

1.3 Les méthodes directes

1.3.1 Définition

Définition 1.10 (Méthode directe). On appelle *méthode directe de résolution de (1.1)* une méthode qui donne exactement x (A et b étant connus) solution de (1.1) après un nombre fini d'opérations élémentaires ($+$, $-$, \times , $/$).

Parmi les méthodes de résolution du système (1.1), la plus connue est la *méthode de Gauss* (avec pivot), encore appelée *méthode d'échelonnement* ou *méthode LU* dans sa forme matricielle.

Nous rappelons la méthode de Gauss et sa réécriture matricielle qui donne la méthode *LU* et nous étudierons plus en détails la méthode de Choleski, qui est adaptée aux matrices symétriques.

1.3.2 Méthode de Gauss, méthode LU

Soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice inversible, et $b \in \mathbb{R}^n$. On cherche à calculer $x \in \mathbb{R}^n$ tel que $Ax = b$. Le principe de la méthode de Gauss est de se ramener, par des opérations simples (combinaisons linéaires), à un système triangulaire équivalent, qui sera donc facile à inverser.

Commençons par un exemple pour une matrice 3×3 . Nous donnerons ensuite la méthode pour une matrice $n \times n$.

Un exemple 3×3

On considère le système $Ax = b$, avec

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & -1 \\ -1 & 1 & -2 \end{bmatrix} \quad b = \begin{bmatrix} 2 \\ 1 \\ -2 \end{bmatrix}.$$

On écrit la **matrice augmentée**, constituée de la matrice A et du second membre b .

$$\tilde{A} = [A \quad b] = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ -1 & 1 & -2 & -2 \end{bmatrix}.$$

Gauss et opérations matricielles Allons y pour Gauss :

La première ligne a un 1 en première position (en gras dans la matrice), ce coefficient est non nul, et c'est un **pivot**. On va pouvoir diviser toute la première ligne par ce nombre pour en soustraire un multiple à toutes les lignes d'après, dans le but de faire apparaître des 0 dans tout le bas de la colonne.

La deuxième équation a déjà un 0 dessous, donc on n'a rien besoin de faire. On veut ensuite annuler le premier coefficient de la troisième ligne. On retranche donc (-1) fois la première ligne à la troisième³ :

$$\begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ -1 & 1 & -2 & -2 \end{bmatrix} \xrightarrow{\ell_3 \leftarrow \ell_3 + \ell_1} \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ 0 & 1 & -1 & 0 \end{bmatrix}$$

Ceci revient à multiplier \tilde{A} à gauche par la matrice $E_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$.

La deuxième ligne a un terme non nul en deuxième position (2) : c'est un pivot. On va maintenant annuler le deuxième terme de la troisième ligne ; pour cela, on retranche $1/2$ fois la ligne 2 à la ligne 3 :

3. Bien sûr, ceci revient à ajouter la première ligne ! Il est cependant préférable de parler systématiquement de "retrancher" quitte à utiliser un coefficient négatif, car c'est ce qu'on fait conceptuellement : pour l'élimination on enlève un multiple de la ligne du pivot à la ligne courante.

$$\begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \xrightarrow{\ell_3 \leftarrow \ell_3 - 1/2 \ell_2} \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix}.$$

Ceci revient à multiplier la matrice précédente à gauche par la matrice $E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix}$. On a ici obtenu une

matrice sous forme triangulaire supérieure à trois pivots : on peut donc faire la remontée pour obtenir la solution du système, et on obtient (en notant x_i les composantes de \mathbf{x}) : $x_3 = 1$ puis $x_2 = 1$ et enfin $x_1 = 1$.

On a ainsi résolu le système linéaire.

On rappelle que le fait de travailler sur la matrice augmentée est extrêmement pratique car il permet de travailler simultanément sur les coefficients du système linéaire et sur le second membre.

Finalement, au moyen des opérations décrites ci-dessus, on a transformé le système linéaire

$$A\mathbf{x} = \mathbf{b} \text{ en } U\mathbf{x} = E_2 E_1 \mathbf{b}, \text{ où } U = E_2 E_1 A$$

est une matrice triangulaire supérieure.

Factorisation LU Tout va donc très bien pour ce système, mais supposons maintenant qu'on ait à résoudre 3089 systèmes, avec la même matrice A mais 3089 seconds membres \mathbf{b} différents⁴. Il serait un peu dommage de recommencer les opérations ci-dessus 3089 fois, alors qu'on peut en éviter une bonne partie. Comment faire ? L'idée est de "factoriser" la matrice A , c.à.d de l'écrire comme un produit $A = LU$, où L est triangulaire inférieure (lower triangular) et U triangulaire supérieure (upper triangular). On reformule alors le système $A\mathbf{x} = \mathbf{b}$ sous la forme $LU\mathbf{x} = \mathbf{b}$ et on résout maintenant deux systèmes faciles à résoudre car triangulaires : $L\mathbf{y} = \mathbf{b}$ et $U\mathbf{x} = \mathbf{y}$. La factorisation LU de la matrice découle immédiatement de l'algorithme de Gauss. Voyons comment sur l'exemple précédent.

1/ On remarque que $U = E_2 E_1 A$ peut aussi s'écrire $A = LU$, avec $L = (E_2 E_1)^{-1}$.

2/ On sait que $(E_2 E_1)^{-1} = (E_1)^{-1} (E_2)^{-1}$.

3/ Les matrices inverses E_1^{-1} et E_2^{-1} sont faciles à déterminer : comme E_2 consiste à retrancher 1/2 fois la ligne 2 à la ligne 3, l'opération inverse consiste à ajouter 1/2 fois la ligne 2 à la ligne 3, et donc

$$E_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{bmatrix}.$$

Il est facile de voir que $E_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$ et donc $L = E_1^{-1} E_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & \frac{1}{2} & 1 \end{bmatrix}$.

La matrice L est une matrice triangulaire inférieure (et c'est d'ailleurs pour cela qu'on l'appelle L , pour "lower" in English...) dont les coefficients sont particulièrement simples à trouver : les termes diagonaux sont tous égaux à un, et **chaque terme non nul sous-diagonal $\ell_{i,j}$ est égal au coefficient par lequel on a multiplié la ligne pivot i avant de la retrancher à la ligne j .**

4/ On a bien donc $A = LU$ avec L triangulaire inférieure (lower triangular) et U triangulaire supérieure (upper triangular).

La procédure qu'on vient d'expliquer s'appelle **méthode LU** pour la résolution des systèmes linéaires, et elle est d'une importance considérable dans les sciences de l'ingénieur, puisqu'elle est utilisée dans les programmes informatiques pour la résolution des systèmes linéaires.

Dans l'exemple que nous avons étudié, tout se passait très bien car nous n'avons pas eu de zéro en position pivotale. Si on a un zéro en position pivotale, la factorisation peut quand même se faire, mais au prix d'une permutation.

4. Ceci est courant dans les applications. Par exemple on peut vouloir calculer la réponse d'une structure de génie civil à 3089 chargements différents.

$$A^{(i+1)} = \begin{bmatrix} a_{1,1}^{(1)} & \dots & a_{1,N}^{(1)} \\ 0 & & \\ \vdots & & \\ 0 & \dots & 0 & a_{i+1,i+1}^{(i+1)} & \dots & a_{i+1,N}^{(i+1)} \\ & & & a_{i+2,i+1}^{(i+1)} & \dots & \\ & & & & \ddots & \\ 0 & \dots & 0 & a_{N,i+1}^{(i+1)} & \dots & a_{N,N}^{(i+1)} \end{bmatrix}$$

FIGURE 1.3: Allure de la matrice de Gauss à l'étape $i + 1$

Le résultat général que l'on peut démontrer est que si la matrice A est inversible, alors il existe une matrice de permutation P , une matrice triangulaire inférieure L et une matrice triangulaire supérieure U telles que $PA = LU$: voir le théorème 1.19.

