{R} : \|A\mathbf{v}\| \leq \alpha \|\mathbf{v}\|, \quad \forall \mathbf{v} \in \mathbb{C}^n \}. \quad (6)$$

Il existe au moins un vecteur  $\mathbf{u} \in \mathbb{C}^n$  tel que

$$\mathbf{u} \neq 0 \quad \text{et} \quad \|A\mathbf{u}\| = \|A\|_s \|\mathbf{u}\|. \quad (7)$$

Enfin une norme subordonnée vérifie toujours

$$\|I\|_s = 1 \quad (8)$$



## Théorème 3

Soit  $A \in \mathcal{M}_n(\mathbb{K})$ . On a

$$\|A\|_1 \stackrel{\text{def}}{=} \sup_{\substack{v \in \mathbb{K}^n \\ v \neq 0}} \frac{\|Av\|_1}{\|v\|_1} = \max_{j \in \llbracket 1, n \rrbracket} \sum_{i=1}^n |a_{ij}| \quad (9)$$

$$\|A\|_2 \stackrel{\text{def}}{=} \sup_{\substack{v \in \mathbb{K}^n \\ v \neq 0}} \frac{\|Av\|_2}{\|v\|_2} = \sqrt{\rho(A^*A)} = \sqrt{\rho(AA^*)} = \|A^*\|_2 \quad (10)$$

$$\|A\|_\infty \stackrel{\text{def}}{=} \sup_{\substack{v \in \mathbb{K}^n \\ v \neq 0}} \frac{\|Av\|_\infty}{\|v\|_\infty} = \max_{i \in \llbracket 1, n \rrbracket} \sum_{j=1}^n |a_{ij}| \quad (11)$$

La norme  $\|\bullet\|_2$  est invariante par transformation unitaire :

$$UU^* = I \implies \|A\|_2 = \|AU\|_2 = \|UA\|_2 = \|U^*AU\|_2. \quad (12)$$



### Corollaire 3.1

- ① Si une matrice  $\mathbb{A}$  est hermitienne, ou symétrique (donc normale), on a  $\|\mathbb{A}\|_2 = \rho(\mathbb{A})$ .
- ② Si une matrice  $\mathbb{A}$  est unitaire, ou orthogonale (donc normale), on a  $\|\mathbb{A}\|_2 = 1$ .

# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

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



## ♥ Definition 3.2

Soit  $V$  un espace vectoriel muni d'une norme  $\|\bullet\|$ , on dit qu'une suite  $(\mathbf{v}_k)$  d'éléments de  $V$  **converge vers un élément**  $\mathbf{v} \in V$ , si

$$\lim_{k \rightarrow \infty} \|\mathbf{v}_k - \mathbf{v}\| = 0$$

et on écrit

$$\mathbf{v} = \lim_{k \rightarrow \infty} \mathbf{v}_k.$$



## Théorème 4: admis

Soit  $\mathbb{B}$  une matrice carrée. Les conditions suivantes sont équivalentes :

- ①  $\lim_{k \rightarrow \infty} \mathbb{B}^k = 0$ ,
- ②  $\lim_{k \rightarrow \infty} \mathbb{B}^k \mathbf{v} = 0$  pour tout vecteur  $\mathbf{v}$ ,
- ③  $\rho(\mathbb{B}) < 1$ ,
- ④  $\|\mathbb{B}\| < 1$  pour au moins une norme matricielle subordonnée  $\|\bullet\|$ .



## Théorème 5: admis

Soit  $\mathbb{B}$  une matrice carrée, et  $\|\bullet\|$  une norme matricielle quelconque. Alors

$$\lim_{k \rightarrow \infty} \left\| \mathbb{B}^k \right\|^{1/k} = \rho(\mathbb{B}).$$

Soient  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  inversible et  $\mathbf{b} \in \mathbb{K}^n$ .

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

De petites erreurs sur les données engendrent-elles de petites erreurs sur la solution?

## Exemple de R.S. Wilson

Soient

$$\mathbb{A} = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, \quad \Delta\mathbb{A} = \begin{pmatrix} 0 & 0 & \frac{1}{10} & \frac{1}{5} \\ \frac{2}{25} & \frac{1}{25} & 0 & 0 \\ 0 & -\frac{1}{50} & -\frac{11}{100} & 0 \\ -\frac{1}{100} & -\frac{1}{100} & 0 & -\frac{1}{50} \end{pmatrix}$$

et  $\mathbf{b}^t = (32, 23, 33, 31)$ ,  $(\Delta\mathbf{b})^t = (\frac{1}{100}, -\frac{1}{100}, \frac{1}{100}, -\frac{1}{100})$ . Des calculs exacts donnent

