

# Algèbre linéaire numérique

## Chapitre 3 : Résolution de systèmes linéaires

J. Gergaud, S. Gratton & X. Vasseur



Département Sciences du Numérique

5 octobre 2021

$$Ax = b \quad (1)$$

On suppose que les données du système  $A$  et  $b$  sont soumises à des perturbations  $\Delta A$  et  $\Delta b$ . La perturbation  $\Delta x$  résultante satisfait l'équation

$$(A + \Delta A)(x + \Delta x) = b + \Delta b, \text{ avec } Ax = b. \quad (2)$$

Soit  $\|\cdot\|$  la norme euclidienne ou sa norme matricielle induite.

## Proposition

*Montrez qu'au premier ordre, on a l'inégalité*

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\| \|A^{-1}\| \left( \frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right).$$

## Proposition

*Montrez qu'il est possible d'obtenir le résultat de perturbation suivant, sans se placer au premier ordre, mais en supposant que la perturbation  $\Delta A$  est bornée. Si  $\|\Delta A\| \|A^{-1}\| \leq 1/2$ , on a*

$$\frac{\|\Delta x\|}{\|x\|} \leq 2\|A\| \|A^{-1}\| \left( \frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right).$$

## Définition

On appelle *conditionnement* de la matrice  $A$  la quantité

$$\|A^{-1}\| \|A\|.$$

Supposons à présent que l'on ait à notre disposition une approximation  $\tilde{x}$  de  $x$ , obtenue par exemple (mais pas nécessairement) sur un ordinateur.

## Définition

On appelle *erreur inverse* associée à  $\tilde{x}$  la quantité

$$\eta(\tilde{x}) = \frac{1}{\|A\|} \min (\|\Delta A\| \text{ tels que } (A + \Delta A)\tilde{x} = b) .$$

Par analogie, l'erreur de calcul  $\|\Delta x\|/\|x\| = \|\tilde{x} - x\|/\|x\|$  s'appelle aussi *erreur directe*.

L'erreur inverse mesure (en norme relative) la distance du problème exact au problème perturbé que  $\tilde{x}$  résout exactement. Elle détermine la mesure relative  $\frac{\|\Delta A\|}{\|A\|}$  de la perturbation de  $A$  *équivalente* au calcul de la solution  $\tilde{x}$ .

On suppose que  $\tilde{x}$  est le résultat d'un calcul sur ordinateur.

- le calcul de  $\tilde{x}$  est **fiable** si l'erreur inverse associée est de l'ordre de la précision machine  $\epsilon$ , soit

$$\eta(\tilde{x}) \sim C\epsilon,$$

où  $C$  est une constante *pas trop grande*, qui peut dépendre des données du problème (ici  $A$ ,  $b$ ,  $n$ ).

- $A$  et/ou le second membre  $b$  sont entachés d'erreur le calcul de  $\tilde{x}$  est **fiable** lorsque l'erreur inverse associée est de l'ordre de ces erreurs.

## Proposition

*Soit  $r = A\tilde{x} - b$  le vecteur résiduel associé à  $\tilde{x}$ . L'erreur inverse en  $\tilde{x}$  est déterminée par la formule*

$$\eta(\tilde{x}) = \frac{\|r\|}{\|A\|\|\tilde{x}\|}.$$

*Si  $\eta(\tilde{x})\|A\|\|A^{-1}\| \leq 1/2$  alors*

$$\frac{\|\Delta x\|}{\|x\|} \leq 2\|A\|\|A^{-1}\|\eta(\tilde{x}).$$

$$\begin{array}{ccccc} \text{erreur directe} & \leq & \text{conditionnement} & \times & \text{erreur inverse en } \tilde{x} \\ & & \downarrow & & \downarrow \\ & & \text{problème} & & \text{algorithme de calcul} \\ & & \text{mathématique} & & \text{en précision finie} \end{array}$$