Le cas général d'une matrice $n \times n$

De manière plus générale, pour une matrice A carrée d'ordre n , la méthode de Gauss s'écrit :

On pose $A^{(1)} = A$ et $b^{(1)} = b$. Pour $i = 1, \dots, n-1$, on cherche à calculer $A^{(i+1)}$ et $b^{(i+1)}$ tels que les systèmes $A^{(i)}x = b^{(i)}$ et $A^{(i+1)}x = b^{(i+1)}$ soient équivalents, où $A^{(i+1)}$ est une matrice dont les coefficients sous-diagonaux des colonnes 1 à i sont tous nuls, voir figure 1.3 :

Une fois la matrice $A^{(n)}$ (triangulaire supérieure) et le vecteur $b^{(n)}$ calculés, il sera facile de résoudre le système $A^{(n)}x = b^{(n)}$. Le calcul de $A^{(n)}$ est l'étape de "factorisation", le calcul de $b^{(n)}$ l'étape de "descente", et le calcul de x l'étape de "remontée". Donnons les détails de ces trois étapes.

Etape de factorisation et descente Pour passer de la matrice $A^{(i)}$ à la matrice $A^{(i+1)}$, on va effectuer des combinaisons linéaires entre lignes qui permettront d'annuler les coefficients de la i -ème colonne situés en dessous de la ligne i (dans le but de se rapprocher d'une matrice triangulaire supérieure). Evidemment, lorsqu'on fait ceci, il faut également modifier le second membre b en conséquence. L'étape de factorisation et descente s'écrit donc :

1. Pour $k \leq i$ et pour $j = 1, \dots, n$, on pose $a_{k,j}^{(i+1)} = a_{k,j}^{(i)}$ et $b_k^{(i+1)} = b_k^{(i)}$.
2. Pour $k > i$, si $a_{i,i}^{(i)} \neq 0$, on pose :

$$a_{k,j}^{(i+1)} = a_{k,j}^{(i)} - \frac{a_{k,i}^{(i)}}{a_{i,i}^{(i)}} a_{i,j}^{(i)}, \text{ pour } j = i, \dots, n, \quad (1.33)$$

$$b_k^{(i+1)} = b_k^{(i)} - \frac{a_{k,i}^{(i)}}{a_{i,i}^{(i)}} b_i^{(i)}. \quad (1.34)$$

La matrice $A^{(i+1)}$ est de la forme donnée sur la figure 1.3. Remarquons que le système $A^{(i+1)}x = b^{(i+1)}$ est bien équivalent au système $A^{(i)}x = b^{(i)}$.

Si la condition $a_{i,i}^{(i)} \neq 0$ est vérifiée pour $i = 1$ à n , on obtient par le procédé de calcul ci-dessus un système linéaire $A^{(n)}x = b^{(n)}$ équivalent au système $Ax = b$, avec une matrice $A^{(n)}$ triangulaire supérieure facile à inverser. On verra un peu plus loin les techniques de pivot qui permettent de régler le cas où la condition $a_{i,i}^{(i)} \neq 0$ n'est pas vérifiée.

Étape de remontée Il reste à résoudre le système $A^{(n)}x = b^{(n)}$. Ceci est une étape facile. Comme $A^{(n)}$ est une matrice inversible, on a $a_{i,i}^{(i)} \neq 0$ pour tout $i = 1, \dots, n$, et comme $A^{(n)}$ est une matrice triangulaire supérieure, on peut donc calculer les composantes de x en “remontant”, c’est-à-dire de la composante x_n à la composante x_1 :

$$x_n = \frac{b_n^{(n)}}{a_{n,n}^{(n)}},$$

$$x_i = \frac{1}{a_{i,i}^{(i)}} \left[b_i^{(n)} - \sum_{j=i+1,n} a_{i,j}^{(n)} x_j \right], i = n-1, \dots, 1.$$

Il est important de savoir mettre sous forme algorithmique les opérations que nous venons de décrire : c’est l’étape clef avant l’écriture d’un programme informatique qui nous permettra de faire faire le boulot par l’ordinateur !

Algorithme 1.11 (Gauss sans permutation).

1. (Factorisation et descente)

Pour commencer, on pose $u_{i,j} = a_{i,j}$ et $y_i = b_i$ pour $i, j \in \{1, \dots, n\}$.

Puis, pour i allant de 1 à $n-1$, on effectue les calculs suivants :

(a) On ne change pas la i -ème ligne (qui est la ligne du pivot)

(b) On change les lignes $i+1$ à n et le second membre y en utilisant la ligne i .

Pour k allant de $i+1$ à n :

$\ell_{k,i} = \frac{u_{k,i}}{u_{i,i}}$ (si $a_{i,i} = 0$, prendre la méthode avec pivot partiel)

pour j allant de $i+1$ à n ,

$u_{k,j} = u_{k,j} - \ell_{k,i} u_{i,j}$

Fin pour

$y_k = y_k - \ell_{k,i} y_i$

Fin pour

2. (Remontée) On calcule x :

$x_n = \frac{y_n}{u_{n,n}}$

Pour i allant de $n-1$ à 1,

$x_i = y_i$

Pour j allant de $i+1$ à n ,

$x_i = x_i - u_{i,j} x_j$

Fin pour

$x_i = \frac{1}{u_{i,i}} x_i$

Fin pour

Coût de la méthode de Gauss (nombre d’opérations) On peut montrer (on fera le calcul de manière détaillée pour la méthode de Choleski dans la section suivante, le calcul pour Gauss est similaire) que le nombre d’opérations nécessaires n_G pour effectuer les étapes de factorisation, descente et remontée est $\frac{2}{3}n^3 + O(n^2)$; on rappelle qu’une fonction f de \mathbb{N} dans \mathbb{N} est $O(n^2)$ veut dire qu’il existe un réel constant C tel que $f(n) \leq Cn^2$. On a donc $\lim_{n \rightarrow +\infty} \frac{n_G}{n^3} = \frac{2}{3}$: lorsque n est grand, le nombre d’opérations se comporte comme $(2/3)n^3$.

En ce qui concerne la place mémoire, on peut très bien stocker les itérés $A^{(i)}$ dans la matrice A de départ, ce qu’on n’a pas voulu faire dans le calcul précédent, par souci de clarté.

Décomposition LU Si le système $Ax = b$ doit être résolu pour plusieurs second membres b , on a déjà dit qu'on a intérêt à ne faire l'étape de factorisation (*i.e.* le calcul de $A^{(n)}$), qu'une seule fois, alors que les étapes de descente et remontée (*i.e.* le calcul de $b^{(n)}$ et x) seront faits pour chaque vecteur b . L'étape de factorisation peut se faire en décomposant la matrice A sous la forme LU . Supposons toujours pour l'instant que lors de l'algorithme de Gauss, la condition $a_{i,i}^{(i)} \neq 0$ est vérifiée pour tout $i = 1, \dots, n$. La matrice L a comme coefficients $\ell_{k,i} = \frac{a_{k,i}^{(i)}}{a_{i,i}^{(i)}}$ pour $k > i$, $\ell_{i,i} = 1$ pour tout $i = 1, \dots, n$, et $\ell_{i,j} = 0$ pour $j > i$, et la matrice U est égale à la matrice $A^{(n)}$. On peut vérifier que $A = LU$ grâce au fait que le système $A^{(n)}x = b^{(n)}$ est équivalent au système $Ax = b$. En effet, comme $A^{(n)}x = b^{(n)}$ et $b^{(n)} = L^{-1}b$, on en déduit que $LUx = b$, et comme A et LU sont inversibles, on en déduit que $A^{-1}b = (LU)^{-1}b$ pour tout $b \in \mathbb{R}^n$. Ceci démontre que $A = LU$. La méthode LU se déduit donc de la méthode de Gauss en remarquant simplement que, ayant conservé la matrice L , on peut effectuer les calculs sur b après les calculs sur A , ce qui donne :

Algorithme 1.12 (LU simple (sans permutation)).

1. (Factorisation)

On pose $u_{i,j} = a_{i,j}$ pour $i, j \in \{1, \dots, n\}$.

Puis, pour i allant de 1 à $n-1$, on effectue les calculs suivants :

(a) On ne change pas la i -ème ligne

(b) On modifie les lignes $i+1$ à n ((mais pas le second membre) en utilisant la ligne i).

