ALGORITMOS EN ÁLGEBRA LINEAL

Notas de curso (UBA - 2do cuatrimestre de 2005)

http://atlas.mat.ub.es/personals/sombra/curso.html

Michelle Schatzman

Martín Sombra

Institut Camille Jordan (Mathématiques), Université de Lyon 1 ; 43 Bd. du 11 Novembre 1918, 69622 Villeurbanne Cedex, Francia

Universitat de Barcelona, Departament d'Àlgebra i Geometria ; Gran Via 585, 08007 Barcelona, España

CHAPITRE 4

Résolution de systèmes linéaires généraux

L'algorithme d' élimination de Gauss c'est l'exemple classique de méthode *directe* de résolution, dont dans l'absence d'erreur d'arrondi, il fourni la solution exacte de $Ax = b$ au bout d'un temps fini. Cet algorithme fut esquissé par Gauss dans [**4**] puis décrit de façon explicite en 1823 dans [**5**, § 31]. C'est un fait remarquables que ces développements sont bien antérieurs à l'utilisation de la notation matricielle ; dans le cas de Gauss la factorisation LU fut exprimée en termes de formes quadratiques.

Sa complexité $2N^3/3$ n'est pas optimale. Strassen [**10**] a montré qu'on peut résoudre $Ax = b$ avec une méthode de type "divide and conquer" avec $O(N^\beta)$ ops, ou β est l'exposant de la multiplication de deux matrices de taille $N \times N$. À l'heure actuelle on sait que $2 \leq \beta \leq 2.37$ grâce à l'algorithme de Winograd [**1**]. Malgré quelques efforts pour rendre l'algorithme de Strassen praticable pour l'exposant $\beta = \log_2(7) = 2.78$ [**8**] ; l'algorithme d'élimination reste la méthode de choix pour la résolution de systèmes linéaires non structurés ou quand on veut un algorithme stable avec un temps d'exécution garanti.

Dans ce chapitre on fait l'étude détaillé de cet algorithme. On le décrit matriciellement, puis on compte le nombre d'opérations arithmétiques sans et avec pivotage, ce qui nous donnera son coût en arithmétique flottante. En estimant la taille des calculs intermédiaires, on obtient sa complexité en calcul exacte.

Dans une seconde étape, on introduit la notion de conditionnement d'un système linéaire et on étude l'erreur *a priori* et *a posteriori* d l'algorithme d'élimination. Finalement on donne un aperçu sans démonstrations de l'application de cet algorithme au cas des matrices bande.

Références pour cette section : [**2, 6, 9**].

1. L'algorithme d'élimination sans pivotage

Pour gagner en simplicité on se restreindra au cas d'une matrice carrée inversible $A = [a_{i,j}]_{1 \leq i,j \leq N} \in \mathbb{F}^{N \times N}$ et $b \in \mathbb{R}^N$; cependant l'algorithme marche aussi bien pour des matrices rectangulaires quelconques. Par exemple

$$\begin{aligned} x + 2y &= 1, \\ 2x - y &= 1. \end{aligned}$$

Pour le résoudre, l'algorithme d'élimination soustrait deux fois la première équation à la deuxième pour éliminer la variable x dans cette dernière. On abouti à un système triangulaire

$$\begin{aligned} x + 2y &= 1, \\ -5y &= -1 \end{aligned}$$

qu'on résout par *backward substitution* en $y = 1/5$ et $x = 1 - 2y = 3/5$.

1.1. Interprétation matricielle. Écrivons cela en général et considérons d'abord la situation *sans pivotage*. Supposons que $\pi_1 := a_{1,1} \neq 0$ et posons

$$a' := \begin{bmatrix} a_{2,1} \\ \vdots \\ a_{N,1} \end{bmatrix}, \quad p' := \frac{1}{\pi_1} a', \quad \ell' := \begin{bmatrix} a_{1,2} \cdots a_{1,N} \end{bmatrix},$$

et $A_{2,2} = [a_{i,j}]_{2 \leq i,j \leq N-1} \in \mathbb{F}^{(N-1)\times(N-1)}$ la matrice A privée des premières ligne et colonne. Soit

$$\widehat{M} := \begin{bmatrix} 1 & \\ -p' & \mathbf{1}_{N-1} \end{bmatrix},$$

alors

$$\widehat{M} \cdot A = \begin{bmatrix} 1 & \\ -p' & \mathbf{1}_{N-1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 & \ell' \\ a' & A_{2,2} \end{bmatrix} = \begin{bmatrix} \pi_1 & \ell' \\ 0 & A_{2,2} - p' \cdot \ell' \end{bmatrix}. \tag{1}$$

La sous-matrice $S_1 = A_{2,2} - p' \cdot \ell' \in \mathbb{F}^{(N-1)\times(N-1)}$ est le *complément de Schur*; on a

$$(S_1)_{i,j} = (A_{2,2} - p' \cdot \ell')_{i,j} = a_{i,j} - p'_i \ell'_j = a_{i,j} - a_{i,1} a_{1,1}^{-1} a_{1,j}, \qquad (2 \leq i, j \leq N)$$

donc (1) est bien l'algorithme d'élimination classique. Également on doit multiplier b :

$$\widehat{M} \cdot b = \begin{bmatrix} 1 & \\ -p' & \mathbf{1}_{N-1} \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b' \end{bmatrix} = \begin{bmatrix} b_1 \\ b' - p' \cdot b_1 \end{bmatrix}. \tag{2}$$

Le système originel $Ax = b$ est équivalent au système $\widehat{M} \cdot Ax = \widehat{M} \cdot b$ car $\widehat{M}$ est inversible. À partir de la solution $x' = (x_2, \ldots, x_N)$ de $S_1 x' = b' - p' \cdot b_1$ on obtient x_1 comme

$$x_1 = b_1 - \ell' \cdot x'.$$

On vérifie que

$$\widehat{L} := \widehat{M}^{-1} = \begin{bmatrix} 1 & \\ p' & \mathbf{1}_{N-1} \end{bmatrix}$$

donc

$$A = \begin{bmatrix} 1 & \\ p' & \mathbf{1}_{N-1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 & \ell' \\ & S_1 \end{bmatrix}.$$

Si le deuxième pivot $\pi_2 := s_{2,2}$ est non nul, on peut appliquer le même procédé à S_1 et remplir la deuxième colonne de $\widehat{M} \cdot A$ avec des zéros à partir de la troisième ligne. Le procédé ne change pas ni la première ni la deuxième lignes de $\widehat{M} \cdot A$. Pourvu qu'on ait la chance de ne pas croisser aucun pivot nul dans notre chemin, on obtient par récurrence un système de type

$$\begin{bmatrix} \pi_1 & * & \cdots & * \\ 0 & \pi_2 & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \pi_N \end{bmatrix} \cdots \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}$$

équivalent au système originel $Ax = b$.

Définition 1.1. Soit $A \in \mathbb{F}^{N\times N}$ et pour $1 \leq k \leq N$ notons $A^{(k)}$ le bloc principal

$$A^{(k)} = [a_{i,j}]_{1 \leq i,j \leq k} \in \mathbb{F}^{k\times k}.$$

On dit que A est *fortement inversible* si $A^{(k)}$ est inversible pour tout $1 \leq k \leq N$.

PROPOSITION 1.2. *Soit $A \in \mathbb{F}^{N\times N}$, alors A est fortement inversible si et seulement si tous les pivots successifs au cours de l'élimination sont non nuls. Dans ce cas, il existe des uniques matrices $L \in \mathbb{F}^{N\times N}$ triangulaires inférieure avec 1s dans la diagonal, et $U \in \mathbb{F}^{N\times N}$ triangulaire supérieure telles que*

$$A = L \cdot U.$$

DÉMONSTRATION. Avec les notations précédentes

$$A = \begin{bmatrix} 1 & \\ p' & \mathbf{1}_{N-1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 & \ell' \\ & S_1 \end{bmatrix}$$

donc pour $1 \leq k \leq N-1$ on a

$$\det A^{(k+1)} = \pi_1 \cdot \det S_1^{(k)}.$$

On en déduit que$A$ est fortement inversible si et seulement si il est de même pour le complément de Schur S_1 et $\pi_1 \neq 0$, et par récurrence si et seulement si $\pi_i \neq 0$ pour $1 \leq i \leq N$.

