

# Chapitre 5

## Systemes linéaires

## Structure du chapitre

- Méthodes directes  
Elimination de Gauss, résolution de systèmes triangulaires
- Etudes des erreurs de calcul  
Conditionnement d'un système, erreurs d'arrondis
- Méthodes itératives  
Similaire à la résolution de systèmes non linéaires, efficace pour des matrices creuses
- Calcul des valeurs propres d'une matrice
- Optimisation linéaire - algorithme du simplexe

- 75% des problèmes scientifiques font intervenir des opérateurs linéaires : problèmes **très courants**
- Grande différence entre **théorie** et **pratique**
  - Théorie : connue depuis des siècles mais pas toujours adaptée.
    - Exemple : règles de Cramer : complexité  $n!$
  - Pratique : connue plus récemment
    - Bon comportement de l'élimination de Gauss, décomposition LU.
- Obtenir un bon logiciel linéaire pour la grande échelle : précision d'horloger
  - Prix de logiciels (académique) :
    - Matlab (généraliste) : 1000 €
    - Cplex (optimisation linéaire) : 1000 €

## Système triangulaire

Particulièrement simple à résoudre :

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 3 & 2 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

$$x_4 = -4$$

$$x_3 = 3 - 2x_4 \Rightarrow x_3 = 11$$

$$3x_2 = 2 - 2x_3 - x_4 \Rightarrow x_2 = -\frac{16}{3}$$

$$x_1 = 1 - 2x_2 - 3x_3 - 4x_4 \Rightarrow x_1 = -\frac{16}{3}$$

Principe de la **substitution arrière**

Calcul du nombre d'opérations à effectuer pour résoudre un système triangulaire :

$$\begin{aligned}u_{11} x_1 + u_{12} x_2 + \cdots + u_{1n} x_n &= b_1 \\u_{22} x_2 + \cdots + u_{2n} x_n &= b_2 \\&\vdots \\u_{nn} x_n &= b_n\end{aligned}$$

$$\begin{aligned}x_n &= \frac{b_n}{u_{nn}} \\x_{n-1} &= \frac{b_{n-1} - u_{n-1,n} x_n}{u_{n-1,n-1}} \\&\vdots \\x_1 &= \frac{b_1 - u_{1n} x_n - u_{1,n-1} x_{n-1} - \cdots - u_{12} x_2}{u_{11}}\end{aligned}$$

$$x_i = \frac{b_i - \sum_{k=i+1}^n u_{ik} x_k}{u_{ii}} \quad i = n, n-1, \dots, 1$$

$(n-i)$  additions,  $(n-i)$  multiplications, 1 soustraction, 1 division

Nombre total d'opérations :

$$\begin{aligned} \sum_{i=1}^n (2n - 2i + 2) &= 2n^2 + 2n - 2 \sum_{i=1}^n i \\ &= 2n^2 + 2n - n(n+1) = n^2 + n \end{aligned}$$

$\mathcal{O}(n^2)$  opérations

## Elimination de Gauss

Principe : Transformer les lignes de la matrice afin d'obtenir un système triangulaire que l'on résout ensuite par la substitution arrière.

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 2 \\ 1 & -1 & 2 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -2 & -2 & -4 \\ 0 & 1 & 2 & 2 \\ 0 & -3 & -1 & -4 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -2 & -2 & -4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -2 & -2 & -4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

1e étape : On crée des 0 en dessous de l'élément  $A(1, 1)$

$$L_2 \leftarrow L_2 - L_1$$

$$L_4 \leftarrow L_4 - L_1$$

2e étape : On crée des 0 en dessous de l'élément  $A(2, 2)$

$$L_3 \leftarrow L_3 + \frac{1}{2}L_2$$

$$L_4 \leftarrow L_4 - \frac{3}{2}L_2$$

3e étape : On crée un 0 en dessous de l'élément  $A(3, 3)$

$$L_4 \leftarrow L_4 - 2L_3$$

On obtient un système triangulaire que l'on peut résoudre par substitution arrière

Solution :

## Théorème

Considérons le système  $Ax = b$ . Si on effectue l'élimination de Gauss pour obtenir un système triangulaire, le nombre d'opérations nécessaires est

$$\frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n.$$

*Preuve* : A l'étape  $i$ , on effectue, **pour chaque ligne à éliminer**

1 division

$(n - i + 1)$  multiplications

$(n - i + 1)$  soustractions

Il reste, à ce stade,  $(n - i)$  lignes à éliminer

# Elimination de Gauss

- Le **pivot** est l'élément en dessous duquel on crée des zéros !
- **Problème** : si le pivot = 0

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

- Nécessité d'**échanger des lignes**  
Parmi tous les éléments en dessous du pivot, **au moins un** doit être non nul, sinon la matrice est **singulière**.