Pour k allant de $i+1$ à n :

$\ell_{k,i} = \frac{u_{k,i}}{u_{i,i}}$ (si $u_{i,i} = 0$, prendre la méthode avec pivot partiel)

pour j allant de $i+1$ à n ,

$u_{k,j} = u_{k,j} - \ell_{k,i}u_{i,j}$

Fin pour

2. (Descente) On calcule y (avec $Ly = b$)

Pour i allant de 1 à n ,

$y_i = b_i - \sum_{k=1}^{i-1} \ell_{i,k}y_k$ (on a ainsi implicitement $\ell_{i,i} = 1$)

3. (Remontée) On calcule x (avec $Ux = y$)

Pour i allant de n à 1,

$x_i = \frac{1}{u_{i,i}}(y_i - \sum_{j=i+1}^n u_{i,j}x_j)$

Remarque 1.13 (Optimisation mémoire). L'introduction des matrices L et U et des vecteurs y et x n'est pas nécessaire. Tout peut s'écrire avec la matrice A et le vecteur b , que l'on modifie au cours de l'algorithme. A la fin de la factorisation, U est stockée dans la partie supérieure de A (y compris la diagonale) et L dans la partie strictement inférieure de A (c'est-à-dire sans la diagonale, la diagonale de L est connue car toujours formée de 1). Dans l'algorithme précédent, on remplace donc “ u ” et “ l ” par “ a ”. De même, on remplace “ x ” et “ y ” par “ b ”. A la fin des étapes de descente et de remontée, la solution du problème est alors stockée dans b .

L'introduction de L , U , x et y peut toutefois aider à comprendre la méthode.

Nous allons maintenant donner une condition nécessaire et suffisante (CNS) pour qu'une matrice A admette une décomposition LU avec U inversible et sans permutation. Commençons par un petit lemme technique qui va nous permettre de prouver cette CNS.

Lemme 1.14 (Décomposition LU de la matrice principale d'ordre k). Soit $n \in \mathbb{N}$, $A \in \mathcal{M}_n(\mathbb{R})$ et $k \in \{1, \dots, n\}$. On appelle matrice principale d'ordre k de A la matrice $A_k \in \mathcal{M}_k(\mathbb{R})$ définie par $(A_k)_{i,j} = a_{i,j}$ pour $i = 1, \dots, k$ et $j = 1, \dots, k$. On suppose qu'il existe une matrice $L_k \in \mathcal{M}_k(\mathbb{R})$ triangulaire inférieure de coefficients diagonaux tous égaux à 1 et une matrice triangulaire supérieure $U_k \in \mathcal{M}_k(\mathbb{R})$ inversible, telles que $A_k = L_k U_k$. Alors A s'écrit sous la forme “par blocs” suivante :

$$A = \begin{bmatrix} L_k & 0_{k \times (n-k)} \\ C_k & \text{Id}_{n-k} \end{bmatrix} \begin{bmatrix} U_k & B_k \\ 0_{(n-k) \times k} & D_k \end{bmatrix}, \quad (1.35)$$

où $0_{p,q}$ désigne la matrice nulle de dimension $p \times q$, $B_k \in \mathcal{M}_{k,n-k}(\mathbb{R})$ et $C_k \in \mathcal{M}_{n-k,k}(\mathbb{R})$ et $D_k \in \mathcal{M}_{n-k,n-k}(\mathbb{R})$; de plus, la matrice principale d'ordre $k+1$ s'écrit sous la forme

$$A_{k+1} = \begin{bmatrix} L_k & 0_{1 \times k} \\ c_k & 1 \end{bmatrix} \begin{bmatrix} U_k & b_k \\ 0_{k \times 1} & d_k \end{bmatrix} \quad (1.36)$$

où $b \in \mathcal{M}_{k,1}(\mathbb{R})$ est la première colonne de la matrice B_k , $c_k \in \mathcal{M}_{1,k}$ est la première ligne de la matrice C_k , et d_k est le coefficient de la ligne 1 et colonne 1 de D_k .

DÉMONSTRATION – On écrit la décomposition par blocs de A :

$$A = \begin{bmatrix} A_k & E_k \\ F_k & G_k \end{bmatrix},$$

avec $A_k \in \mathcal{M}_k(\mathbb{R})$, $E_k \in \mathcal{M}_{k,n-k}(\mathbb{R})$, $F_k \in \mathcal{M}_{n-k,k}(\mathbb{R})$ et $G_k \in \mathcal{M}_{n-k,n-k}(\mathbb{R})$. Par hypothèse, on a $A_k = L_k U_k$. De plus L_k et U_k sont inversibles, et il existe donc une unique matrice $B_k \in \mathcal{M}_{k,n-k}(\mathbb{R})$ (resp. $C_k \in \mathcal{M}_{n-k,k}(\mathbb{R})$) telle que $L_k B_k = E_k$ (resp. $C_k U_k = F_k$). En posant $D_k = G_k - C_k B_k$, on obtient (1.35). L'égalité (1.36) en découle immédiatement. ■

Proposition 1.15 (CNS pour LU sans permutation). Soit $n \in \mathbb{N}$, $A \in \mathcal{M}_n(\mathbb{R})$. Les deux propriétés suivantes sont équivalentes.

(P1) Il existe un unique couple (L, U) , avec L matrice triangulaire inférieure de coefficients égaux à 1 et U une matrice inversible triangulaire supérieure, tel que $A = LU$.

(P2) Les mineurs principaux⁵ de A sont tous non nuls.

DÉMONSTRATION – Si $A = LU$ avec L triangulaire inférieure de coefficients égaux à 1 et U inversible triangulaire supérieure, alors $A_k = L_k U_k$ où les matrices L_k et U_k les matrices principales d'ordre k de L et U , qui sont encore respectivement triangulaire inférieure de coefficients égaux à 1 et inversible triangulaire supérieure. On a donc

$$\det(A_k) = \det(L_k) \det(U_k) \neq 0 \text{ pour tout } k = 1, \dots, n,$$

et donc (P1) \Rightarrow (P2).

Montrons maintenant la réciproque. On suppose que les mineurs sont non nuls, et on va montrer que $A = LU$. On va en fait montrer que pour tout $k = 1, \dots, n$, on a $A_k = L_k U_k$ où L_k triangulaire inférieure de coefficients égaux à 1 et U_k inversible triangulaire supérieure. Le premier mineur est non nul, donc $a_{11} = 1 \times a_{11}$, et la récurrence est bien initialisée. On la suppose vraie à l'étape k . Par le lemme 1.14, on a donc A_{k+1} qui est de la forme (1.36), et donc une $A_{k+1} = L_{k+1} U_{k+1}$. Comme $\det(A_{k+1}) \neq 0$, la matrice U_{k+1} est inversible, et l'hypothèse de récurrence est vérifiée à l'ordre $k+1$. On a donc bien (P2) \Rightarrow (P1) (l'unicité de L et U est laissée en exercice). ■

Que faire en cas de pivot nul : la technique de permutation La caractérisation que nous venons de donner pour qu'une matrice admette une décomposition LU sans permutation est intéressante mathématiquement, mais de peu d'intérêt en pratique. On ne va en effet jamais calculer n déterminants pour savoir si on doit ou non permuter. En pratique, on effectue la décomposition LU sans savoir si on a le droit ou non de le faire, avec ou sans permutation. Au cours de l'élimination, si $a_{i,i}^{(i)} = 0$, on va permuter la ligne i avec une des lignes suivantes telle que $a_{k,i}^{(i)} \neq 0$. Notons que si le "pivot" $a_{i,i}^{(i)}$ est très petit, son utilisation peut entraîner des erreurs d'arrondi importantes dans les calculs et on va là encore permuter. En fait, même dans le cas où la CNS donnée par la proposition 1.15 est vérifiée, la plupart des fonctions de bibliothèques scientifiques vont permuter.

Plaçons-nous à l'itération i de la méthode de Gauss. Comme la matrice $A^{(i)}$ est forcément non singulière, on a :

$$\det(A^{(i)}) = a_{1,1}^{(i)} a_{2,2}^{(i)} \cdots a_{i-1,i-1}^{(i)} \det \begin{pmatrix} a_{i,i}^{(i)} & \cdots & a_{i,n}^{(i)} \\ \vdots & \ddots & \vdots \\ a_{n,i}^{(i)} & \cdots & a_{n,n}^{(i)} \end{pmatrix} \neq 0.$$

5. On rappelle que le mineur principal d'ordre k est le déterminant de la matrice principale d'ordre k .

On a donc en particulier

$$\det \begin{pmatrix} a_{i,i}^{(i)} & \dots & a_{i,n}^{(i)} \\ \vdots & \ddots & \vdots \\ a_{n,i}^{(i)} & \dots & a_{n,n}^{(i)} \end{pmatrix} \neq 0.$$

On déduit qu'il existe $i_0 \in \{i, \dots, n\}$ tel que $a_{i_0,i}^{(i)} \neq 0$. On choisit alors $i_0 \in \{i, \dots, n\}$ tel que $|a_{i_0,i}^{(i)}| = \max\{|a_{k,i}^{(i)}|, k = i, \dots, n\}$. Le choix de ce max est motivé par le fait qu'on aura ainsi moins d'erreur d'arrondi. On échange alors les lignes i et i_0 (dans la matrice A et le second membre b) et on continue la procédure de Gauss décrite plus haut.