Supposons que A est fortement inversible, et procédons par récurrence. Soit $S_1 = L_2 \cdot U_2$ la décomposition LU en dimension $N-1$, alors

$$\begin{aligned} A &= \begin{bmatrix} 1 & \\ p' & \mathbf{1}_{N-1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 & \ell' \\ & L_2 \cdot U_2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & \\ p' & \mathbf{1}_{N-1} \end{bmatrix} \cdot \begin{bmatrix} 1 & \\ & L_2 \end{bmatrix} \cdot \begin{bmatrix} \pi_1 & \ell' \\ & U_2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & \\ p' & L_2 \end{bmatrix} \cdot \begin{bmatrix} \pi_1 & \ell' \\ & U_2 \end{bmatrix}. \end{aligned}$$

Pour l'unicité, soient $L', U' \in \mathbb{R}^{N\times N}$ respectivement triangulaire inférieure avec 1s dans la diagonal et triangulaire supérieure telles que $A = L' \cdot U'$, alors

$$(L')^{-1} \cdot L = U' \cdot U^{-1}$$

c'est une matrice triangulaire inférieure avec 1s dans la diagonal et triangulaire supérieure à la fois. La seule possibilité est $\mathbf{1}_N$ donc $L' = L$ et $U' = U$. □

La matrice L contient beaucoup d'information :

PROPOSITION 1.3. *Le coefficient $L_{i,j}$ ($1 \leq j < i \leq N$) est la valeur par laquelle on multiplie la j-ème ligne pour la soustraire à la i-ème ligne, dans j-ème pas de l'algorithme d'élimination.*

DÉMONSTRATION. La matrice A peut s'écrire comme

$$A = \widehat{L}_1 \cdots \widehat{L}_{N-1} \cdot U$$

avec

$$\widehat{L}_j = \begin{bmatrix} \mathbf{1}_{j-1} & & \\ & 1 & \\ & p'_j & \mathbf{1}_{N-j} \end{bmatrix}$$

correspondant au j-ème pas de l'algorithme d'élimination, où $(p'_j)_i$ ($1 \leq j < i \leq N$) est la valeur par laquelle on multiplie la j-ème ligne pour la soustraire à la i-ème ligne, et on vérifie que

$$\mathrm{col}_j(L) = \mathrm{col}_j(\widehat{L}_1 \cdots \widehat{L}_{N-1}) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ p'_j \end{bmatrix}.$$

□

EXERCICE 4.1. ◁ Soit

$$A = \begin{matrix} \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} & \begin{matrix} i \\ N-i \end{matrix} \\ \begin{matrix} i & N-i \end{matrix} & \end{matrix}$$

une matrice fortement inversible. En particulier $A_{1,1}$ est inversible, et on posera

$$S_i := A_{2,2} - A_{2,1} \cdot A_{1,1}^{-1} \cdot A_{1,2} \in \mathbb{F}^{(N-i)\times(N-i)}$$

pour son *i-ème complément de Schur*. Montrer que S_i coïncide avec le bloc correspondant dans la matrice A après i pas de l'algorithme d'élimination de Gauss. ▷

1.2. L'algorithme d'élimination en pratique. A partir de la décomposition $A = L \cdot U$ la résolution d'un système $Ax = b$ se fait en deux étapes :

(1) on résout $Ly = b$;
(2) on résout $Ux = y$.

En effet $Ax = LUx = Ly = b$. Chacun de ces systèmes est facile à résoudre par substitution successive puisque triangulaires. La partie la plus lourde du point de vue des calculs est la construction de L et de U. Remarquons que si on doit résoudre plusieurs systèmes avec la matrice A, il faut garder la factorisation LU pour ne pas avoir à la recalculer à chaque fois.

Voici l' algorithme d'élimination sans pivotage en pseudo-code :

Algorithme de factorisation LU sans pivotage :
Entrée : $a_{i,j}$ **pour** $1 \leq i, j \leq N$ **;**
Sortie : $\ell_{i,j}, u_{i,j}$ **pour** $1 \leq i, j \leq N$**.**
For i **from** 1 **to** $N-1$ **do**

(1) for j **from** i **to** N **do**

$$\ell_{j,i} \leftarrow a_{j,i}/a_{i,i} \quad , \quad u_{i,j} \leftarrow a_{i,j};$$

od ;

(2) for j, k **from** $i+1$ **to** N **do**

$$a_{j,k} \leftarrow a_{j,k} - \ell_{j,i} \cdot u_{i,k};$$

od ;

od ;

$$\ell_{N,N} \leftarrow 1 \quad , \quad u_{N,N} \leftarrow a_{N,N};$$

end.

Notons qu'une fois que la i-ème colonne de A est utilisée pour calculer la i-ème colonne de L, elle n'est plus réutilisée. Similairement, la i-ème ligne de A n'est plus utilisée après le calcul de la i-ème ligne de U. Ceci permets d'écrire L et U sur A au fur et mesure qu'on les calcule, et donc on n'a pas besoin d'espace additionnel pour les stocker : L et U occupent respectivement le triangle inférieur et supérieur de A.

Cette observation simplifie l'algorithme, et on le réduit encore en utilisant la notation Matlab :

Algorithme de factorisation LU sans pivotage, réécrivant L et U sur A :
Entrée : $A \in \mathbb{F}^{N\times N}$ **;**
Sortie : Réécriture de L, U sur A.
For i **from** 1 **to** $N-1$ **do**

(1) $A(i+1:N, i) = A(i+1:N, i)/A(i,i)$ **;**

(2) $A(i{+}1:N, i{+}1:N) = A(i{+}1:N, i{+}1:N) - A(i{+}1:N, i)\cdot A(i, i{+}1:N)$ **;**

od ; end.

1.3. Complexité. Estimons le nombre d'opérations arithmétiques requises par cet algorithme, ce qui reviendra à estimer sa complexité en calcul flottant :

PROPOSITION 1.4. *Soit $A \in \mathbb{F}^{N\times N}$ une matrice fortement inversible, alors la factorisation LU via l'algorithme d'élimination sans pivotage se fait en $2N^3/3$ ops.*

DÉMONSTRATION. **Pas 1.** Pour chaque $1 \le i \le N-1$ on fait $N-i$ divisions (on ne compte pas l'assignation triviale $\ell_{i,i} \leftarrow a_{i,i}/a_{i,i} = 1$), soit au total

$$\sum_{i=1}^{N-1} N - i = \frac{N(N-1)}{2} \quad \text{ops.}$$

Pas 2. Pour chaque $1 \le i \le N-1$ on fait $2(N-i)^2$ opérations, soit au total

$$\sum_{i=1}^{N-1} 2(N-i)^2 = 2\sum_{i=1}^{N-1} i^2 = \frac{(N-1)N(2N-1)}{6} \quad \text{ops;}$$

donc la factorisation LU demande

$$\frac{N(N-1)}{2} + \frac{(N-1)N(2N-1)}{6} \le \frac{2}{3}N^3 \quad \text{ops.}$$

□

Considérons maintenant la résolution de $Ax = b$. Le système $Ly = b$ s'écrit comme

$$\begin{aligned} y_1 &= b_1 \\ \ell_{2,1}y_1 + y_2 &= b_2 \\ &\vdots \\ \ell_{N,1}y_1 + \ell_{N-1,1}y_2 + \cdots + y_N &= b_N. \end{aligned}$$

Pour résoudre la première équation on fait 0 ops, pour la deuxième on fait 2 ops, et en général pour résoudre la j-ème équation on fait $2(j-1)$ ops. Au total, la résolution de $Ly = b$ demande

$$\sum_{j=1}^{N} 2(j-1) = (N-1)N \quad \text{ops.}$$

Le système $Ux = y$ est similaire, sauf qu'on fait une opération de plus par ligne (l'inversion de $s_{j,j}$) donc sa résolution demande

$$(N-1)N + N = N^2$$

ops. On obtient :

PROPOSITION 1.5. *La résolution de $Ax = b$ à partir de la factorisation $A = L\cdot U$ se fait en $2N^2$ ops.*

Si on veut résoudre un système $1,000 \times 1,000$ *via* élimination gaussienne, cela nous demandera

$$2 \times 1,000^3/3 + 2 \times 1,000^2 \sim 7 \times 10^8 \quad \text{ops,}$$

ce qui prendra moins d'une seconde à 10^9 flops (*floating point operations per second*). Par contre, résoudre un système $1,000,000 \times 1,000,000$ nous demandera

$$2 \times 1,000,000^3/3 + 2 \times 1,000,000^2 \sim 7 \times 10^{17} \quad \text{ops,}$$

ce qui prendra plus de 21 ans à 10^9 flops.