## Elimination de Gauss : choix des pivots

- Risque d'**instabilité numérique** si un des pivots **est trop proche de 0**.
- Exemple :

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

- Sans permutation des lignes, on obtient :

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0.0001 & 1 \\ 0 & 0 & 9999 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 10000 \end{pmatrix}$$

et une solution (avec quatre chiffres de précision)

$$(x_1, x_2, x_3) \approx (0, 0, 1)$$

au lieu de la vraie solution

$$(x_1, x_2, x_3) \approx (1, -1.0001, 1.0001).$$

- **Pivotage partiel** : choisir l'élément le plus grand de la colonne en cours  
**Pivotage complet** : choisir l'élément le plus grand de la sous-matrice restante

# Décomposition LU

## Principe

On **factorise** la matrice  $A$  en

$$A = LU$$

où  $L$  est **triangulaire inférieure** et  $U$  est **triangulaire supérieure**. On résout les deux systèmes **triangulaires**

$$Ly = b \quad \text{et} \quad Ux = y.$$

- **Complexité** :  $2n^2$
- **Elimination de Gauss** :  $\frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n$

## Décomposition LU et élimination de Gauss

### Théorème

Soit  $A$  une matrice donnée d'ordre  $n$  pour laquelle l'élimination de Gauss peut être effectuée sans permutation de lignes. Alors cette matrice a une décomposition LU dont les éléments de  $L$  et  $U$  sont donnés par les éléments de l'élimination de Gauss.

### En pratique

$$L = (m_{ik}), \quad i \geq k, \quad U = (a_{kj}^{(k)}), \quad k \leq j$$
$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \rightarrow \begin{pmatrix} a_{11}^{(n)} & a_{12}^{(n)} & \cdots & a_{1,n-1}^{(n)} & a_{1n}^{(n)} \\ m_{21} & a_{22}^{(n)} & \cdots & a_{2,n-1}^{(n)} & a_{2n}^{(n)} \\ \vdots & \vdots & & \vdots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & a_{nn}^{(n)} \end{pmatrix}$$

### Sensibilité aux erreurs des données

Certains **problèmes linéaires** sont **mal conditionnés**.

Une petite perturbation des **données** entraîne une grosse variation de la solution.

Exemple :

Considérons le système

$$\begin{pmatrix} 51 & 30 & 50 \\ 102 & 59 & 100 \\ 152 & 90 & 149 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 4 \end{pmatrix}.$$

Solution :  $x = (-9 \ 2 \ 8)^T$ .

**Petite erreur** dans la matrice :

$$\begin{pmatrix} 50.99 & 30 & 50 \\ 102 & 59 & 100 \\ 152 & 90 & 149 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 4 \end{pmatrix}.$$

Nouvelle solution  $x = (-2.91 \ 1.94 \ 1.83)^T$

### Définition

On dit que  $\|x\|$  est une **norme vectorielle** si les axiomes suivants sont satisfaits.

- $\|x\| > 0$  pour tout  $x \neq 0$  et  $\|x\| = 0$  implique  $x = 0$
- $\|x + y\| \leq \|x\| + \|y\|$
- $\|\alpha x\| = |\alpha| \|x\|$  pour tout  $\alpha \in \mathbb{R}$

### Définition

On dit que  $\|A\|$  est une **norme matricielle** si les axiomes suivants sont satisfaits.

- $\|A\| > 0$  pour toute  $A \neq 0$ , et  $\|A\| = 0$  implique  $A = 0$
- $\|A + B\| \leq \|A\| + \|B\|$
- $\|\alpha A\| = |\alpha| \|A\|$  pour tout  $\alpha \in \mathbb{R}$

Si de plus, les deux axiomes supplémentaires suivants sont satisfaits

- $\|Ax\| \leq \|A\| \|x\|$
- $\|AB\| \leq \|A\| \|B\|$

alors on dit que la norme matricielle  $\|A\|$  est **compatible** avec la norme vectorielle  $\|x\|$ .