L'intérêt de cette stratégie de pivot est qu'on aboutit toujours à la résolution du système (dès que A est inversible).

Remarque 1.16 (Pivot total). *La méthode que nous venons de d'écrire est souvent nommée technique de pivot "partiel". On peut vouloir rendre la norme du pivot encore plus grande en considérant tous les coefficients restants et pas uniquement ceux de la colonne i . A l'étape i , on choisit maintenant i_0 et $j_0 \in \{i, \dots, n\}$ tels que $|a_{i_0,j_0}^{(i)}| = \max\{|a_{k,j}^{(i)}|, k = i, \dots, n, j = i, \dots, n\}$, et on échange alors les lignes i et i_0 (dans la matrice A et le second membre b), les colonnes i et j_0 de A et les inconnues x_i et x_{j_0} . La stratégie du pivot total permet une moins grande sensibilité aux erreurs d'arrondi. L'inconvénient majeur est qu'on change la structure de A : si, par exemple la matrice avait tous ses termes non nuls sur quelques diagonales seulement, ceci n'est plus vrai pour la matrice $A^{(n)}$.*

Ecrivons maintenant l'algorithme de la méthode LU avec pivot partiel ; pour ce faire, on va simplement remarquer que l'ordre dans lequel les équations sont prises n'a aucune importance pour l'algorithme. Au départ de l'algorithme, on initialise la bijection t de $\{1, \dots, n\}$ dans $\{1, \dots, n\}$ par l'identité, c.à.d. $t(i) = i$; cette bijection t va être modifiée au cours de l'algorithme pour tenir compte du choix du pivot.

Algorithme 1.17 (LU avec pivot partiel).

1. (Initialisation de t) Pour i allant de 1 à n , $t(i) = i$. Fin pour

2. (Factorisation)

Pour i allant de 1 à n , on effectue les calculs suivants :

(a) Choix du pivot (et de $t(i)$) : on cherche $i^* \in \{i, \dots, n\}$ t.q. $|a_{t(i^*),i}| = \max\{|a_{t(k),i}|, k \in \{i, \dots, n\}\}$ (noter que ce max est forcément non nul car la matrice est inversible).

On modifie alors t en inversant les valeurs de $t(i)$ et $t(i^*)$.

$p = t(i^*)$; $t(i^*) = t(i)$; $t(i) = p$.

On ne change pas la ligne $t(i)$:

$u_{t(i),j} = a_{t(i),j}$ pour $j = i, \dots, n$,

(b) On modifie les lignes $t(k)$, $k > i$ (et le second membre), en utilisant la ligne $t(i)$.

Pour $k = i + 1, \dots, n$ (noter qu'on a uniquement besoin de connaître l'ensemble, et pas l'ordre) :

$$\ell_{t(k),i} = \frac{a_{t(k),i}}{a_{t(i),i}}$$

Pour j allant de $i + 1$ à n ,

$$a_{t(k),j} = a_{t(k),j} - \ell_{t(k),i} u_{t(i),j}$$

Fin pour

Fin pour

3. (Descente) On calcule y

Pour i allant de 1 à n ,

$$y_{t(i)} = b_{t(i)} - \sum_{j=1}^{i-1} \ell_{t(j),i} y_{t(j)}$$

Fin pour

4. (Remontée) On calcule x

Pour i allant de n à 1,

$$x_{t(i)} = \frac{1}{u_{t(i),i}}(y_i - \sum_{j=i+1}^n u_{t(i),j} x_j)$$

Fin pour

NB : On a changé l'ordre dans lequel les équations sont considérées (le tableau t donne cet ordre, et donc la matrice P). Donc, on a aussi changé l'ordre dans lequel interviennent les composantes du second membre : le système $Ax = b$ est devenu $PAx = Pb$. Par contre, on n'a pas touché à l'ordre dans lequel interviennent les composantes de x et y .

Il reste maintenant à signaler la propriété magnifique de cet algorithme... Il est inutile de connaître *a priori* la bijection pour cet algorithme. A l'étape i de l'item 1 (et d'ailleurs aussi à l'étape i de l'item 2), il suffit de connaître $t(j)$ pour j allant de 1 à i , les opérations de 1(b) se faisant alors sur toutes les autres lignes (dans un ordre quelconque). Il suffit donc de partir d'une bijection arbitraire de $\{1, \dots, n\}$ dans $\{1, \dots, n\}$ (par exemple l'identité) et de la modifier à chaque étape. Pour que l'algorithme aboutisse, il suffit que $a_{t(i),i} \neq 0$ (ce qui toujours possible car A est inversible).

Remarque 1.18 (Ordre des équations et des inconnues). *L'algorithme se ramène donc à résoudre $LUx = b$, en résolvant d'abord $Ly = b$ puis $Ux = y$. Notons que lors de la résolution du système $Ly = b$, les équations sont dans l'ordre $t(1), \dots, t(k)$ (les composantes de b sont donc aussi prises dans cet ordre), mais le vecteur y est bien le vecteur de composantes (y_1, \dots, y_n) , dans l'ordre initial. Puis, on résout $Ux = y$, et les équations sont encore dans l'ordre $t(1), \dots, t(k)$ mais les vecteurs x et y ont comme composantes respectives (x_1, \dots, x_n) et (y_1, \dots, y_n) .*

Le théorème d'existence L'algorithme LU avec pivot partiel nous permet de démontrer le théorème d'existence de la décomposition LU pour une matrice inversible.

Théorème 1.19 (Décomposition LU d'une matrice). *Soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice inversible, il existe une matrice de permutation P telle que, pour cette matrice de permutation, il existe un et un seul couple de matrices (L, U) où L est triangulaire inférieure de termes diagonaux égaux à 1 et U est triangulaire supérieure, vérifiant*

$$PA = LU.$$

DÉMONSTRATION –

1. **L'existence** de la matrice P et des matrices L, U peut s'effectuer en s'inspirant de l'algorithme "LU avec pivot partiel" 1.17). Posons $A^{(0)} = A$.

À chaque étape i de l'algorithme 1.17 peut s'écrire comme $A^{(i)} = E^{(i)} P^{(i)} A^{(i-1)}$, où $P^{(i)}$ est la matrice de permutation qui permet le choix du pivot partiel, et $E^{(i)}$ est une matrice d'élimination qui effectue les combinaisons linéaires de lignes permettant de mettre à zéro tous les coefficients de la colonne i situés en dessous de la ligne i . Pour simplifier, raisonnons sur une matrice 4×4 (le raisonnement est le même pour une matrice $n \times n$. On a donc en appliquant l'algorithme de Gauss :

$$E^{(3)} P^{(3)} E^{(2)} P^{(2)} E^{(1)} P^{(1)} A = U$$

Les matrices $P^{(i+1)}$ et $E^{(i)}$ ne permutent en général pas. Prenons par exemple E_2 , qui est de la forme

$$E^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & a & 1 & 0 \\ 0 & b & 0 & 1 \end{bmatrix}$$

Si $P^{(3)}$ est la matrice qui échange les lignes 3 et 4, alors

$$P^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ et } P^{(3)} E^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & b & 0 & 1 \\ 0 & a & 1 & 0 \end{bmatrix}, \text{ alors que } E^{(2)} P^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & a & 0 & 1 \\ 0 & b & 1 & 0 \end{bmatrix}$$