Si l'erreur directe est grande, cela peut être dû au problème à résoudre (conditionnement grand) et/ou à l'algorithme (grande erreur inverse). **Le rôle de l'erreur inverse est de permettre de distinguer dans l'erreur directe entre le facteur dû au problème et le facteur dû à l'algorithme.**

$$A = LU$$

- $A$  est une matrice rectangulaire de  $\mathcal{M}_{m,n}(\mathbf{R})$ .
- $A_k$  désigne la sous-matrice principale  $A$  d'ordre  $k$ ,  $k = 1, \dots, \min(m, n)$ .
- $L \in \mathcal{M}_{m,m}(\mathbf{R})$  dénote une matrice triangulaire inférieure (carrée) à éléments diagonaux égaux à 1.
- $U \in \mathcal{M}_{m,n}(\mathbf{R})$  désigne une matrice triangulaire supérieure rectangulaire :  $u_{ij} = 0$  si  $i > j$ .

Appliquons l'algorithme du pivot vu dans les classes antérieures à la matrice

$$A = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 11 \end{pmatrix}$$



## Algorithme $LU$ : exemple

$$A = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 11 \end{pmatrix}.$$

En utilisant 2 comme pivot pour la deuxième ligne et 3 pour la troisième ligne, on obtient

$$\begin{pmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -10 \end{pmatrix}.$$

On note que

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -10 \end{pmatrix}.$$

## Algorithme $LU$ : exemple

En utilisant 2 comme pivot dans la troisième ligne, on obtient

$$\begin{pmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 2 \end{pmatrix}.$$

On note que

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 2 \end{pmatrix},$$

et que la matrice triangulaire inférieure contient les multiplicateurs.

# Algorithme de Gauß sans pivotage

## Proposition

*Si les  $A_k$  sont inversibles pour  $k = 1, \dots, s = \min(m-1, n)$ , alors il existe  $L$  et  $U$  telles que  $A = LU$ . De plus,  $u_{kk} = \det A_k / \det A_{k-1}$ . L'algorithme ci-dessous réalise cette tâche en  $2\frac{n^3}{3}$  opérations si  $A \in \mathcal{M}_{n,n}(\mathbb{R})$ .*

### Algorithme de factorisation de Gauss (sans pivotage)

```
Pour  $k = 1$  à  $s = \min(m-1, n)$ 
  si  $a_{kk} = 0$  alors exit
  sinon
     $w := a_{k,k+1:n}$ 
    pour  $i = k+1$  à  $m$ 
       $a_{ik} := a_{ik}/a_{kk}$ 
       $\eta := a_{ik}$ 
       $a_{i,k+1:n} := a_{i,k+1:n} - \eta w$ 
    finpour
  finsi
finpour
```

Un système triangulaire d'ordre  $n$  se résout par substitution en  $n^2$  opérations. Ainsi la résolution de  $Ax = b$  lorsque  $A = LU$  se fait par

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

# Exemple

Soit

$$A = \begin{pmatrix} x & x & x & x \\ x & x & 0 & 0 \\ x & 0 & x & 0 \\ x & 0 & 0 & x \end{pmatrix}$$

- 1 Combien faut-il d'opérations pour obtenir la factorisation  $LU$  via l'algorithme de Gauß sans pivotage ?
- 2 Si on échange les première et dernière lignes ainsi que les première et dernière colonnes, combien faut-il d'opérations pour obtenir la factorisation  $LU$  via l'algorithme de Gauß sans pivotage ?

Considérons le système

$$\begin{cases} 10^{-5}x + y = 1 \\ x + y = 2 \end{cases}.$$

- A cause du petit coefficient  $10^{-5}$ , il est clair que la solution doit être proche de  $x = y = 1$ . Supposons que l'on résolve ce système sans pivotage avec une arithmétique à 4 chiffres. Quelle solution obtient-on ?