1.4. Faut-il inverser des matrices ? L'inverse de $A \in \mathbb{F}^{N\times N}$ est la matrice A^{-1} dont l'image des vecteurs e_j de la base standard est la colonne v_j de A correspondante. Le calcul de cette inverse revient donc à résoudre les systèmes

$$Ax = v_j \quad , \quad 1 \leq j \leq N.$$

À l'aide de la factorisation LU, la résolution de ces systèmes se fait en deux étapes

$$Lw_j = e_j \quad , \quad Uv_j = w_j.$$

Quel est le coût de ces résolutions ? Le système triangulaire $Lw_j = e_j$ est assez spécial. On sait *a priori* que $(w_j)_k = 0$ pour $1 \leq k \leq j-1$, donc ce système se réduit à

$$\begin{array}{lcl} w_j & = & 1 \\ \ell_{j+1,j}w_j + w_{j+1} & = & 0 \\ & \vdots & \\ \ell_{N,j}w_j + \ell_{N-1,j+1}w_{j+1} + \cdots + w_N & = & 0 \end{array}$$

donc le coût de sa résolution est de $(N-j+1)^2$ ops. La complexité du calcul de $w_1, \ldots, w_N$ est donc

$$\sum_{j=1}^{N}(N-j+1)^2 = \frac{N^3}{3} + O(N^2).$$

par contre on ne peut pas faire ce genre d'économie pour le calcul des v_js, donc le calcul de A^{-1} à partir de la décomposition LU es estimé en

$$\frac{N^3}{3} + O(N^2) + N^3 = \frac{4}{3}N^3 + O(N^2).$$

PROPOSITION 1.6. *Soit $A \in \mathbb{F}^{N\times N}$ une matrice fortement inversible, alors le calcul de A^{-1} via l'algorithme d'élimination sans pivotage se fait en $2N^3 + O(N^2)$ ops.*

DÉMONSTRATION. Conséquence de l'antérieur joint au coût $2N^3/3$ de la factorisation LU. □

Donc l'inversion coûte approximativement 3 fois le prix de résoudre un système linéaire. On pourrait penser que cette perte est compensée si l'on doit résoudre plusieurs systèmes avec la même matrice A. Voyons si c'est correcte : supposons qu'on doit résoudre

$$Ax_k = b_k \quad , \quad 1 \leq k \leq K$$

pour $K \gg 0$. Si l'on garde la décomposition LU pour résoudre chacun de ces systèmes, cela nous prendra

$$2N^3/3 + 2KN^2 \quad \text{ops}.$$

Si par contre on calcule d'abord l'inverse A^{-1} puis on calcule $x_k \leftarrow A^{-1}b_k$, ceci nous prendra

$$2N^3 + 2KN^2 + O(N^2) \quad \text{ops},$$

c'est-à-dire $4N^3/3$ ops de plus par rapport au résultat précédent. En conséquent, en général on préférera la décomposition LU au calcul de A^{-1}, pour la résolution de systèmes d'équations linéaires.

2. Élimination avec pivotage

Si dans le cours de l'élimination on rencontre un pivot zéro, l'algorithme se plante. Et si on rencontre un pivot qui n'est pas zéro mais tout petit, l'algorithme ne se plante pas mais le calcul peut être atteint des erreurs d'arrondi considérables, comme on va le voir dans l'exemple suivant.

Soit $\varepsilon \neq 0$ petit, et considérons le système

$$\begin{aligned} \varepsilon x + y &= 1, \\ x + y &= 2. \end{aligned}$$

En applicant l'élimination en exacte on obtient le système équivalent

$$\begin{aligned} \varepsilon x + y &= 1, \\ \left(1 - \frac{1}{\varepsilon}\right) y &= 2 - \frac{2}{\varepsilon} \end{aligned}$$

d'où

$$y = \frac{1-2\epsilon}{1-\varepsilon} \cong 1 \quad , \quad x = \frac{1-y}{\epsilon} = \frac{1}{1-\varepsilon} \cong 1.$$

Supposons maintenant que ε est suffisamment petit par rapport à la précision de la machine ; par exemple prenons

$$\varepsilon = 10^{-4}$$

sur un système de flottants à trois décimaux. Donc

$$1 \ominus \frac{1}{\varepsilon} = -\frac{1}{\varepsilon} \quad , \quad 2 \ominus \frac{1}{\varepsilon} = -\frac{1}{\varepsilon}$$

entraînant

$$y = 1$$

et la substitution donne

$$x = (1 \ominus y) \oslash \varepsilon = 0$$

ce qui est un erreur inacceptable. Essayons en changeant l'ordre des équations :

$$\begin{aligned} x + y &= 2, \\ \varepsilon x + y &= 1. \end{aligned}$$

Le processus d'élimination nous donne cette fois le système

$$\begin{aligned} x + y &= 2, \\ (1-\varepsilon) y &= 1 - 2\varepsilon. \end{aligned}$$

En arithmétique flottante devient

$$y = 1 \quad , \quad x = 2 - y = 1,$$

maintenant l'erreur est tout à fait raisonnable. L'erreur dans le premier essai vient de diviser un nombre par un petit pivot ε, ce qui amplifie les erreurs.

2.1. Pivotage partiel et total.

Définition 2.1. Soit $\sigma \in S_N$ une permutation, la *matrice de permutacions* correspondante P_σ est l'identité avec les lignes permutées suivant σ.

Lemme 2.2. *Soient $\sigma, \tau \in S_N$ des permutations et $P_\sigma, P_\tau \in \mathbb{R}^{N\times N}$ les matrices associées, et $A \in \mathbb{F}^{N\times N}$ une matrice quelconque, alors*

(1) *$P_\sigma \cdot A$ est la matrice A avec les lignes permutées suivant σ ;*
(2) *$P_\sigma^{-1} = P_{\sigma^{-1}} = P_\sigma^*$;*
(3) *$P_\sigma \cdot P_\tau = P_{\sigma\circ\tau}$.*

DÉMONSTRATION. Exercice. □

Le *pivotage partiel* consiste à chaque étape i de l'élimination à choisir le coefficient $a_{k,i}$ de plus grande valeur absolue dans la première colonne du complément de Schur S_i, puis échanger les lignes i et k pour utiliser ce coefficient comme pivot. Ceci consiste à l'introduction d'un pas intermédiaire dans l'algorithme d'élimination juste avantle pas (1) :

(0.5) $a'_i \leftarrow a_{i,i}$, $k \leftarrow i$;
for ℓ **from** i **to** N **do**
if $|a_{\ell,i}| > a'_i$ **then** $a'_i \leftarrow a_{\ell,i}$, $k \leftarrow \ell$;
od ;

Notons que ceci assure que les coefficients de la matrice L sont tous de valeur absolue ≤ 1.

Le *pivotage total* consiste à choisir le coefficient $a_{k,j}$ de plus grande valeur absolue dans *tout* le complément de Schur S_i, pour ensuite ensuite échanger les lignes i et k et la colonne i et j.