## Normes matricielles : exemples

### 4 normes matricielles les plus courantes

- $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$
- $\|A\|_2 = (\text{valeur propre maximum de } A^T A)^{1/2}$
- $\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$
- $\|A\|_F = \left( \sum_{i,j=1}^n |a_{ij}|^2 \right)^{1/2}$

### Exemple

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$\|A\|_1 = \max\{1 + 4 + 7, 2 + 5 + 8, 3 + 6 + 9\} = 18,$$

$$\begin{aligned} \|A\|_2 &= \max\{\text{val. propre de } A^T A\}^{1/2} \\ &= \max\{0, 1.07, 16.85\} \quad \approx 16.85 \end{aligned}$$

$$\|A\|_\infty = \max\{1 + 2 + 3, 4 + 5 + 6, 7 + 8 + 9\} = 24$$

$$\|A\|_F = \sqrt{1 + 4 + 9 + 16 + \dots + 81} \quad \approx 16.88.$$

# Effet des perturbations sur les données

## Nombre de conditionnement

$$\kappa(A) = \|A\| \|A^{-1}\|$$

## Proposition

Soit  $A \in \mathbb{R}^{n \times n}$  une matrice non singulière,  $b \in \mathbb{R}^n$  et la solution  $x \in \mathbb{R}^n$  de  $Ax = b$ . On peut borner l'erreur relative faite sur  $x$ , lorsque l'on résout  $Ax = (b + \delta b)$  au lieu de  $Ax = b$ , par

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A) \frac{\|\delta b\|}{\|b\|}.$$

## Proposition

Soit  $A \in \mathbb{R}^{n \times n}$  une matrice non singulière,  $b \in \mathbb{R}^n$  et la solution  $x \in \mathbb{R}^n$  de  $Ax = b$ . On peut borner l'erreur relative faite sur  $x$ , lorsque l'on résout  $(A + \delta A)x = b$  au lieu de  $Ax = b$ , par

$$\frac{\|\delta x\|}{\|x + \delta x\|} \leq \kappa(A) \frac{\|\delta A\|}{\|A\|}.$$

## Exemple de matrice mal conditionnée

$$A = \begin{pmatrix} 51 & 30 & 50 \\ 102 & 59 & 100 \\ 152 & 90 & 149 \end{pmatrix}$$

$$\|A\|_{\infty} = 391$$

$$A^{-1} = \begin{pmatrix} -209 & 30 & 50 \\ 2 & -1 & 0 \\ 212 & -30 & -51 \end{pmatrix}$$

$$\|A^{-1}\|_{\infty} = 293$$

$$\kappa(A) = 391 \times 293 = 114563$$

$$\frac{\|\delta A\|}{\|A\|} = \frac{0.01}{391} \approx 2.5 \cdot 10^{-5}$$

On a donc

$$\frac{\|\delta x\|}{\|x + \delta x\|} \leq 2.8$$

Or, on avait  $x = (-9 \ 2 \ 8)^T$  et  $\delta x \approx (7 \ 0.1 \ 6)^T$  et

$$\frac{\|\delta x\|}{\|x\|} \approx \frac{9}{12} \approx 0.75$$

### Proposition

Soit  $\bar{L} = (\bar{m}_{ik})$  et  $\bar{U} = (\bar{a}_{kj}^{(n)})$  les facteurs triangulaires calculés par l'algorithme d'élimination de Gauss. Alors il existe une matrice  $E$  telle que  $\bar{L}\bar{U}$  est la décomposition exacte de  $A + E$ , c'est-à-dire

$$\bar{L}\bar{U} = A + E$$

Si l'on a utilisé une stratégie de pivotage (partiel ou complet) et une arithmétique en virgule flottante avec un epsilon machine  $\epsilon_M$  alors la matrice  $E$  est bornée par

$$\|E\|_\infty \leq n^2 g_n \epsilon_M \|A\|_\infty$$

où

$$g_n = \frac{\max_{i,j,k} |\bar{a}_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}.$$