$$\mathbb{A}\mathbf{x} = \mathbf{b} \quad \Longleftrightarrow \quad \mathbf{x}^t = (1, 1, 1, 1)$$

$$\mathbb{A}\mathbf{u} = (\mathbf{b} + \Delta\mathbf{b}) \quad \Longleftrightarrow \quad \mathbf{u}^t = \left( \frac{91}{50}, -\frac{9}{25}, \frac{27}{20}, \frac{79}{100} \right) \\ \approx (1.8, -0.36, 1.3, 0.79)$$

$$(\mathbb{A} + \Delta\mathbb{A})\mathbf{v} = \mathbf{b} \quad \Longleftrightarrow \quad \mathbf{v}^t = (-81, 137, -34, 22)$$

$$(\mathbb{A} + \Delta\mathbb{A})\mathbf{y} = (\mathbf{b} + \Delta\mathbf{b}) \quad \Longleftrightarrow \quad \mathbf{y}^t = \left( -\frac{18283543}{461600}, \frac{31504261}{461600}, -\frac{3741501}{230800}, \frac{5235241}{461600} \right) \\ \approx (-39.61, 68.25, -16.21, 11.34)$$

Soient  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  inversible et  $\mathbf{b} \in \mathbb{K}^n$ .

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

De petites erreurs sur les données engendrent-elles de petites erreurs sur la solution?

Non, pas forcément!

Le système linéaire prédédent est **mal conditionné**.

On dit qu'un système linéaire est **bien conditionné** ou qu'il a un **bon conditionnement** si de petites perturbations des données n'entraînent qu'une variation *raisonnable* de la solution.

Est-il possible de "mesurer" le **conditionnement** d'une matrice?

## ♥ Definition 6.1

Soit  $\|\cdot\|$  une norme matricielle subordonnée, le conditionnement d'une matrice régulière  $\mathbb{A}$ , associé à cette norme, est le nombre

$$\text{cond}(\mathbb{A}) = \|\mathbb{A}\| \|\mathbb{A}^{-1}\|.$$

Nous noterons  $\text{cond}_p(\mathbb{A}) = \|\mathbb{A}\|_p \|\mathbb{A}^{-1}\|_p$ .



## Proposition:



Soit  $A$  une matrice régulière. On a les propriétés suivantes

- ①  $\forall \alpha \in \mathbb{K}^*, \text{cond}(\alpha A) = \text{cond}(A)$ .
- ②  $\text{cond}_p(A) \geq 1, \forall p \in \llbracket 1, +\infty \rrbracket$ .
- ③  $\text{cond}_2(A) = 1$  si et seulement si  $A = \alpha Q$  avec  $\alpha \in \mathbb{K}^*$  et  $Q$  matrice unitaire



## Théorème 7:



Soit  $\mathbb{A}$  une matrice inversible. Soient  $\mathbf{x}$  et  $\mathbf{x} + \Delta\mathbf{x}$  les solutions respectives de

$$\mathbb{A}\mathbf{x} = \mathbf{b} \quad \text{et} \quad \mathbb{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}.$$

Supposons  $\mathbf{b} \neq \mathbf{0}$ , alors l'inégalité

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \text{cond}(\mathbb{A}) \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}$$

est satisfaite, et c'est la meilleure possible : pour une matrice  $\mathbb{A}$  donnée, on peut trouver des vecteurs  $\mathbf{b} \neq \mathbf{0}$  et  $\Delta\mathbf{b} \neq \mathbf{0}$  tels qu'elle devienne une égalité.





## Théorème:



Soient  $\mathbb{A}$  et  $\mathbb{A} + \Delta\mathbb{A}$  deux matrices inversibles. Soient  $\mathbf{x}$  et  $\mathbf{x} + \Delta\mathbf{x}$  les solutions respectives de

$$\mathbb{A}\mathbf{x} = \mathbf{b} \text{ et } (\mathbb{A} + \Delta\mathbb{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b}.$$

Supposons  $\mathbf{b} \neq \mathbf{0}$ , alors on a

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x} + \Delta\mathbf{x}\|} \leq \text{cond}(\mathbb{A}) \frac{\|\Delta\mathbb{A}\|}{\|\mathbb{A}\|}.$$

### Remarque 7.1

Une matrice est donc **bien conditionnée** si son conditionnement est proche de 1.

# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

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

## Méthodes itératives pour la résolution du système linéaire

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

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

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

vérifie

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

# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

## • Notations

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

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

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

# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

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

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

La méthode itérative de **Jacobi** :

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

ou encore

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

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

La méthode itérative de **Jacobi** :

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

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

ou encore

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

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



# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

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

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

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

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

ou encore

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

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

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

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

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

ou encore

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

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

# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

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

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

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

où  $\hat{x}_i^{[k+1]}$  est obtenu à partir de l'une des deux méthodes précédentes.  
Avec la méthode de Gauss-Seidel : méthode S.O.R. (successive over relaxation)

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



### Exercice 8.1:



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

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

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



## Proposition:



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

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

$$\mathbb{J} = \mathbb{D}^{-1}(\mathbb{E} + \mathbb{F}) = \mathbb{L} + \mathbb{U}, \quad (13)$$

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

$$\mathcal{L}_w = \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \left( \frac{1-w}{w} \mathbb{D} + \mathbb{F} \right) = (\mathbb{I} - w\mathbb{L})^{-1} ((1-w)\mathbb{I} + w\mathbb{U}). \quad (14)$$

et elle vérifie

$$\rho(\mathcal{L}_w) \geq |w - 1|. \quad (15)$$

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

$$\mathcal{L}_1 = (\mathbb{D} - \mathbb{E})^{-1}\mathbb{F} = \mathbb{I} - \mathbb{L}^{-1}\mathbb{U}. \quad (16)$$

# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

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



## Théorème:



En décomposant  $\mathbb{A}$  sous la forme  $\mathbb{A} = \mathbb{M} - \mathbb{N}$  avec  $\mathbb{M}$  régulière et en posant

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

la suite définie par

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

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





## Corollaire:



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



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

Soit  $A$  une matrice à diagonale strictement dominante ou une matrice inversible à diagonale fortement dominante alors

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



## Théorème

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



## Théorème

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

# Plan

- 1 Méthodes directes
- 2 Normes vectorielles et normes matricielles
  - Normes vectorielles
  - Normes matricielles
  - Suites de vecteurs et de matrices
- 3 Conditionnement d'un système linéaire
- 4 Méthodes itératives
  - Principe

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

# Principe de base

Résoudre :

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

Méthodes itératives :

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

Algorithme :

$\mathbf{x}^{[0]}$  donné

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

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

**Fin Pour**

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

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

⇒ boucle **Tantque**

**Critères d'arrêt :**

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

Comment choisir le critère d'arrêt pour la convergence?

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

⇒ boucle **Tantque**

**Critères d'arrêt :**

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

Comment choisir le critère d'arrêt pour la convergence?

Exemple de critère d'arrêt pour la convergence :

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

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

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

⇒ boucle **Tantque**

**Critères d'arrêt :**

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

Comment choisir le critère d'arrêt pour la convergence?

Exemple de critère d'arrêt pour la convergence :

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

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

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

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

---

**Algorithme 1** Méthode itérative pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$ 

---

**Données :**

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

**Résultat :**

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

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



Pour Jacobi , la suite des itérées est définie par

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

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

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

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

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

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

```

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

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

```

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

```

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

```

---

**Algorithme 2** Méthode itérative de Jacobi pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$ 

---

**Données :**

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

**Résultat :**

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

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

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

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

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

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

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

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

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

```

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

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

```

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

```

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

```

---

**Algorithme 3** Méthode itérative de Gauss-Seidel pour la résolution d'un système linéaire  $A\mathbf{x} = \mathbf{b}$ 

---

**Données :**

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

**Résultat :**

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

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

---

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

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

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

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

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

$k \leftarrow k + 1$

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

Pour  $i \leftarrow 1$  à  $n$  faire

$s \leftarrow 0$

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

$s \leftarrow s - A(i, j) * p(j)$

Fin Pour

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

Fin Pour

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

Fin Tantque

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

$X \leftarrow \mathbf{x}$

Fin Si

Fin Fonction

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

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

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

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

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

$k \leftarrow k + 1$

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

Pour  $i \leftarrow 1$  à  $n$  faire

$s \leftarrow 0$

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

$s \leftarrow s - A(i, j) * x(j)$

Fin Pour

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

$s \leftarrow s - A(i, j) * p(j)$

Fin Pour

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

Fin Pour

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

Fin Tantque

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

$X \leftarrow \mathbf{x}$

Fin Si

Fin Fonction

```

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

```

```

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

```

Même ossature puisque toutes deux basées sur l'Algorithme générique  
 Peut-on simplifier, clarifier et raccourcir les codes?