Considérons le système

$$\begin{cases} 10^{-5}x + y = 1 \\ x + y = 2 \end{cases}.$$

- A cause du petit coefficient  $10^{-5}$ , il est clair que la solution doit être proche de  $x = y = 1$ . Supposons que l'on résolve ce système sans pivotage avec une arithmétique à 4 chiffres. Quelle solution obtient-on ?
- Par soustraction, il vient  $(1 - 10^5)y = 2 - 10^5$ . Or dans une arithmétique à quatre chiffres, les quantités  $1 - 10^5$  et  $2 - 10^5$  se représentent par  $-10^5$ , donc  $10^5 y = 10^5$ , soit  $y = 1$ . Mais alors  $10^{-5} x + 1 = 1$  implique  $x = 0$  (très différent de 1 !).

- Avec pivotage maintenant. On échange les deux équations, soit

$$\begin{cases} x + y = 2 \\ 10^{-5}x + y = 1 \end{cases}.$$

Quelle solution obtient-on ? Calculez dans les deux l'erreur inverse associée.

- Avec pivotage maintenant. On échange les deux équations, soit

$$\begin{cases} x + y = 2 \\ 10^{-5}x + y = 1 \end{cases}.$$

Quelle solution obtient-on ? Calculez dans les deux l'erreur inverse associée.

- On échange les deux équations, soit

$$\begin{cases} x + y = 2 \\ 10^{-5}x + y = 1 \end{cases}.$$

Donc  $(1 - 10^{-5}) y = 1 - 2 \cdot 10^{-5}$  entraîne encore  $y = 1$ . Mais cette fois-ci  $x + 1 = 2$  implique  $x = 1$ . Le seul fait de pivoter a remplacé un résultat faux par un résultat satisfaisant.





$$\begin{aligned}\|A\| &= \sqrt{\rho(A^T A)} \text{ avec } A = \begin{pmatrix} 10^{-5} & 1 \\ 1 & 1 \end{pmatrix} \\ &\simeq 1.61\end{aligned}$$



$$\begin{aligned}\|A\| &= \sqrt{\rho(A^T A)} \text{ avec } A = \begin{pmatrix} 10^{-5} & 1 \\ 1 & 1 \end{pmatrix} \\ &\simeq 1.61\end{aligned}$$

- Pour  $z_1 = (0, 1)^T$ ,  $r_1 = Az_1 - b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$  et donc

$$\eta(z_1) = \frac{\|r_1\|}{\|A\|\|z_1\|} = \frac{1}{\|A\|} \simeq 6.2 \cdot 10^{-1}$$



$$\begin{aligned}\|A\| &= \sqrt{\rho(A^T A)} \text{ avec } A = \begin{pmatrix} 10^{-5} & 1 \\ 1 & 1 \end{pmatrix} \\ &\simeq 1.61\end{aligned}$$

- Pour  $z_1 = (0, 1)^T$ ,  $r_1 = Az_1 - b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$  et donc

$$\eta(z_1) = \frac{\|r_1\|}{\|A\|\|z_1\|} = \frac{1}{\|A\|} \simeq 6.2 \cdot 10^{-1}$$

- Pour  $z_2 = (1, 1)^T$ ,  $r_2 = Az_2 - b = \begin{pmatrix} 10^{-5} \\ 0 \end{pmatrix}$  et donc

$$\eta(z_2) = \frac{\|r_2\|}{\|A\|\|z_2\|} = \frac{10^{-5}}{\sqrt{2}\|A\|} \simeq 4.4 \cdot 10^{-6}$$

Or sur une arithmétique à 4 chiffres décimaux correspond à une précision machine  $\Psi = 10^{-4+1} = 10^{-3}$ . Seule l'erreur inverse  $\eta(z_2)$  est de l'ordre de  $10^{-6}$

L'algorithme suivant réalise une factorisation de Gauss avec pivotage :