Mais par contre, comme la multiplication à gauche par $P^{(i+1)}$ permute les lignes $i+1$ et $i+k$, pour un certain $k \geq 1$, et que la multiplication à droite permute les colonnes $i+1$ et $i+k$, la matrice $\widetilde{E^{(i)}} = P^{(i+1)} E^{(i)} P^{(i+1)}$ est encore une matrice triangulaire inférieure avec la même structure que $E^{(i)}$: on a juste échangé les coefficients extradiagonaux des lignes $i+1$ et $i+k$. On a donc

$$P^{(i+1)} E^{(i)} = \widetilde{E^{(i)}} P^{(i+1)}. \quad (1.37)$$

Dans l'exemple précédent, on effectue le calcul :

$$P^{(3)} E^{(2)} P^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & b & 1 & 0 \\ 0 & a & 0 & 1 \end{bmatrix} = \widetilde{E^{(2)}},$$

qui est une matrice triangulaire inférieure de coefficients tous égaux à 1, et comme $P^{(3)} P^{(3)} = \text{Id}$, on a donc :

$$P^{(3)} E^{(2)} = \widetilde{E^{(2)}} P^{(3)}.$$

Pour revenir à notre exemple $n = 4$, on peut donc écrire :

$$E^{(3)} \widetilde{E^{(2)}} P^{(3)} \widetilde{E^{(1)}} P^{(2)} P^{(1)} A = U$$

Mais par le même raisonnement que précédemment, on a $P^{(3)} \widetilde{E^{(1)}} = \widetilde{\widetilde{E^{(1)}}} P^{(3)}$ où $\widetilde{\widetilde{E^{(1)}}}$ est encore une matrice triangulaire inférieure avec des 1 sur la diagonale. On en déduit que

$$E^{(3)} \widetilde{E^{(2)}} \widetilde{\widetilde{E^{(1)}}} P^{(3)} P^{(2)} P^{(1)} A = U, \text{ soit encore } PA = LU$$

où $P = P^{(3)} P^{(2)} P^{(1)}$ bien une matrice de permutation, et $L = (E^{(3)} \widetilde{E^{(2)}} \widetilde{\widetilde{E^{(1)}}})^{-1}$ est une matrice triangulaire inférieure avec des 1 sur la diagonale.

Le raisonnement que nous venons de faire pour $n = 3$ se généralise facilement à n quelconque. Dans ce cas, l'échelonnement de la matrice s'écrit sous la forme

$$U = E^{(n-1)} P^{(n-1)} \dots E^{(2)} P^{(2)} E^{(1)} P^{(1)} A,$$

et se transforme grâce à (1.37) en

$$U = F^{(n-1)} \dots F^{(2)} F^{(1)} P^{(n-1)} \dots P^{(2)} P^{(1)} A,$$

où les matrices $F^{(i)}$ sont des matrices triangulaires inférieures de coefficients diagonaux tous égaux à 1. Plus précisément, $F^{(n-1)} = E^{(n-1)}$, $F^{(n-2)} = \widetilde{E^{(n-2)}}$, $F^{(n-3)} = \widetilde{\widetilde{E^{(n-3)}}}$, etc... On a ainsi démontré l'existence de la décomposition LU .

2. Pour montrer l'unicité du couple (L, U) à P donnée, supposons qu'il existe une matrice P et des matrices L_1, L_2 , triangulaires inférieures et U_1, U_2 , triangulaires supérieures, telles que

$$PA = L_1 U_1 = L_2 U_2$$

Dans ce cas, on a donc $L_2^{-1} L_1 = U_2 U_1^{-1}$. Or la matrice $L_2^{-1} L_1$ est une matrice triangulaire inférieure dont les coefficients diagonaux sont tous égaux à 1, et la matrice $U_2 U_1^{-1}$ est une matrice triangulaire supérieure. On en déduit que $L_2^{-1} L_1 = U_2 U_1^{-1} = \text{Id}$, et donc que $L_1 = L_2$ et $U_1 = U_2$. ■

Remarque 1.20 (Décomposition LU pour les matrices non inversibles). *En fait n'importe quelle matrice carrée admet une décomposition de la forme $PA = LU$. Mais si la matrice A n'est pas inversible, son échelonnement va nous donner des lignes de zéros pour les dernières lignes. La décomposition LU n'est dans ce cas pas unique. Cette remarque fait l'objet de l'exercice 27.*

1.3.3 Méthode de Choleski

On va maintenant étudier la méthode de Choleski, qui est une méthode directe adaptée au cas où A est symétrique définie positive. On rappelle qu'une matrice $A \in \mathcal{M}_n(\mathbb{R})$ de coefficients $(a_{i,j})_{i=1,n,j=1,n}$ est symétrique si $A = A^t$, où A^t désigne la transposée de A , définie par les coefficients $(a_{j,i})_{i=1,n,j=1,n}$, et que A est définie positive si $Ax \cdot x > 0$ pour tout $x \in \mathbb{R}^n$ tel que $x \neq 0$. Dans toute la suite, $x \cdot y$ désigne le produit scalaire des deux vecteurs x et y de \mathbb{R}^n . On rappelle (exercice) que si A est symétrique définie positive elle est en particulier inversible.

Description de la méthode

Commençons par un exemple. On considère la matrice $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$, qui est symétrique. Calculons sa décomposition LU . Par échelonnement, on obtient

$$A = LU = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & -\frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{bmatrix}$$

La structure LU ne conserve pas la symétrie de la matrice A . Pour des raisons de coût mémoire, il est important de pouvoir la conserver. Une façon de faire est de décomposer U en sa partie diagonale fois une matrice triangulaire. On obtient

$$U = \begin{bmatrix} 2 & 0 & 0 \\ 0 & \frac{3}{2} & 0 \\ 0 & 0 & \frac{4}{3} \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & 1 & -\frac{2}{3} \\ 0 & 0 & 1 \end{bmatrix}$$

On a donc $U = DL^t$, et comme tous les coefficients de D sont positifs, on peut écrire $D = \sqrt{D}\sqrt{D}$, où \sqrt{D} est la matrice diagonale dont les éléments diagonaux sont les racines carrées des éléments diagonaux de A . On a donc $A = L\sqrt{D}\sqrt{D}L^t = \tilde{L}\tilde{L}^t$, avec $\tilde{L} = L\sqrt{D}$. Notons que la matrice \tilde{L} est toujours triangulaire inférieure, mais ses coefficients diagonaux ne sont plus astreints à être égaux à 1. C'est la décomposition de Choleski de la matrice A .

De fait, la méthode de Choleski consiste donc à trouver une décomposition d'une matrice A symétrique définie positive de la forme $A = LL^t$, où L est triangulaire inférieure de coefficients diagonaux strictement positifs. On résout alors le système $Ax = b$ en résolvant d'abord $Ly = b$ puis le système $L^tx = y$. Une fois la matrice A "factorisée", c'est-à-dire la décomposition LL^t obtenue (voir paragraphe suivant), on effectue les étapes de "descente" et "remontée" :

1. Etape 1 : "descente" Le système $Ly = b$ s'écrit :

$$Ly = \begin{bmatrix} \ell_{1,1} & 0 & & \\ \vdots & \ddots & \vdots & \\ \ell_{n,1} & \dots & \ell_{n,n} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

Ce système s'écrit composante par composante en partant de $i = 1$.

$$\begin{aligned} \ell_{1,1}y_1 &= b_1, \text{ donc } y_1 = \frac{b_1}{\ell_{1,1}} \\ \ell_{2,1}y_1 + \ell_{2,2}y_2 &= b_2, \text{ donc } y_2 = \frac{1}{\ell_{2,2}}(b_2 - \ell_{2,1}y_1) \\ &\vdots \\ \sum_{j=1,i} \ell_{i,j}y_j &= b_i, \text{ donc } y_i = \frac{1}{\ell_{i,i}}(b_i - \sum_{j=1,i-1} \ell_{i,j}y_j) \\ &\vdots \\ \sum_{j=1,n} \ell_{n,j}y_j &= b_n, \text{ donc } y_n = \frac{1}{\ell_{n,n}}(b_n - \sum_{j=1,n-1} \ell_{n,j}y_j). \end{aligned}$$

On calcule ainsi y_1, y_2, \dots, y_n .