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

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

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

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

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

$k \leftarrow k + 1$

$p \leftarrow x$

Pour  $i \leftarrow 1$  à  $n$  faire

$s \leftarrow 0$

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

$s \leftarrow s - A(i, j) * p(j)$

Fin Pour

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

Fin Pour

$r \leftarrow b - A * x,$

Fin Tantque

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

$X \leftarrow x$

Fin Si

Fin Fonction

---

**Algorithme 4** itération de Jacobi : calcul de  $x$  tel que

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

---

**Données :**

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

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

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

**Résultat :**

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

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

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

3:  $S \leftarrow 0$

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

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

6: Fin Pour

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

8: Fin Pour

9: Fin Fonction

---

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

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

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

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

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

$k \leftarrow k + 1$

$p \leftarrow x$

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

$r \leftarrow b - A * x,$

Fin Tantque

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

$X \leftarrow x$

Fin Si

Fin Fonction

---

**Algorithme 5** itération de Jacobi : calcul de  $x$  tel que

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

---

**Données :**

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

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

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

**Résultat :**

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

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

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

3:  $S \leftarrow 0$

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

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

6: Fin Pour

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

8: Fin Pour

9: Fin Fonction

---

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

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

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

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

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

$k \leftarrow k + 1$

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

Pour  $i \leftarrow 1$  à  $n$  faire

$S \leftarrow 0$

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

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

Fin Pour

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

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

Fin Pour

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

Fin Pour

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

Fin Tantque

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

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

Fin Si

Fin Fonction

**Algorithme 6** Itération de Gauss-Seidel : calcul de  $\mathbf{x}$  tel que

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

**Données :**

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

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

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

**Résultat :**

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

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

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

3:  $S \leftarrow 0$

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

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

6: Fin Pour

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

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

9: Fin Pour

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

11: Fin Pour

12: Fin Fonction

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

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

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

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

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

$k \leftarrow k + 1$

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

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

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

Fin Tantque

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

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

Fin Si

Fin Fonction

---

**Algorithme 7** Itération de Gauss-Seidel : calcul de  $\mathbf{x}$  tel que

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

---

**Données :**

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

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

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

**Résultat :**

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

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

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

3:  $S \leftarrow 0$

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

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

6: Fin Pour

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

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

9: Fin Pour

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

11: Fin Pour

12: Fin Fonction

---

```

Fonction  $\mathbf{X} \leftarrow \text{RSLGAUSSSEIDEL2}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
   $k \leftarrow 0, \mathbf{X} \leftarrow \emptyset$ 
   $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
   $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
  Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
     $k \leftarrow k + 1$ 
     $\mathbf{p} \leftarrow \mathbf{x}$ 
     $\mathbf{x} \leftarrow \text{ITERGAUSSSEIDEL}(\mathbb{A}, \mathbf{b}, \mathbf{p})$ 
     $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
  Fin Tantque
  Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
     $\mathbf{X} \leftarrow \mathbf{x}$ 
  Fin Si
Fin Fonction

```

```

Fonction  $\mathbf{X} \leftarrow \text{RSLJACOBI2}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
   $k \leftarrow 0, \mathbf{X} \leftarrow \emptyset$ 
   $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
   $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
  Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
     $k \leftarrow k + 1$ 
     $\mathbf{p} \leftarrow \mathbf{x}$ 
     $\mathbf{x} \leftarrow \text{ITERJACOBI}(\mathbb{A}, \mathbf{b}, \mathbf{p})$ 
     $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x},$ 
  Fin Tantque
  Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
     $\mathbf{X} \leftarrow \mathbf{x}$ 
  Fin Si
Fin Fonction

```

Les deux codes sont fortement similaires!

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

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

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

---

**Algorithme 7** Méthode itérative pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$

---

Données :

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

Résultat :

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

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

Fonction  $X \leftarrow \text{RSLJACOBI3}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$   
 $X \leftarrow \text{RSLMETHITER}(\mathbb{A}, \mathbf{b}, \text{ITERJACOBI}, \mathbf{x}^0, \varepsilon, \text{kmax})$   
 Fin Fonction

Fonction  $X \leftarrow \text{RSLGAUSSSEIDEL3}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$   
 $X \leftarrow \text{RSLMETHITER}(\mathbb{A}, \mathbf{b}, \text{ITERGAUSSSEIDEL}, \mathbf{x}^0, \varepsilon, \text{kmax})$   
 Fin Fonction

---

**Algorithme 7** Méthode itérative pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$

---

Données :

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

Résultat :

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

```

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

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

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

---

**Algorithme 8** 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  $\mathbb{K}^n$

Paramètre  $w$  "en trop" dans l'appel de la fonction **ITER-SOR** pour pouvoir utiliser la fonction générique **RSLMETHITER** !

```
1: Fonction x ← ITER-SOR ( 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) * x(i)
11:  Fin Pour
12: Fin Fonction
```



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

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

---

**Algorithme 9** 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) * x(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, x0, ε, kmax )
  ITERFUN ← ((M, r, s) ↦ ITERSOR(M, r, s, w))
  X ← RSLMETHITER(A, b, ITERFUN, x0, ε, kmax)
Fin Fonction
```