THÉORÈME 2.3. *Soit $A \in \mathbb{F}^{N\times N}$ une matrice inversible, alors il existent des (non uniques) matrices $P, L, U \in \mathbb{R}^{N\times N}$, P matrice de permutations, L triangulaire inférieure avec 1s dans la diagonal telle que $|L_{i,j}| \leq 1$ pour tout i, j, et U triangulaire supérieure, telles que*

$$A = P \cdot L \cdot U.$$

DÉMONSTRATION. Soit $\sigma \in S_N$ la permutation obtenue *via* pivotage partiel, alors

$$P_\sigma \cdot A = L \cdot U$$

avec de plus $|\ell_{i,j}| \leq 1$, donc $A = P^*_\sigma \cdot L \cdot U$. □

Similairement le pivotage total corresponds à une factorisation

$$A = P \cdot L \cdot U \cdot Q$$

avec P, Q matrices de permutations.

Le pas (0.5) rajoute $N - i$ comparaisons pour chaque $1 \leq i \leq N - 1$, soit un total de

$$\sum_{i=1}^{N-1} N - i = \frac{N(N-1)}{2} \quad \text{comparaisons.}$$

Le coût total de l'*élimination gaussienne avec pivotage partiel* (EGPP) reste en $2N^3/3 + O(N^2)$. Le pivotage total demande $(N - i)^2$ comparaisons pour chaque $1 \leq i \leq N - 1$, soit

$$\sum_{i=1}^{N-1} (N-i)^2 \cong \frac{N^3}{3} \quad \text{comparaisons.}$$

Le coût total de l'*élimination gaussienne avec pivotage total* (EGPT) monte à $N^3 + O(N^2)$.

EGPP est la façon la plus habituelle d'implémenter l'élimination en pratique. À cause de son coût plus élevé, EGPT n'est presque jamais utilisée, bien qu'il y ait des rares exemples où EGPP tombe à défaut et EGPT réussi à calculer la solution correcte.

3. Complexité de l'élimination en exacte

Considérons l'exemple suivant :

$$A = \begin{bmatrix} 7 & -2 & 1 \\ 1 & 5 & 3 \\ 1 & 1 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{7} & 1 & 0 \\ \frac{1}{7} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 7 & -2 & 1 \\ 0 & 37/7 & 20/7 \\ 0 & 16/7 & 55/7 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 0 \\ 1/7 & 1 & 0 \\ 1/7 & \frac{16/7}{37/7} & 1 \end{bmatrix} \cdot \begin{bmatrix} 7 & -2 & 1 \\ 0 & 37/7 & 20/7 \\ 0 & 0 & (55/7) - (20/7)\frac{16/7}{37/7} \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 0 \\ 1/7 & 1 & 0 \\ 1/7 & 16/37 & 1 \end{bmatrix} \cdot \begin{bmatrix} 7 & -2 & 1 \\ 0 & 37/7 & 20/7 \\ 0 & 0 & 265/37 \end{bmatrix}.$$

À chaque étape, le coût des opérations arithmétiques devient de plus en plus lourd. Ceci est dû à l'augmentation de la taille des calculs intermédiaires, caractéristique des calculs exacts. Une estimation naïve (*via* récurrence) montre qu'on s'attend *a priori* à des calculs intermédiaires de taille binaire 2^{N-1} fois le taille des coefficients de A, ce qui rendrait l'algorithme impraticable puisque de complexité exponentielle. Comme on le voit dans l'exemple, heureusement des simplifications se produisent. En fait la croissance de la taille des calculs intermédiaires est *linéaire* :

PROPOSITION 3.1. *Soit $A \in \mathbb{Z}^{N\times N}$ et posons $h(A) := \max_{1\leq i,j\leq N} h(a_{i,j})$ la hauteur de A, alors*

$$h(L), h(U) \leq N(h(A) + \log(N)).$$

DÉMONSTRATION. Soit $C \in \mathbb{Z}^{N\times}$, à l'aide du développement du déterminant on voit que $h(\det(C)) \leq N(h(C) + \log(N))$. La proposition est conséquence de l'exercice ci-dessous. □

EXERCICE 4.2. ◁ Soit $A \in \mathbb{F}^{N\times N}$ une matrice admettant une décomposition LU : (3)

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \ell_{2,1} & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ \ell_{N,1} & \ell_{N,2} & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,N} \\ 0 & s_{2,2} & \cdots & s_{2,N} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & s_{N,N} \end{bmatrix}.$$

Pour $1 \leq i_1 < \cdots < i_p \leq N$ et $1 \leq j_1 < \cdots < j_p \leq N$ soit

$$A(i_1, \ldots, i_p; j_1, \ldots, j_p) := [a_{i_k, j_\ell}]_{1\leq k,\ell\leq p} \ \in \mathbb{F}^{p\times p}$$

la sous-matrice de A associée aux lignes $i_1, \ldots, i_p$ et aux colonnes $j_1, \ldots, j_p$. Montrer que

$$\ell_{i,j} = \frac{\det(A(1, 2, \ldots, j-1, i; 1, 2, \ldots, j-1, j))}{\det(A(1, 2, \ldots, j-1, j; 1, 2, \ldots, j-1, j))} \qquad (i > j)$$

et

$$s_{i,j} = \frac{\det(A(1, 2, \ldots, i-1, i; 1, 2, \ldots, i-1, j))}{\det(A(1, 2, \ldots, i-2, i-1; 1, 2, \ldots, i-2, i-1))} \qquad (i \leq j).$$

Indication : Effacez de façon convenable des lignes et des colonnes dans la factorisation (3). ▷

PROPOSITION 3.2. *Soit $A \in \mathbb{Z}^{N\times N}$ une matrice inversible, alors la factorisation PLU se fait en $O^*(N^4 h(A))$ opérations binaires.*

Rappelons que la notation O^* signifie "à des facteurs logarithmiques près".

DÉMONSTRATION. Par rapport à l'algorithme EGPP, il faut juste remplacer les opérations flottants par les opérations de $\mathbb{Q}$ avec des expressions réduites. Le corollaire 1.3 du chapitre 3 entraîne

$$\mathcal{C}_{EGPP}(\tau) \leq O^*(N^3 \cdot h_{EGPP}(\tau))$$

avec $\tau := h(A)$. Le résultat est donc conséquence de la proposition 3.1. □

4. Conditionnement d'un opérateur

Les sorties des algorithmes de l'analyse numérique sont rarement exactes. Les sources d'erreurs possibles sont

(1) erreurs dans l'entrée dues aux erreurs de méditions et d'arrondi ;
(2) propagation d'erreurs d'arrondi au cours de l'exécution de l'algorithme ;
(3) seulement pour les méthodes itératives : erreur d'approximation.

Dans cette section on traitera la théorie de perturbations qui nous permettra traiter les erreurs de type (1). Soient

$$\widehat{A} = A + \delta A \in \mathbb{F}^{N\times N} \quad , \quad \widehat{b} = b + \delta b \in \mathbb{F}^N$$

des perturbations d'une matrice inversible A et d'un vecteur b. Quel est l'erreur $\delta x = \widehat{x} - x$ de la résolution *exacte* du système $\widehat{A}\widehat{x} = \widehat{b}$? On a

$$\begin{array}{rrcl} & (A+\delta A)(x+\delta x) & = & b+\delta b \\ - & Ax & = & b \\ \hline & \delta Ax + (A+\delta A)\delta x & = & \delta b. \end{array}$$

On en obtient $\delta x = A^{-1}(-\delta A\widehat{x} + \delta b)$ et donc pour une norme vectorielle $|\cdot|$ et la norme d'opérateurs subordonnée $||\cdot||$

$$|\delta x| \leq ||A^{-1}||(||\delta A|| \cdot |\widehat{x}| + ||\delta b||).$$

DÉFINITION 4.1. La quantité

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

est le *conditionnement* de la matrice A relatif à la norme vectorielle $|\cdot|$. Pour A non inversible on pose $\kappa(A) := \infty$.