2. Etape 2 : “remontée” On calcule maintenant x solution de $L^t x = y$.

$$L^t x = \begin{bmatrix} \ell_{1,1} & \ell_{2,1} & \dots & \ell_{n,1} \\ 0 & \ddots & & \\ \vdots & & \ddots & \\ 0 & \dots & & \ell_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

On a donc :

$$\ell_{n,n} x_n = y_n \text{ donc } x_n = \frac{y_n}{\ell_{n,n}}$$

$$\ell_{n-1,n-1} x_{n-1} + \ell_{n,n-1} x_n = y_{n-1} \text{ donc } x_{n-1} = \frac{y_{n-1} - \ell_{n,n-1} x_n}{\ell_{n-1,n-1}}$$

\vdots

$$\sum_{j=1,n} \ell_{j,1} x_j = y_1 \text{ donc } x_1 = \frac{y_1 - \sum_{j=2,n} \ell_{j,1} x_j}{\ell_{1,1}}.$$

On calcule ainsi x_n, x_{n-1}, \dots, x_1 .

Existence et unicité de la décomposition

Soit A une matrice symétrique définie positive. On sait déjà par le théorème 1.19 page 37, qu’il existe une matrice de permutation et L triangulaire inférieure et U triangulaire supérieure telles que $PA = LU$. L’avantage dans le cas où la matrice est symétrique définie positive, est que la décomposition est toujours possible sans permutation. On prouve l’existence et unicité en construisant la décomposition, c.à.d. en construisant la matrice L .

Pour comprendre le principe de la preuve, commençons d’abord par le cas $n = 2$. Dans ce cas on peut écrire

$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$. On sait que $a > 0$ car A est s.d.p. . L’échelonnement de A donne donc

$$A = LU = \begin{bmatrix} 1 & 0 \\ \frac{b}{a} & 1 \end{bmatrix} \begin{bmatrix} a & b \\ 0 & c - \frac{b^2}{a} \end{bmatrix}$$

En extrayant la diagonale de U , on obtient :

$$A = LU = \begin{bmatrix} 1 & 0 \\ \frac{b}{a} & 1 \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & c - \frac{b^2}{a} \end{bmatrix} \begin{bmatrix} a & \frac{b}{a} \\ 0 & 1 \end{bmatrix}.$$

Et donc

$$A = \tilde{L} \tilde{L}^t \text{ avec } \tilde{L} = \begin{bmatrix} \sqrt{a} & 0 \\ b \sqrt{\frac{a}{ac-b^2}} & 1 \end{bmatrix}.$$

Théorème 1.21 (Décomposition de Choleski). *Soit $A \in \mathcal{M}_n(\mathbb{R})$ ($n \geq 1$) une matrice symétrique définie positive. Alors il existe une unique matrice $L \in \mathcal{M}_n(\mathbb{R})$, $L = (\ell_{i,j})_{i,j=1}^n$, telle que :*

1. L est triangulaire inférieure (c’est-à-dire $\ell_{i,j} = 0$ si $j > i$),
2. $\ell_{i,i} > 0$, pour tout $i \in \{1, \dots, n\}$,
3. $A = LL^t$.

DÉMONSTRATION –

I- Existence de L : démonstration par récurrence sur n

1. Dans le cas $n = 1$, on a $A = (a_{1,1})$. Comme A est symétrique définie positive, on a $a_{1,1} > 0$. On peut donc définir $L = (\ell_{1,1})$ où $\ell_{1,1} = \sqrt{a_{1,1}}$, et on a bien $A = LL^t$.
2. On suppose que la décomposition de Choleski s'obtient pour $A \in \mathcal{M}_p(\mathbb{R})$ symétrique définie positive, pour $1 \leq p \leq n$ et on va démontrer que la propriété est encore vraie pour $A \in \mathcal{M}_{n+1}(\mathbb{R})$ symétrique définie positive. Soit donc $A \in \mathcal{M}_{n+1}(\mathbb{R})$ symétrique définie positive ; on peut écrire A sous la forme :

$$A = \left[\begin{array}{c|c} B & a \\ \hline a^t & \alpha \end{array} \right] \quad (1.38)$$

où $B \in \mathcal{M}_n(\mathbb{R})$ est symétrique, $a \in \mathbb{R}^n$ et $\alpha \in \mathbb{R}$. Montrons que B est définie positive, c.à.d. que $By \cdot y > 0$, pour tout $y \in \mathbb{R}^n$ tel que $y \neq 0$. Soit donc $y \in \mathbb{R}^n \setminus \{0\}$, et $x = \begin{bmatrix} y \\ 0 \end{bmatrix} \in \mathbb{R}^{n+1}$. Comme A est symétrique définie positive, on a :

$$0 < Ax \cdot x = \left[\begin{array}{c|c} B & a \\ \hline a^t & \alpha \end{array} \right] \begin{bmatrix} y \\ 0 \end{bmatrix} \cdot \begin{bmatrix} y \\ 0 \end{bmatrix} = \begin{bmatrix} By \\ a^t y \end{bmatrix} \cdot \begin{bmatrix} y \\ 0 \end{bmatrix} = By \cdot y$$

donc B est définie positive. Par hypothèse de récurrence, il existe une matrice $M \in \mathcal{M}_n(\mathbb{R})$ $M = (m_{i,j})_{i,j=1}^n$ telle que :

- (a) $m_{i,j} = 0$ si $j > i$
- (b) $m_{i,i} > 0$
- (c) $B = MM^t$.

On va chercher L sous la forme :

$$L = \left[\begin{array}{c|c} M & 0 \\ \hline b^t & \lambda \end{array} \right] \quad (1.39)$$

avec $b \in \mathbb{R}^n$, $\lambda \in \mathbb{R}_+^*$ tels que $LL^t = A$. Pour déterminer b et λ , calculons LL^t où L est de la forme (1.39) et identifions avec A :

$$LL^t = \left[\begin{array}{c|c} M & 0 \\ \hline b^t & \lambda \end{array} \right] \left[\begin{array}{c|c} M^t & b \\ \hline 0 & \lambda \end{array} \right] = \left[\begin{array}{c|c} MM^t & Mb \\ \hline b^t M^t & b^t b + \lambda^2 \end{array} \right]$$

On cherche $b \in \mathbb{R}^n$ et $\lambda \in \mathbb{R}_+^*$ tels que $LL^t = A$, et on veut donc que les égalités suivantes soient vérifiées :

$$Mb = a \text{ et } b^t b + \lambda^2 = \alpha.$$

Comme M est inversible (en effet, le déterminant de M s'écrit $\det(M) = \prod_{i=1}^n m_{i,i} > 0$), la première égalité ci-dessus donne : $b = M^{-1}a$ et en remplaçant dans la deuxième égalité, on obtient : $(M^{-1}a)^t (M^{-1}a) + \lambda^2 = \alpha$, donc $a^t (M^t)^{-1} M^{-1} a + \lambda^2 = \alpha$ soit encore $a^t (MM^t)^{-1} a + \lambda^2 = \alpha$, c'est-à-dire :

$$a^t B^{-1} a + \lambda^2 = \alpha \quad (1.40)$$

Pour que (1.40) soit vérifiée, il faut que

$$\alpha - a^t B^{-1} a > 0 \quad (1.41)$$

Montrons que la condition (1.41) est effectivement vérifiée : Soit $z = \begin{pmatrix} B^{-1}a \\ -1 \end{pmatrix} \in \mathbb{R}^{n+1}$. On a $z \neq 0$ et donc $Az \cdot z > 0$ car A est symétrique définie positive. Calculons Az :

$$Az = \left[\begin{array}{c|c} B & a \\ \hline a^t & \alpha \end{array} \right] \begin{bmatrix} B^{-1}a \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ a^t B^{-1}a - \alpha \end{bmatrix}.$$

On a donc $Az \cdot z = \alpha - a^t B^{-1} a > 0$ ce qui montre que (1.41) est vérifiée. On peut ainsi choisir $\lambda = \sqrt{\alpha - a^t B^{-1} a}$ (> 0) de telle sorte que (1.40) est vérifiée. Posons :

$$L = \left[\begin{array}{c|c} M & 0 \\ \hline (M^{-1}a)^t & \lambda \end{array} \right].$$

La matrice L est bien triangulaire inférieure et vérifie $\ell_{i,i} > 0$ et $A = LL^t$.

On a terminé ainsi la partie “existence”.