### Algorithme de factorisation de Gauss avec pivotage partiel

Pour  $k = 1$  à  $n - 1$

déterminer  $p \in \{k, \dots, n\}$  tel que  $|a_{pk}| = \max_{k \leq i \leq n} |a_{ik}|$

$r_k := p$

échanger  $a_{k,1:n}$  et  $a_{p,1:n}$

$w := a_{k,k+1:n}$

pour  $i = k + 1$  à  $n$

$a_{ik} := a_{ik} / a_{kk}$

$\eta := a_{ik}$

$a_{i,k+1:n} := a_{i,k+1:n} - \eta w$

finpour

finpour

### Proposition

*L'algorithme ci-dessus détermine pour toute matrice  $A$  carrée inversible, une matrice de permutation  $P = P_{n-1} \dots P_1$  telle que  $PA = LU$*

- i) nous avons considéré le pivotage *partiel* où seul l'ordre des équations peut être modifié,
- ii) le pivotage *total* où l'ordre des équations et celui des variables peut être modifiés.

# Algorithme de Gauss avec pivotage total

L'algorithme ci-dessous détermine des matrices de permutation  $P = P_{n-1} \dots P_1$  et  $\Pi = \Pi_1 \dots \Pi_{n-1}$  telles que  $PA\Pi = LU$ .

## Algorithme de factorisation de Gauss avec pivotage total

```
Pour  $k = 1$  à  $n - 1$ 
  déterminer  $p$  et  $q \in \{k, \dots, n\}$  tels que  $|a_{pq}| = \max_{\substack{k \leq i \leq n \\ k \leq j \leq n}} |a_{ij}|$ 

   $r_k := p$ 
   $c_k := q$ 
  échanger  $a_{k,1:n}$  et  $a_{p,1:n}$ 
  échanger  $a_{1:n,k}$  et  $a_{1:n,q}$ 
   $w = a_{k,k+1:n}$ 
  pour  $i = k + 1$  à  $n$ 
     $a_{ik} := a_{ik} / a_{kk}$ 
     $\eta := a_{ik}$ 
     $a_{i,k+1:n} := a_{i,k+1:n} - \eta w$ 
  finpour
finpour
```

## Définition

On appelle facteur de croissance de la factorisation de Gauss la quantité

$$\rho_n = \max_{i,j,k} \frac{|\tilde{a}_{ij}|^{(k)}}{\|A\|_\infty}$$

où  $\tilde{A}^{(k)}$  est la version calculée de  $A^{(k)} = (a_{ij}^{(k)})$  à l'étape  $k$ .

Soit  $W_n$  la matrice de Wilkinson de taille  $n$

$$W_n = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ \vdots & & \ddots & 1 & 1 \\ -1 & \cdots & \cdots & -1 & \alpha \end{pmatrix} \quad \text{avec } \alpha = 0.9.$$

On considère  $b = (1, \dots, 1)^T$ , alors la solution exacte est  $x$  défini par ( $\Delta = 2^{n-1} - 1 + \alpha$ )

$$\begin{cases} x_i = -2^{i-1} \frac{1-\alpha}{\Delta} \text{ pour } i = 1 \dots n-1 \\ x_n = \frac{2^{i-1}}{\Delta} \end{cases},$$

# Expérimentation numérique

On note  $K = \|A\| \|A^{-1}\|$ ,  $R = \|A\tilde{x} - b\|$ ,  $EI = \frac{R}{\|A\| \|\tilde{x}\|}$  représente l'erreur inverse et  $ED = \frac{\|x - \tilde{x}\|}{\|x\|}$  l'erreur directe.