Avec cette définition

$$\frac{|\delta x|}{|\widehat{x}|} \leq \kappa(A)\left(\frac{||\delta A||}{||A||} + \frac{|\delta b|}{||A|| \cdot |\widehat{x}|}\right). \tag{4}$$

Le conditionnement de A mesure l'erreur relatif $|\delta x|/|\widehat{x}|$ en termes de l'erreur relatif $||\delta A||/||A||$. La majoration ci-dessus dépends de δx (*via* $\widehat{x}$) et donc paraît difficile à interpréter. Cependant elle est utile en pratique parce que on connaît la quantité calculée $\widehat{x}$ et on peut évaluer la majoration directement. Alternativement, on peut déduire une borne plus attractive du point de vue théorique :

$$\begin{aligned} \frac{|\delta x|}{|x|} &\leq \kappa(A)\left(\frac{||\delta A||}{||A||}\Big(1 + \frac{|\delta x|}{|x|}\Big) + \frac{|\delta b|}{||A|| \cdot ||x||}\right) \\ &\leq \kappa(A)\left(\frac{||\delta A||}{||A||}\Big(1 + \frac{|\delta x|}{|x|}\Big) + \frac{|\delta b|}{|b|}\right) \end{aligned}$$

donc

$$\left(1 - \kappa(A)\frac{||\delta A||}{||A||}\right)\frac{|\delta x|}{|x|} \leq \kappa(A)\left(\frac{||\delta A||}{||A||} + \frac{|\delta b|}{|b|}\right)$$

c'est-à-dire

$$\frac{|\delta x|}{|x|} \leq \frac{\kappa(A)}{1-\kappa(A)\dfrac{||\delta A||}{||A||}}\left(\frac{||\delta A||}{||A||}+\frac{|\delta b|}{|b|}\right). \tag{5}$$

On a

$$\frac{\kappa(A)}{1-\kappa(A)\dfrac{||\delta A||}{||A||}} \cong \kappa(A) \qquad \text{pour } \frac{||\delta A||}{||A||} \to 0.$$

Par exemple, pensons au cas où $\widehat{A} = \mathrm{fl}(A)$ et $\widehat{b} = \mathrm{fl}(b)$ sont produits par la troncation à t bits correctes d'une matrice et d'un vecteur réels. Soit β la base de l'arithmétique flottante en question, alors

$$\frac{|x - \mathrm{fl}(x)|}{|x|} \leq \mu := \beta^{1-t}/2$$

pour tout $x \neq 0$. Donc si l'on pose $\delta b = b - \mathrm{fl}(b)$ on a $|(\delta b)_i| \leq \mu |b_i|$ pour tout i donc

$$|\delta b|_\infty = \max_i |(\delta b)_i| \leq \mu \max_i |b_i| = \mu |b|_\infty.$$

Similairement si l'on pose $\delta A = A - \mathrm{fl}(A)$ on a

$$||\delta A||_\infty = \max_i \sum_{j=1}^N |(\delta A)_{i,j}| \leq \mu \cdot \max_i \sum_{j=1}^N |A_{i,j}| = \mu \cdot ||A||_\infty$$

soit

$$\frac{|\delta b|_\infty}{|b|_\infty}, \frac{||\delta A||_\infty}{||A||_\infty} \leq \mu. \tag{6}$$

La majoration (5) entraîne

$$\frac{|\delta x|_\infty}{|x|_\infty} \leq 2\mu \cdot \frac{\kappa_\infty(A)}{1-\kappa_\infty(A)\mu}$$

et si $\kappa_\infty(A)\mu \leq 1/2$ (ou de façon équivalente $\kappa_\infty(A) \leq \beta^{t-1}$) on a

$$\frac{|\delta x|_\infty}{|x|_\infty} \leq 4\mu\kappa_\infty(A).$$

Autrement-dît, la *perte de précision* due au troncations dans l'entrée est de (pour la base $\beta = 2$)

$$\log_2(\kappa_\infty(A)) + 2 \quad \text{bits.}$$

Une façon alternative d'estimer l'erreur est *via* le *résidu* de $\widehat{x}$:

$$r = A\widehat{x} - b \in \mathbb{F}^N.$$

On a $\delta x = A^{-1}r$ donc

$$||\delta x|| \leq ||A^{-1}||||r||.$$

On a $\kappa(A) \geq 1$ pour toute matrice A. Le conditionnement est invariant par multiplication par scalaires : $\kappa(\lambda \cdot A) = \kappa(A)$. Si $\kappa(A)$ n'est pas trop grand par rapport à la dimension, on dit que la matrice est *bien conditionnée*, autrement on dit qu'elle est *mal conditionnée*.

Soient $\sigma_1 \geq \cdots \geq \sigma_N > 0$ les valeurs singulières d'une matrice inversible A, alors $||A||_2 = \sigma_1$ et $||A^{-1}||_2 = \sigma_N^{-1}$ donc pour la norme 2

$$\kappa_2(A) = \sigma_1/\sigma_N$$

est l'élongation de l'ellipsoïde $\{Ax : |x|_2 = 1\} \subset \mathbb{F}^N$. En particulier, le conditionnement d'une matrice unitaire U est parfait : $\kappa_2(U) = 1$.

Le conditionnement admets une caractérisation géométrique comme l'inverse de la distance à l'ensemble des matrices singulières :

PROPOSITION 4.2. *Soit A une matrice inversible, alors*

$$\frac{1}{\kappa(A)} = \min\left\{ \frac{||\delta A||_2}{||A||_2} : A + \delta A \text{ est singulière} \right\}.$$

DÉMONSTRATION. C'est une reformulation du théorème d'approximation de Schmid : dans la notation du théorème 2.3 du chapitre 1

$$\mathrm{dist}_{||\cdot||_2}(A, M_{N-1,N-1}) = \sigma_N = ||A^{-1}||^{-1}.$$

□

Les conditionnements associés aux différentes métriques sont comparables, or les constantes dépendent de la dimension :

$$\frac{1}{N}\kappa_2(A) \leq \kappa_1(A) \leq N\kappa_2(A);$$
$$\frac{1}{N}\kappa_\infty(A) \leq \kappa_2(A) \leq N\kappa_\infty(A);$$
$$\frac{1}{N^2}\kappa_2(A) \leq \kappa_\infty(A) \leq N^2\kappa_1(A).$$

Vue l'importance du nombre de conditionnement, on peut se demander quel est la probabilité qu'une matrice soit bien (ou mal) conditionnée. Concrètement, soit

$$N = 10^6 = (10^2)^3$$

notre taille de matrices préférée, Quel est le volume relatif dans la sphère unité (par rapport à la métrique de Frobenius) de $\mathbb{R}^{N\times N}$ de l'ensemble des matrices A telles que

$$\kappa_2(A) \leq 10^4?$$

La réponse est $\ldots$ 10^{17776} ! Donc la probabilité de tomber la-dessus est pratiquement 0. Fort heureusement, le conditionnement n'est pas trop grand pour beaucoup d'applications intéressantes : pour une matrice A issue de la discrétisation d'une EDP elliptique, typiquement on a

$$\kappa_2(A) = O(h^{-2})$$

où h est le pas de discrétisation.

5. Analyse formelle de l'erreur dans l'algorithme d'élimination

5.1. Stabilité du produit scalaire. Commençons notre étude d'erreurs d'arrondi (c'est-à-dire les erreurs de type (2) dans la liste dans la section 4) en considérant les erreurs d'arrondi de l'algorithme standard pour le produit scalaire :

Entrée : $x, y \in \mathbb{R}^N$ **;**
Sortie : $\langle x, y\rangle$**.**
$s_0 \leftarrow 0$ **;**
For $k = 1$ **to** N **do**
$s_k \leftarrow s_{k-1} + x_k \cdot y_k$ **;**
od ; end.

En essayant de quantifier les erreurs d'arrondi, on est tout de suite confronté avec un problème notationnel : de distinguer les quantités calculées des quantités exactes. Quand le contexte est clair, on utilisera la notation $\mathrm{fl}(\cdot)$ pour noter les quantités calculées. Ainsi $\mathrm{fl}(\langle x, y\rangle)$ désigne la sortie de l'algorithme ci-dessus. Aussi on notera

$$\mathrm{abs}(x) := (|x_1|, \ldots, |x_N|) \in \mathbb{R}^N_+$$

et pour $x, y \in \mathbb{R}^N$ on dit $x \leq y$ si $x_i \leq y_i$ pour $1 \leq i \leq N$. Similairement pour une matrice A on pose

$$\mathrm{abs}(A) := [|A_{i,j}|]_{1 \leq i,j \leq N} \in \mathbb{R}_+^{N \times N}.$$

Soit

$$\mu := \beta^{1-t}/2$$

la précision du système flottant choisi.