II- Unicité et calcul de L . Soit $A \in \mathcal{M}_n(\mathbb{R})$ symétrique définie positive ; on vient de montrer qu’il existe $L \in \mathcal{M}_n(\mathbb{R})$ triangulaire inférieure telle que $\ell_{i,j} = 0$ si $j > i$, $\ell_{i,i} > 0$ et $A = LL^t$. On a donc :

$$a_{i,j} = \sum_{k=1}^n \ell_{i,k} \ell_{j,k}, \quad \forall (i,j) \in \{1 \dots n\}^2. \quad (1.42)$$

1. Calculons la 1-ère colonne de L ; pour $j = 1$, on a :

$$\begin{aligned} a_{1,1} &= \ell_{1,1} \ell_{1,1} \text{ donc } \ell_{1,1} = \sqrt{a_{1,1}} \quad (a_{1,1} > 0 \text{ car } \ell_{1,1} \text{ existe}), \\ a_{2,1} &= \ell_{2,1} \ell_{1,1} \text{ donc } \ell_{2,1} = \frac{a_{2,1}}{\ell_{1,1}}, \\ a_{i,1} &= \ell_{i,1} \ell_{1,1} \text{ donc } \ell_{i,1} = \frac{a_{i,1}}{\ell_{1,1}} \quad \forall i \in \{2, \dots, n\}. \end{aligned}$$

2. On suppose avoir calculé les q premières colonnes de L . On calcule la colonne $(q+1)$ en prenant $j = q+1$ dans (1.42)

$$\begin{aligned} \text{Pour } i = q+1, a_{q+1,q+1} &= \sum_{k=1}^{q+1} \ell_{q+1,k} \ell_{q+1,k} \text{ donc} \\ \ell_{q+1,q+1} &= (a_{q+1,q+1} - \sum_{k=1}^q \ell_{q+1,k}^2)^{1/2} > 0. \end{aligned} \quad (1.43)$$

Notons que $a_{q+1,q+1} - \sum_{k=1}^q \ell_{q+1,k}^2 > 0$ car L existe : il est indispensable d’avoir d’abord montré l’existence de L pour pouvoir exhiber le coefficient $\ell_{q+1,q+1}$.

On procède de la même manière pour $i = q+2, \dots, n$; on a :

$$a_{i,q+1} = \sum_{k=1}^{q+1} \ell_{i,k} \ell_{q+1,k} = \sum_{k=1}^q \ell_{i,k} \ell_{q+1,k} + \ell_{i,q+1} \ell_{q+1,q+1}$$

et donc

$$\ell_{i,q+1} = \left(a_{i,q+1} - \sum_{k=1}^q \ell_{i,k} \ell_{q+1,k} \right) \frac{1}{\ell_{q+1,q+1}}. \quad (1.44)$$

On calcule ainsi toutes les colonnes de L . On a donc montré que L est unique par un moyen constructif de calcul de L .

■

Remarque 1.22 (Choleski et LU). *Considérons une matrice A symétrique définie positive. Alors une matrice P de permutation dans le théorème 1.21 possible n’est autre que l’identité. Il suffit pour s’en convaincre de remarquer qu’une fois qu’on s’est donné la bijection $t = \text{Id}$ dans l’algorithme 1.17, celle-ci n’est jamais modifiée et donc on a $P = \text{Id}$. Les théorèmes d’existence et d’unicité 1.19 et 1.21 nous permettent alors de remarquer que $A = LU = \tilde{L}\tilde{L}^t$ avec $\tilde{L} = L\sqrt{D}$, où D est la matrice diagonale extraite de U , et \sqrt{D} désigne la matrice dont les coefficients sont les racines carrées des coefficients de D (qui sont tous positifs). Voir à ce sujet l’exercice 28 page 50.*

La décomposition LU permet de caractériser les matrices symétriques définies positives.

Proposition 1.23 (Caractérisation des matrices symétriques définies positives par la décomposition LU). *Soit A une matrice symétrique admettant une décomposition LU sans permutation, c'est-à-dire qu'on suppose qu'il existe L triangulaire inférieure de coefficients diagonaux tous égaux à 1, et U triangulaire supérieure telle que $A = LU$. Alors A est symétrique définie positive si et seulement si tous les pivots (c'est-à-dire les coefficients diagonaux de la matrice U) sont strictement positifs.*

DÉMONSTRATION — Soit A une matrice symétrique admettant une décomposition LU sans permutation. Si A est symétrique définie positive, le théorème 1.21 de décomposition de Choleski donne immédiatement le résultat.

Montrons maintenant la réciproque : supposons que $A = LU$ avec tous les pivots strictement positifs. On a donc $A = LU$, et U est inversible car c'est une matrice triangulaire supérieure dont tous les coefficients diagonaux sont strictement positifs. Donc A est aussi inversible, et la décomposition LU est donc unique, par le théorème 1.19 de décomposition LU d'une matrice inversible. On a donc $A = LU = LD\tilde{L}^t$ où D est la matrice diagonale dont la diagonale est celle de U , et \tilde{L} est la matrice triangulaire inférieure de coefficients diagonaux tous égaux à 1 définie par $\tilde{L}^t = D^{-1}U$. On a donc aussi par symétrie de A

$$A^t = \tilde{L}DL^t = A = LU$$

et par unicité de la décomposition LU , on en déduit que $\tilde{L} = L$ et $DL^t = U$, ce qui entraîne que $A = LDL^t = CC^t$ avec $C = L\sqrt{D}$. On a donc pour tout $x \in \mathbb{R}^n$, $Ax \cdot x = CC^t x \cdot x = \|Cx\|^2$ et donc que A est symétrique définie positive. ■

Attention : la proposition précédente est fautive si la décomposition est avec permutation, méditer pour s'en convaincre l'exemple $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

Remarque 1.24 (Pivot partiel et Choleski.). *Considérons une matrice A symétrique définie positive. On a vu dans le théorème qu'on n'a pas besoin de permutation pour obtenir la décomposition LL^t d'une matrice symétrique définie positive. Par contre, on utilise malgré tout la technique de pivot partiel pour minimiser les erreurs d'arrondi. On peut illustrer cette raison par l'exemple suivant :*

$$A = \begin{bmatrix} -10^{-n} & 1 \\ 1 & 1 \end{bmatrix}$$

À titre d'illustration, pour $n = 12$ en FORTRAN (double précision), on obtient la bonne solution, c.à.d. $(-1, 1)$, avec le programme `gausslupivot` donné plus haut, alors que le programme sans pivot `gausslu` donne comme solution $(0, 1)$.

Calcul du coût de la méthode de Choleski

Calcul du coût de calcul de la matrice L Dans le procédé de calcul de L exposé ci-dessus, le nombre d'opérations pour calculer la première colonne est n . Calculons, pour $p = 0, \dots, n-1$, le nombre d'opérations pour calculer la $(p+1)$ -ième colonne : pour la colonne $(p+1)$, le nombre d'opérations par ligne est $2p+1$, car le calcul de $\ell_{p+1,p+1}$ par la formule (1.43) nécessite p multiplications, p soustractions et une extraction de racine, soit $2p+1$ opérations ; le calcul de $\ell_{i,p+1}$ par la formule (1.44) nécessite p multiplications, p soustractions et une division, soit encore $2p+1$ opérations. Comme les calculs se font des lignes $p+1$ à n (car $\ell_{i,p+1} = 0$ pour $i \leq p$), le nombre d'opérations pour calculer la $(p+1)$ -ième colonne est donc $(2p+1)(n-p)$. On en déduit que le nombre d'opérations N_L nécessaires au calcul de L est :

$$\begin{aligned} N_L &= \sum_{p=0}^{n-1} (2p+1)(n-p) = 2n \sum_{p=0}^{n-1} p - 2 \sum_{p=0}^{n-1} p^2 + n \sum_{p=0}^{n-1} 1 - \sum_{p=0}^{n-1} p \\ &= (2n-1) \frac{n(n-1)}{2} + n^2 - 2 \sum_{p=0}^{n-1} p^2. \end{aligned}$$

(On rappelle que $2 \sum_{p=0}^{n-1} p = n(n-1)$.) Il reste à calculer $C_n = \sum_{p=0}^n p^2$, en remarquant par exemple que

$$\begin{aligned} \sum_{p=0}^n (1+p)^3 &= \sum_{p=0}^n 1 + p^3 + 3p^2 + 3p = \sum_{p=0}^n 1 + \sum_{p=0}^n p^3 + 3 \sum_{p=0}^n p^2 + 3 \sum_{p=0}^n p \\ &= \sum_{p=1}^{n+1} p^3 = \sum_{p=0}^n p^3 + (n+1)^3. \end{aligned}$$