### Proposition

Soit le système linéaire  $Lx = b$  où la matrice  $L = (l_{ij})$  est triangulaire inférieure. Le vecteur  $\bar{x}$  calculé par substitution avant est la solution exacte du système triangulaire perturbé

$$(L + \delta L)\bar{x} = b$$

où la matrice de perturbation  $\delta L$  est bornée par

$$\|\delta L\|_{\infty} \leq \frac{n(n+1)}{2} \epsilon_M \max_{i,j} |l_{ij}|$$

## Changements d'échelle des équations

- Changement d'échelle d'une inconnue :  $x_j = \alpha_j x'_j$
- Changement d'unité du second membre : **multiplication de la  $j^e$  équation par  $\beta_j$**
- Nouveau système :  $A'x' = b'$  où  $x'_i = \frac{x_i}{\alpha_i}$  càd  $D_2 A D_1 x' = D_2 b$

$$\begin{pmatrix} \beta_1 & & \\ & \ddots & \\ & & \beta_n \end{pmatrix} A \begin{pmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_n \end{pmatrix} \begin{pmatrix} x'_1 \\ \vdots \\ x'_n \end{pmatrix} = \begin{pmatrix} \frac{b_1}{\beta_1} \\ \vdots \\ \frac{b_n}{\beta_n} \end{pmatrix}$$

### Théorème

Soient  $\bar{x}$  et  $\bar{x}'$  les solutions calculées des deux systèmes  $Ax = b$  et  $(D_2 A D_1)x' = D_2 b$ . Si  $D_1$  et  $D_2$  sont des matrices diagonales dont les éléments sont des puissances entières de la base du système de numération utilisé, de sorte que les changements d'échelles n'introduisent pas d'erreurs d'arrondi, alors l'élimination de Gauss effectuée en arithmétique à virgule flottante sur les deux systèmes produira, à *condition que l'on choisisse les mêmes pivots dans les deux cas*, des résultats qui ne diffèrent que par leurs exposants, et l'on aura exactement  $\bar{x} = D_1 \bar{x}'$ .

**Attention** si les changements d'échelle induisent d'autres choix de pivots !

- Dans certains cas, on a des systèmes linéaires énormes mais très **creux**
- Elimination de Gauss peu appropriée à cause du **fill in** : Matrice initiale :

$$\begin{pmatrix} \times & \times & \times & \cdots & \times \\ \times & \times & & \cdots & \\ \times & & \times & \cdots & \\ \vdots & & & \ddots & \\ \times & & & \cdots & \times \end{pmatrix} \begin{pmatrix} \times & \times & \times & \cdots & \times \\ \times & \times & \times & \cdots & \times \\ \times & \times & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \times & \times & \times & \cdots & \times \end{pmatrix}$$

Taux de remplissage :  $\frac{3n}{n^2}$  puis potentiellement 100 %

- On voudrait conserver une matrice creuse et effectuer des opérations **bon marché**.  
Exemple : multiplication matrice - vecteur.
- Principe des méthodes itératives.  
**Problème** : ne converge pas toujours !

## Méthodes itératives : principe

Idée : utiliser la **méthode de point fixe** vue pour les systèmes non linéaires !

On réécrit  $Ax = b$  comme

$$Qx = Qx - Ax + b \quad (1)$$

Pour **un choix judicieux** de  $Q$ , le système (1),

$$x^{(k+1)} = Q^{-1} \left( (Q - A)x^{(k)} + b \right)$$

- est **facile** à résoudre **à chaque itération**
- $x^{(k)}$  converge vers la **solution** de  $Ax = b$

## Méthodes itératives : convergence

1e question : comment garantir la convergence ?

### Proposition

Soit un système  $Ax = b$  dont la solution est  $\bar{x}$ . Si on applique le processus (1) à partir d'un point de départ  $x^{(1)}$ , l'erreur  $e^{(k)} := x^{(k)} - \bar{x}$  suit la règle

$$e^{(k)} = (I - Q^{-1}A)e^{(k-1)}.$$

- Choix de la matrice  $Q$  :
- Le mieux :  $Q = A$  (convergence en une itération) mais nécessite de résoudre le système
- On choisira  $Q$  proche de  $A$

## Méthodes itératives : méthode de Jacobi

- Premier type de système facile à résoudre : **systèmes diagonaux**  
Mène à la **méthode de Jacobi**.
- Choix de la matrice diagonale :  $Q = D = \text{diag}(A)$
- Résolution directe du système linéaire (en  $\mathcal{O}(n)$ )
- $x^{(k+1)} = (I - D^{-1}A)x^{(k)} + D^{-1}b$

$$x_1^{(k+1)} = \sum_{j=2}^n -\frac{a_{1j}}{a_{11}} x_j^{(k)} + \frac{b_1}{a_{11}}$$