PROPOSITION 5.1.

$$|\,\mathrm{fl}(\langle x, y\rangle) - \langle x, y\rangle| \leq N\langle \mathrm{abs}(x), \mathrm{abs}(y)\rangle\mu + O(\mu^2).$$

DÉMONSTRATION. Soit s_k le k-ème calcul intermédiaire dans l'algorithme, alors

$$s_1 = x_1 \odot y_1 = x_1 \cdot y_1(1 + \delta_1)$$

avec $|\delta_1| \leq \mu$. Puis

$$\begin{aligned} s_2 &= s_1 \oplus (x_2 \odot y_2) \\ &= (s_1 + (x_2 \odot y_2))(1 + \varepsilon_2) \\ &= (x_1 \cdot y_1(1 + \delta_1) + x_2 \cdot y_2(1 + \delta_2))(1 + \varepsilon_2) \end{aligned}$$

avec $|\delta_2|, |\varepsilon_2| \leq \mu$. Similairement

$$s_N = \mathrm{fl}(\langle x, y\rangle) = \sum_{j=1}^{N} x_j \cdot y_j(1 + \gamma_j)$$

avec

$$1 + \gamma_j = (1 + \delta_j) \prod_{\ell=j}^{N} (1 + \varepsilon_\ell) \qquad \text{(convention : } \varepsilon_1 = 0\text{)}$$

avec $|\delta_j|, |\varepsilon_\ell| \leq \mu$. On vérifie

$$\gamma_j \leq N\mu + O(\mu^2)$$

et ainsi

$$|\,\mathrm{fl}(\langle x, y\rangle) - \langle x, y\rangle| \leq \sum_{j=1}^{N} |x_j| \cdot |y_j| \cdot |\gamma_j| \leq N\langle \mathrm{abs}(x), \mathrm{abs}(y)\rangle\mu + O(\mu^2).$$

□

Si $\langle x, y\rangle \ll \langle \mathrm{abs}(x), \mathrm{abs}(y)\rangle$, l'algorithme est passible de produire d'erreurs considérables. Pensez à l'exemple du calcul de l'exponentielle $\exp(x)$ pour $x < 0$ en sommant N termes de sa série de Taylor : ceci est le produit scalaire des vecteurs

$$X := (1, x, x^2, \ldots, x^{N-1}) \quad , \quad Y := \Big(\frac{1}{0!}, \frac{1}{1!}, \frac{1}{2}, \ldots, \frac{1}{(N-1)!}\Big);$$

dans ce cas

$$\langle X, Y\rangle \cong \exp(x) \ll \langle \mathrm{abs}(X), \mathrm{abs}(Y)\rangle \cong \exp(-x).$$

5.2. Analyse d'erreur *a priori* **et** *a posteriori*. L'analyse du calcul du produit scalaire est un exemple d'analyse d'erreur *a priori*, où l'on estime l'erreur à partir des données.

Le paradigme d'analyse d'erreurs de troncation de la résolution d'un système $Ax = b$ est l'analyse *a posteriori*. Ceci consiste à montrer que le résultat $\widehat{x}$ calculé inexactement par l'algorithme, est le résultat exacte d'une perturbation

$$\widehat{A}\widehat{x} = \widehat{b}$$

avec $\delta A = \widehat{A} - A$ et $\delta b = \widehat{b} - b$ petits. Ceci permets d'unifier l'analyse d'erreurs de type (1) et de type (2), et de les controler par la théorie de perturbations introduite dans la section 4.

Exemple : multiplication en flottant de deux matrices 2×2 :

$$\begin{aligned}\mathrm{fl}(A \cdot B) &= \begin{bmatrix} a_{1,1} & a_{1,2} \\ & a_{2,2} \end{bmatrix} \odot \begin{bmatrix} b_{1,1} & b_{1,2} \\ & b_{2,2} \end{bmatrix} \\ &= \begin{bmatrix} a_{1,1}b_{1,1}(1+\varepsilon_1) & \Big((a_{1,1}b_{1,2}(1+\varepsilon_2) + a_{1,2}b_{2,2}(1+\varepsilon_3)\Big)(1+\varepsilon_4) \\ & a_{2,2}b_{2,2}(1+\varepsilon_5) \end{bmatrix}\end{aligned}$$

avec $|\varepsilon_i| \leq \mu$. Écrivons

$$\widehat{A} := \begin{bmatrix} a_{1,1} & a_{1,2}(1+\varepsilon_3)(1+\varepsilon_4) \\ & a_{2,2}(1+\varepsilon_5) \end{bmatrix} \quad , \quad \widehat{B} := \begin{bmatrix} b_{1,1}(1+\varepsilon_1) & b_{1,2}(1+\varepsilon_2)(1+\varepsilon_4) \\ & b_{2,2} \end{bmatrix},$$

alors

$$\mathrm{abs}(\widehat{A} - A) \leq 2\,\mathrm{abs}(A)\mu + O(\mu^2) \quad , \quad \mathrm{abs}(\widehat{B} - B) \leq 2\,\mathrm{abs}(B)\mu + O(\mu^2).$$

et

$$A \odot B = \widehat{A} \cdot \widehat{B}.$$

5.3. Stabilité de systèmes triangulaires. Considérons un système triangulaire inversible

$$\begin{aligned} u_{1,1}x_1 + u_{1,2}x_2 + \cdots + u_{1,N}x_N &= y_1, \\ u_{2,2}x_2 + \cdots + u_{1,N}x_N &= y_2, \\ &\vdots \\ u_{N,N}x_N &= y_N. \end{aligned}$$

Sa résolution se fait par substitution successive :

For $i = N$ **to** 1 **do**
$x_i \leftarrow \frac{1}{u_{i,i}}(y_i - u_{i,i+1}y_{i+1} - \cdots - u_{i,N}y_N)$ **;**
od ; end.

Analysons la stabilité de cet algorithme :

PROPOSITION 5.2. *Soit $\widehat{x}$ la solution calculée de $Ux = y$, alors*

$$(U + G)\widehat{x} = y$$

avec $\mathrm{abs}(G) \leq N\,\mathrm{abs}(U)\mu + O(\mu^2)$.

Similairement, pour $L \in \mathbb{R}^{N\times N}$ matrice triangulaire inférieure avec 1s dans la diagonal, la solution calculée $\widehat{y}$ de $Ly = b$ vérifie

$$(L + F)\widehat{y} = b$$

pour une matrice $F \in \mathbb{R}^{N\times N}$ telle que $\mathrm{abs}(F) \leq N\,\mathrm{abs}(L)\mu + O(\mu^2)$.

DÉMONSTRATION. On a

$$\widehat{x}_N = y_N \oslash u_{N,N} = \frac{y_N}{u_{N,N}}(1+\gamma_N) = \frac{y_N}{u_{N,N}(1+\gamma_N)^{-1}}$$

avec $|\gamma_N| \leq \mu$ donc on fait

$$G_{N,N} := u_{N,N}(1-(1+\gamma_N)^{-1}) = u_{N,N}\mu + O(\mu^2).$$

Puis

$$\begin{aligned}\widehat{x}_{N-1} &= (y_{N-1} \ominus u_{N-1,N} \odot \widehat{x}_N) \oslash u_{N-1,N-1} \\ &= \frac{1}{u_{N-1,N-1}}\left(y_{N-1} - u_{N-1,N}(1+\eta_{N-1})\widehat{x}_N\right)(1+\gamma_{N-1})(1+\delta_{N-1})\end{aligned}$$

avec $|\gamma_{N-1}|, |\eta_{N-1}|, |\delta_{N-1}| \leq \mu$, donc on fait

$$\begin{aligned}G_{N-1,N-1} &:= u_{N-1,N-1}\eta_{N-1} = u_{N-1,N-1}\mu + O(\mu^2), \\ G_{N-1,N} &:= u_{N-1,N}(1-(1+\gamma_{N-1})^{-1}(1+\delta_{N-1})^{-1}) = 2u_{N-1,N}\mu + O(\mu^2).\end{aligned}$$

Les autres coefficients de G se définent et estiment similairement. □

5.4. Stabilité de la décomposition LU. L'intuition derrière l'analyse d'erreur dans la décomposition LU est que si les quantités dans le produit $\widehat{L}\cdot\widehat{U}$ des matrices triangulaires calculées sont grandes en comparaison avec A, l'information dans A sera essentiellement perdue dans la résolution. Donc on se mets à estimer l'erreur dans le calcul des facteurs triangulaires calculés :

THÉORÈME 5.3. *Soit $A \in \mathbb{R}^{N\times N}$ telle qu'aucun des pivots successivement calculés par l'algorithme d'élimination soit nul, alors les matrices calculées $\widehat{L}, \widehat{U}$ vérifient*

$$\widehat{L}\cdot\widehat{U} = A + H$$

avec $\operatorname{abs}(H) \leq 2N(\operatorname{abs}(\widehat{L})\cdot\operatorname{abs}(\widehat{U}) + \operatorname{abs}(A))\mu + O(\mu^2)$.