$n$	$K$	Pivotage partiel			Pivotage total		
		$R$	$EI$	$ED$	$R$	$EI$	$ED$
10	4.45	$4.2 \cdot 10^{-14}$	$6.8 \cdot 10^{-15}$	$2.4 \cdot 10^{-14}$	$6.7 \cdot 10^{-16}$	$1.1 \cdot 10^{-16}$	$1.0 \cdot 10^{-16}$
20	8.89	$4.7 \cdot 10^{-12}$	$3.7 \cdot 10^{-13}$	$2.7 \cdot 10^{-12}$	0	0	$6.5 \cdot 10^{-17}$
30	13.6	$2.4 \cdot 10^{-08}$	$1.3 \cdot 10^{-09}$	$1.4 \cdot 10^{-08}$	$2.2 \cdot 10^{-16}$	$1.2 \cdot 10^{-17}$	$7.8 \cdot 10^{-17}$
40	18.1	$2.4 \cdot 10^{-05}$	$9.7 \cdot 10^{-07}$	$1.4 \cdot 10^{-05}$	$6.8 \cdot 10^{-16}$	$2.8 \cdot 10^{-17}$	$7.8 \cdot 10^{-17}$
50	22.7	$2.5 \cdot 10^{-02}$	$7.9 \cdot 10^{-04}$	$1.4 \cdot 10^{-02}$	$2.2 \cdot 10^{-16}$	$7.0 \cdot 10^{-18}$	$7.8 \cdot 10^{-17}$

Il apparaît très clairement que sur cette matrice, dès  $n = 30$ , il est nécessaire d'utiliser une stratégie de pivotage total pour que le calcul soit fiable et donne une solution proche de la solution exacte du système. Rappelons néanmoins que dans la plupart des cas, le pivotage partiel suffit pour obtenir un calcul fiable.



# Factorisation de Cholesky d'une matrice symétrique définie positive

## Proposition

*Toute matrice  $A$  symétrique définie positive admet une factorisation de Cholesky  $A = C^T C$ , où  $C$  est une matrice triangulaire inférieure avec ces éléments diagonaux strictement positifs, obtenue grâce à l'algorithme de Cholesky qui coûte  $n^3/3$  opérations flottantes.*

## Proposition

*Soit  $A$  une matrice définie positive. Montrez par identification dans l'équation  $A = CC^T$  que, pour  $k \geq i$   $c_{ki}c_{ji} = a_{ik} - \sum_{p=1}^{i-1} c_{ip}c_{kp}$  et  $c_{ii} = \sqrt{a_{ii} - \sum_{p=1}^{i-1} c_{ip}c_{ip}}$ . Et en déduire un algorithme de calcul de  $C$ .*

# Factorisation de Cholesky d'une matrice symétrique définie positive

## Algorithme de la factorisation de Cholesky

Pour  $i = 1$  à  $n$

$$c_{ii} := \sqrt{a_{ii} - \sum_{p=1}^{i-1} c_{ip}^2}$$

Pour  $k = i + 1$  à  $n$

$$c_{ki} := (a_{ik} - \sum_{p=1}^{i-1} c_{ip}c_{kp})/c_{ii}$$

finpour

finpour

On factorise  $A = C^T C$ , et la solution de  $Ax = b$  s'obtient par

$$CC^T x = b \iff \begin{cases} Cy = b \\ C^T x = y. \end{cases}$$

# Factorisation de Cholesky d'une matrice symétrique définie positive

## Théorème

*En arithmétique exacte,  $A$  est symétrique définie positive si et seulement si l'algorithme se déroule complètement.*

## Théorème

*Si on utilise l'algorithme de Cholesky sur un ordinateur (précision  $\epsilon$ ) pour résoudre  $Ax = b$ , alors*

- *soit l'algorithme lève une exception  $(a_{ii} - \sum_{p=1}^{i-1} c_{ip}^2) < 0$ .*
- *soit il produit une solution  $\tilde{x}$  qui est solution exacte du système perturbé  $(A + \Delta A)\tilde{x} = b$  et*
  - 1  $\|\Delta A\| \leq c_n \epsilon \|A\|$
  - 2 Si  $q_n \|A\| \|A^{-1}\| \epsilon \leq 1$  alors pas d'arrêt, où  $q_n$  est un polynôme de faible degré en  $n$ .