⋮

$$x_n^{(k+1)} = \sum_{j=1}^{n-1} -\frac{a_{nj}}{a_{nn}} x_j^{(k)} + \frac{b_n}{a_{nn}}$$

- Utilise exclusivement les valeurs calculées à l'itération précédente

## Méthodes itératives : méthode de Gauss-Seidel

- Deuxième type de système facile à résoudre : **système triangulaire**  
Mène à la **méthode de Gauss-Seidel**
- Si  $A = L + D + U$ , on prend  $Q = L + D$
- Système à résoudre :  $Qx^{(k+1)} = (Q - A)x^{(k)} + b$

$$\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{pmatrix} =$$
$$\begin{pmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ 0 & 0 & -a_{23} & \cdots & -a_{2n} \\ 0 & 0 & 0 & \cdots & -a_{3n} \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$$



$$x_1^{(k+1)} = - \sum_{j=2}^n \frac{a_{1j}}{a_{11}} x_j^{(k)} + \frac{b_1}{a_{11}}$$

$$x_2^{(k+1)} = - \frac{a_{12}}{a_{22}} x_1^{(k+1)} - \sum_{j=3}^n \frac{a_{2j}}{a_{22}} x_j^{(k)} + \frac{b_2}{a_{22}}$$

## Méthode de Gauss-Seidel

Si on écrit explicitement, on obtient finalement

$$x_1^{(k+1)} = - \sum_{j=2}^n \frac{a_{1j}}{a_{11}} x_j^{(k)} + \frac{b_1}{a_{11}}$$

$$x_2^{(k+1)} = - \frac{a_{21}}{a_{22}} x_1^{(k+1)} - \sum_{j=3}^n \frac{a_{2j}}{a_{22}} x_j^{(k)} + \frac{b_2}{a_{22}}$$

⋮

$$x_i^{(k+1)} = - \frac{a_{i1}}{a_{ii}} x_1^{(k+1)} - \dots - \frac{a_{i,i-1}}{a_{ii}} x_{i-1}^{(k+1)} - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}$$

⋮

$$x_n^{(k+1)} = - \sum_{j=1}^{n-1} \frac{a_{nj}}{a_{nn}} x_j^{(k+1)} + \frac{b_n}{a_{nn}}$$

Très similaire à la méthode de Jacobi.

Mais on utilise à chaque calcul les **valeurs déjà calculées lors de l'itération courante** !

## Accélération de la méthode

Dans certains cas, on peut accélérer la méthode en appliquant un **facteur d'accélération**  $\omega$  bien choisi

$$Q = L + \omega D.$$

$\omega > 1$  : **surrelaxation**

$\omega < 1$  : **sous-relaxation**

Equivalent à

$$x^{(k+1)} = (1 - \omega)x^{(k)} + \omega g(x^{(k)})$$

où  $g(x)$  correspond à l'itéré calculé par Gauss-Seidel.

Si  $\omega = 1$ , correspond exactement à Gauss-Seidel.

## Conditions de convergence

### Proposition

Si les valeurs propres  $\lambda_i$  de  $I - Q^{-1}A$  sont telles que  $|\lambda_i| < 1$  pour tout  $i$ , alors la méthode itérative converge vers la solution  $\bar{x}$  du système  $Ax = b$  pour tout point de départ  $x^{(1)}$ .

### Proposition

Soit  $M = (m_{ij}) \in \mathbb{R}^{n \times n}$  une matrice qui a  $n$  vecteurs propres linéairement indépendants. Alors, pour chaque  $i = 1, \dots, n$ , il existe  $t \in \{1, \dots, n\}$  tel que

$$|\lambda_i| \leq \sum_{j=1}^n |m_{tj}|$$

où  $\lambda_i$  est une valeur propre de  $M$ .

### Proposition

On considère la système  $Ax = b$ . Si  $A$  est à diagonale dominante, c'est-à-dire si

$$\sum_{j \neq i} |a_{ij}| < |a_{ii}|$$

pour tout  $i = 1, \dots, n$ , alors les méthode de Jacobi et de Gauss-Seidel convergent pour tout point de départ  $x^{(1)}$ .