DÉMONSTRATION. La preuve est par récurrence sur N. Le résultat est trivial pour $N = 1$. Maintenant supposons-le vraie pour les matrices de taille $(N-1)\times(N-1)$ et écrivons

$$A = \begin{bmatrix}\pi_1 & \ell_1 \\ c_1 & A_{1,1}\end{bmatrix}.$$

Alors

$$\widehat{p} = c_1 \oslash \pi_1 \in \mathbb{R}^N \quad , \quad \widehat{S}_1 = A_{1,1} \ominus \widehat{p}\odot\ell \in \mathbb{R}^{N\times N}$$

sont calculés dans le premier pas de l'algorithme. Par récurrence $\widehat{L}_1\cdot\widehat{U}_1 = \widehat{S}_1 + H_1$ avec

$$\operatorname{abs}(H_1) \leq 2(N-1)(\operatorname{abs}(\widehat{L}_1)\cdot\operatorname{abs}(\widehat{U}_1) + \operatorname{abs}(\widehat{S}_1))\mu + O(\mu^2),$$

alors

$$\begin{aligned}H := \widehat{L}\cdot\widehat{U} - A &= \begin{bmatrix}1 & \\ \widehat{p} & \widehat{L}_1\end{bmatrix}\cdot\begin{bmatrix}\pi_1 & \ell_1 \\ & \widehat{U}_1\end{bmatrix} - A \\ &= \begin{bmatrix}\pi_1 & \ell_1 \\ \widehat{p}\cdot\pi_1 & \widehat{L}_1\cdot\widehat{U}_1 + \widehat{p}\cdot\ell_1\end{bmatrix} - \begin{bmatrix}\pi_1 & \ell_1 \\ c_1 & A_{1,1}\end{bmatrix} \\ &= \begin{bmatrix}0 & 0 \\ \widehat{p}\cdot\pi_1 - c_1 & \widehat{L}_1\cdot\widehat{U}_1 - (A_{1,1} - \widehat{p}\cdot\ell_1)\end{bmatrix}.\end{aligned}$$

On a

$$\widehat{p} = c_1/\pi_1 + f$$

avec $\operatorname{abs}(f) \leq \operatorname{abs}(c_1/\pi_1)\mu$ donc

$$\operatorname{abs}(\widehat{p}\cdot\pi_1 - c_1) \leq \operatorname{abs}(c_1)\mu.$$

On a

$$\operatorname{abs}(\widehat{S}_1 - (A_{1,1} - \widehat{p}\cdot\ell_1)) \le 2\mu(\operatorname{abs}(A_{1,1}) - \operatorname{abs}(\widehat{p})\cdot\operatorname{abs}(\ell_1)) + O(\mu^2)$$

et donc

$$\begin{aligned}\operatorname{abs}(\widehat{L}_1\cdot\widehat{U}_1 - (A_{1,1} - \widehat{p}\cdot\ell_1)) &= \operatorname{abs}(H_1 + \widehat{S}_1 - (A_{1,1} - \widehat{p}\cdot\ell_1))\\ &\le 2(N-1)(\operatorname{abs}(\widehat{L}_1)\cdot\operatorname{abs}(\widehat{U}_1) + \operatorname{abs}(\widehat{S}_1))\mu\\ &\quad + 2\mu(\operatorname{abs}(A_{1,1}) - \operatorname{abs}(\widehat{p})\cdot\operatorname{abs}(\ell_1)) + O(\mu^2)\\ &\le 2N(\operatorname{abs}(\widehat{L})\cdot\operatorname{abs}(\widehat{U}) + \operatorname{abs}(A))\mu + O(\mu^2).\end{aligned}$$

□

5.5. Stabilité de la résolution. On analyse l'effet des arrondis quand $\widehat{L}$ et $\widehat{U}$ sont utilisées pour résoudre le système $Ax = b$:

THÉORÈME 5.4. *Soient $\widehat{L}$ et $\widehat{U}$ les facteurs de A calculées par l'algorithme d'élimination, et soit $\widehat{x}$ la solution calculée de*

$$\widehat{L}y = b \quad , \quad \widehat{U}x = y,$$

alors $(A+E)\widehat{x} = b$ avec

$$\operatorname{abs}(E) \le N(4\operatorname{abs}(\widehat{L})\cdot\operatorname{abs}(\widehat{U}) + 2\operatorname{abs}(A))\mu + O(\mu^2).$$

DÉMONSTRATION. Par la proposition 5.2 on a

$$\begin{aligned}(\widehat{L}+F)\widehat{y} = b \qquad &\text{avec } \operatorname{abs}(F) \le N\operatorname{abs}(\widehat{L})\mu + O(\mu^2);\\ (\widehat{U}+G)\widehat{x} = \widehat{y} \qquad &\text{avec } \operatorname{abs}(G) \le N\operatorname{abs}(\widehat{U})\mu + O(\mu^2)\end{aligned}$$

donc

$$(\widehat{L}+F)(\widehat{U}+G)\widehat{x} = (\widehat{L}\widehat{U} + \widehat{L}G + F\widehat{U} + FG)\widehat{x} = b.$$

Par le théorème 5.3 on a $\widehat{L}\cdot\widehat{U} = A + H$ avec

$$\operatorname{abs}(H) \le 2N(\operatorname{abs}(\widehat{L})\cdot\operatorname{abs}(\widehat{U}) + \operatorname{abs}(A))\mu + O(\mu^2)$$

et alors en écrivant

$$E := H + F\widehat{U} + \widehat{L}G + FG$$

on a $(A+E)\widehat{x} = b$ et

$$\begin{aligned}\operatorname{abs}(E) &\le \operatorname{abs}(H) + \operatorname{abs}(F)\operatorname{abs}(\widehat{U}) + \operatorname{abs}(\widehat{L})\operatorname{abs}(G) + O(\mu^2)\\ &\le N(4\operatorname{abs}(\widehat{L})\cdot\operatorname{abs}(\widehat{U}) + 2\operatorname{abs}(A))\mu + O(\mu^2).\end{aligned}$$

□

Il est tout à fait possible que le terme $\operatorname{abs}(\widehat{L})\cdot\operatorname{abs}(\widehat{U})$ soit grand. Il n'y a rien qu'empêche de rencontrer l'apparition des petits pivots même si la matrice A est bien conditionnée, comme le montre l'exemple

$$A = \begin{bmatrix}\varepsilon & 1\\ 1 & \end{bmatrix}$$

Donc l'algorithme d'élimination est instable. Cette désavantage est réparée par l'application du *pivotage*. Examinons la stabilité de EGPP, l'algorithme d'élimination avec pivotage partiel par lignes. La solution calculée $\widehat{x}$ satisfait $(A+E)\widehat{x} = b$ avec

$$\operatorname{abs}(E) \le N(4P\operatorname{abs}(\widehat{L})\cdot\operatorname{abs}(\widehat{U}) + 2\operatorname{abs}(A))\mu + O(\mu^2)$$

où P est la matrice de permutations produite par le pivotage partiel. Notons

$$|A|_{\max} := \max_{i,j}|A_{i,j}|$$

la norme ∞ de A vue comme un vecteur. Le pivotage partiel entraîne alors

$$|L|_{\max} \leq 1$$

et donc

$$|E|_{\max} \leq N\mu(2|A|_{\max} + 4N|U|_{\max}) + O(\mu^2) \leq N\mu|A|_{\max}(2 + 4N\rho_{EGPP}(A))$$

où

$$\rho_{EGPP}(A) := |U|_{\max}/|A|_{\max}$$

est le *facteur de croissance des pivots* pour EGPP. En utilisant théorie des perturbations on obtient

$$\begin{aligned}\frac{\delta x}{\widehat{x}} &\leq \kappa_\infty(A)\frac{||\delta A||_\infty}{||A||_\infty} \\ &\leq \kappa_\infty(A)N\frac{|E|_{\max}}{||A||_\infty} \\ &\leq \kappa_\infty(A)N^2(2 + 4N\rho_{EGPP}(A))\mu\end{aligned}$$

La stabilité de EGPP est alors équivalente au fait que $\rho_{EGPP}(A)$ soit petit devant la dimension. En pratique $\rho_{EGPP}(A)$ est presque toujours $\leq N$; malheureusement il y des exemples où il est égal à 2^{N-1}.

PROPOSITION 5.5. *Soit $A \in \mathbb{R}^{N\times N}$, alors*

$$\rho_{EGPP}(A) \leq 2^{N-1}.$$

Cette majoration est optimale.

DÉMONSTRATION. Dans le i-ème pas de l'élimination on fait

$$\widetilde{a}_{j,k} \leftarrow a_{j,k} - \ell_{j,i}a_{i,k} \qquad \text{pour } i+1 \leq j,k \leq N.$$

On a $|\ell_{j,i}| \leq 1$ et donc

$$|S_i|_{\max} \leq 2|S_{i-1}|_{\max};$$

par récurrence

$$|U|_{\max} \leq \max_i |S_i|_{\max} \leq 2^{N-1}|A|_{\max}.$$

L'optimalité sera démontrée en exercice. □

Wilkinson [**11**] a démontré que le facteur de croissance des pivots quand on fait un pivotage *total* est majorée par

$$\rho_{EGPT}(A) \leq N^{1/2}(2\cdot 3^{1/2}\cdots N^{1/(N-1)})^{1/2} \cong N^{1/2+\log(N/4)}.$$

Cette estimation est beaucoup trop grande par rapport à ce qu'on trouve en pratique. C'était une vielle conjecture de démontrer que $\rho_{EGPT}(A) \leq N$, mais elle fut récemment réfutée [**3, 7**]. C'est toujours un problème ouvert de trouver une bonne estimation pour $\rho_{EGPT}(A)$, qu'on croît toujours de l'ordre de $O(N)$.

5.6. Matrices bande. Dans un nombre important d'applications, la matrice A est bande. Typiquement c'est le cas quand A est la discrétisation d'une équation différentielle ordinaire ; dans ce cas on peut ordonner les variables de façon à ce que chaque variable x_i n'apparaît que dans quelques peu équations au voisinage de la i-ème ligne. Comme exemple considérons l' équation de Poisson en dimension 1 avec conditions de Dirichlet aux bords nulles :

$$-u_{xx} = f \quad \text{ pour } x \in \Omega = [0,1] \text{ et } \quad u(0) = u(1) = 0.$$

On discrétise cette équation avec des différences finies. Pour $N \in \mathbb{N}$ on pose

$$h := 1/(N+1)$$

et on considère la grille

$$\Omega_h := \{jh : j = 1, \ldots, N\}$$

des nodes *à l'intérieur* de l'intervalle $[0,1]$ divisé en $N+1$ sous-intervalles, et on pose $\mathcal{G}(\Omega_h) : \Omega_h \to \mathbb{R}$ pour l'ensemble des fonctions réelles de cet ensemble. L'opérateur discret qui en résulte est

$$L_h : \mathcal{G}(\Omega_h) \to \mathcal{G}(\Omega_h) \quad , \quad L_h(u)(x) = h^2\Big(u(x-h) - 2u(x) + u(x+h)\Big)$$

avec la convention $u(0) = u(1) = 0$. Ceci est une approximation d'ordre 2 de l'opérateur laplacien

$$-u_{xx} - L_h(u) = O(h^2)$$

pour $h \to 0$ et des fonctions u suffisamment régulières (par exemple $u \in \mathcal{C}^4(\Omega)$). L'équation approchée $L_h u_h = f_h$ se traduit dans un système linéaire $N \times N$ *bande* :

$$h^2 \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \begin{bmatrix} u(h) \\ u(2h) \\ \vdots \\ u((N-1)h) \\ u(Nh) \end{bmatrix} = \begin{bmatrix} f(h) \\ f(2h) \\ \vdots \\ f((N-1)h) \\ f(Nh) \end{bmatrix}.$$

Formellement, on dit qu'une matrice $A = [a_{i,j}]_{1 \le i,j \le N}$ possède *largeur de bande supérieure* q si $a_{i,j} = 0$ pour $j > i + q$, et *largeur de bande inférieure* p si $a_{i,j} = 0$ pour $i > j + p$:

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,q+1} & & & \\ \vdots & & & a_{2,q+2} & & \\ a_{p+1,1} & & & & \ddots & \\ & a_{p+2,2} & & & & a_{N-q,N} \\ & & \ddots & & & \vdots \\ & & & a_{N,N-p} & \cdots & a_{N,N} \end{bmatrix}$$

La décomposition LU sans pivotage respecte la structure bande :

THÉORÈME 5.6. *Soit $A \in \mathbb{F}^{N \times N}$ une matrice fortement inversible avec largeur de bande supérieure q et inférieure p, alors $A = L \cdot U$ avec L triangulaire inférieure avec largeur de bande p et U triangulaire supérieure avec largeur de bande q.*

La version "bande" de l'algorithme d'élimination sans pivotage calcule L et U en $2pqN$ ops. En particulier, on peut résoudre la discrétisation de l'équation de Poisson en dimension 1 en $8N$ ops.

L'élimination avec pivotage partiel peut être organisée de façon à profiter de la structure bande. Cependant le pivotage change la largeur de bande et la version "bande" de EGPP n'est pas si nette que la version sans pivotage :

THÉORÈME 5.7. *Soit $A \in \mathbb{F}^{N \times N}$ une matrice inversible avec largeur de bande supérieure q et inférieure p, alors $A = L \cdot U$ avec U triangulaire supérieure avec largeur de bande $p+q$ et L est sparse, avec au plus $p+1$ coefficients $\neq 0$ par colonne.*

On renvoie le lecteur à [**6**, § 4.3] pour une preuve de ces résultats ainsi que pour plus de détail sur les matrices bande.

Bibliographie

[1] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1997. With the collaboration of Thomas Lickteig.

[2] James W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[3] Alan Edelman. The complete pivoting conjecture for gaussian elimination. *The Mathematica Journal*, 2(2) :58–61, 1992.

[4] Carl Friedrich Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*. Perthes and Besser, Hamburg, Germany, 1809.

[5] Carl Friedrich Gauss. *Theoria combinationis observationum erroribus minimis obnoxiae/Theory of the combination of observations least subject to errors*, volume 11 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. Pars prior. Pars posterior. Supplementum/Part One. Part Two. Supplement, Dual Latin-English text, Translated and with an introduction and afterword by G. W. Stewart.

[6] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[7] Nick Gould. On growth in Gaussian elimination with complete pivoting. *SIAM J. Matrix Anal. Appl.*, 12(2) :354–361, 1991.

[8] Igor Kaporin. A practical algorithm for faster matrix multiplication. *Numer. Linear Algebra Appl.*, 6(8) :687–700, 1999.

[9] Michelle Schatzman. *Analyse numérique*. InterEditions, Paris, 1991. Cours et exercices pour la licence. [Course and exercises for the bachelor's degree].

[10] Volker Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13 :354–356, 1969.

[11] J. H. Wilkinson. *The algebraic eigenvalue problem*. Clarendon Press, Oxford, 1965.