On a donc $3C_n + 3 \frac{n(n+1)}{2} + n + 1 = (n+1)^3$, d'où on déduit que

$$C_n = \frac{n(n+1)(2n+1)}{6}.$$

On a donc :

$$\begin{aligned} N_L &= (2n-1) \frac{n(n-1)}{2} - 2C_{n-1} + n^2 \\ &= n \left(\frac{2n^2 + 3n + 1}{6} \right) = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6} = \frac{n^3}{3} + 0(n^2). \end{aligned}$$

Coût de la résolution d'un système linéaire par la méthode LL^t Nous pouvons maintenant calculer le coût (en termes de nombre d'opérations élémentaires) nécessaire à la résolution de (1.1) par la méthode de Choleski pour $A \in \mathcal{M}_n(\mathbb{R})$ symétrique définie positive. On a besoin de N_L opérations pour le calcul de L , auquel il faut rajouter le nombre d'opérations nécessaires pour les étapes de descente et remontée. Le calcul de y solution de $Ly = b$ s'effectue en résolvant le système :

$$\begin{bmatrix} \ell_{1,1} & & 0 \\ \vdots & \ddots & \vdots \\ \ell_{n,1} & \dots & \ell_{n,n} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Pour la ligne 1, le calcul $y_1 = \frac{b_1}{\ell_{1,1}}$ s'effectue en une opération.

Pour les lignes $p = 2$ à n , le calcul $y_p = \left(b_p - \sum_{i=1}^{p-1} \ell_{i,p} y_i \right) / \ell_{p,p}$ s'effectue en $(p-1)$ (multiplications) + $(p-2)$ (additions) + 1 soustraction + 1 (division) = $2p-1$ opérations. Le calcul de y (descente) s'effectue donc en $N_1 = \sum_{p=1}^n (2p-1) = n(n+1) - n = n^2$. On peut calculer de manière similaire le nombre d'opérations nécessaires pour l'étape de remontée $N_2 = n^2$. Le nombre total d'opérations pour calculer x solution de (1.1) par la méthode de Choleski est $N_C = N_L + N_1 + N_2 = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6} + 2n^2 = \frac{n^3}{3} + \frac{5n^2}{2} + \frac{n}{6}$. L'étape la plus coûteuse est donc la factorisation de A .

Remarque 1.25 (Décomposition LDL^t). Dans les programmes informatiques, on préfère implanter la variante suivante de la décomposition de Choleski : $A = \tilde{L} D \tilde{L}^t$ où D est la matrice diagonale définie par $d_{i,i} = \ell_{i,i}^2$, $\tilde{L}_{i,i} = L \tilde{D}^{-1}$, où \tilde{D} est la matrice diagonale définie par $d_{i,i} = \ell_{i,i}$. Cette décomposition a l'avantage de ne pas faire intervenir le calcul de racines carrées, qui est une opération plus compliquée que les opérations "élémentaires" (\times , $+$, $-$).

1.3.4 Quelques propriétés

Comparaison Gauss/Choleski

Soit $A \in \mathcal{M}_n(\mathbb{R})$ inversible, la résolution de (1.1) par la méthode de Gauss demande $2n^3/3 + 0(n^2)$ opérations (exercice). Dans le cas d'une matrice symétrique définie positive, la méthode de Choleski est donc environ deux fois moins chère.

Et la méthode de Cramer ?

Soit $A \in \mathcal{M}_n(\mathbb{R})$ inversible. On rappelle que la méthode de Cramer pour la résolution de (1.1) consiste à calculer les composantes de x par les formules :

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad i = 1, \dots, n,$$

où A_i est la matrice carrée d'ordre n obtenue à partir de A en remplaçant la i -ème colonne de A par le vecteur b , et $\det(A)$ désigne le déterminant de A .

Le calcul du déterminant d'une matrice carrée d'ordre n en utilisant les formules "usuelles" (c'est-à-dire en développant par rapport à une ligne ou une colonne) nécessite au moins $n!$ opérations (voir cours L1-L2, ou livres d'algèbre linéaire proposés en avant-propos). Par exemple, pour $n = 10$, la méthode de Gauss nécessite environ 700 opérations, la méthode de Choleski environ 350 et la méthode de Cramer (avec les formules usuelles de calcul du déterminant) plus de 4 000 000. . . Cette dernière méthode est donc à proscrire.

Conservation du profil de A

Dans de nombreuses applications, par exemple lors de la résolution de systèmes linéaires issus de la discrétisation⁶ de problèmes réels, la matrice $A \in \mathcal{M}_n(\mathbb{R})$ est "creuse", au sens où un grand nombre de ses coefficients sont nuls. Il est intéressant dans ce cas pour des raisons d'économie de mémoire de connaître le "profil" de la matrice, donné dans le cas où la matrice est symétrique, par les indices $j_i = \min\{j \in \{1, \dots, n\} \text{ tels que } a_{i,j} \neq 0\}$. Le profil de la matrice est donc déterminé par les diagonales contenant des coefficients non nuls qui sont les plus éloignées de la diagonale principale. Dans le cas d'une matrice creuse, il est avantageux de faire un stockage "profil" de A , en stockant, pour chaque ligne i la valeur de j_i et des coefficients $a_{i,k}$, pour $k = i - j_i, \dots, i$, ce qui peut permettre un large gain de place mémoire.

Une propriété intéressante de la méthode de Choleski est de conserver le profil. On peut montrer (en reprenant les calculs effectués dans la deuxième partie de la démonstration du théorème 1.21) que $\ell_{i,j} = 0$ si $j < j_i$. Donc si on a adopté un stockage "profil" de A , on peut utiliser le même stockage pour L .

Matrices non symétriques

Soit $A \in \mathcal{M}_n(\mathbb{R})$ inversible. On ne suppose plus ici que A est symétrique. On cherche à calculer $x \in \mathbb{R}^n$ solution de (1.1) par la méthode de Choleski. Ceci est possible en remarquant que : $Ax = b \Leftrightarrow A^t Ax = A^t b$ car $\det(A) = \det(A^t) \neq 0$. Il ne reste alors plus qu'à vérifier que $A^t A$ est symétrique définie positive. Remarquons d'abord que pour toute matrice $A \in \mathcal{M}_n(\mathbb{R})$, la matrice AA^t est symétrique. Pour cela on utilise le fait que si $B \in \mathcal{M}_n(\mathbb{R})$, alors B est symétrique si et seulement si $Bx \cdot y = x \cdot By$ et $Bx \cdot y = x \cdot B^t y$ pour tout $(x, y) \in (\mathbb{R}^n)^2$. En prenant $B = A^t A$, on en déduit que $A^t A$ est symétrique. De plus, comme A est inversible, $A^t Ax \cdot x = Ax \cdot Ax = |Ax|^2 > 0$ si $x \neq 0$. La matrice $A^t A$ est donc bien symétrique définie positive.

La méthode de Choleski dans le cas d'une matrice non symétrique consiste donc à calculer $A^t A$ et $A^t b$, puis à résoudre le système linéaire $A^t A \cdot x = A^t b$ par la méthode de Choleski "symétrique".

Cette manière de faire est plutôt moins efficace que la décomposition LU puisque le coût de la décomposition LU est de $2n^3/3$ alors que la méthode de Choleski dans le cas d'une matrice non symétrique nécessite au moins $4n^3/3$ opérations (voir exercice 23).

Systèmes linéaires non carrés

On considère ici des matrices qui ne sont plus carrées. On désigne par $\mathcal{M}_{M,n}(\mathbb{R})$ l'ensemble des matrices réelles à M lignes et n colonnes. Pour $A \in \mathcal{M}_{M,n}(\mathbb{R})$, $M > n$ et $b \in \mathbb{R}^M$, on cherche $x \in \mathbb{R}^n$ tel que

$$Ax = b. \tag{1.45}$$

6. On appelle discrétisation le fait de se ramener d'un problème où l'inconnue est une fonction en un problème ayant un nombre fini d'inconnues scalaires.

Ce système contient plus d'équations que d'inconnues et n'admet donc en général pas de solution. On cherche $x \in \mathbb{R}^n$ qui vérifie le système (1.45) "au mieux". On introduit pour cela une fonction f définie de \mathbb{R}^n dans \mathbb{R} par :

$$f(x) = |Ax - b|^2,$$

où $|x| = \sqrt{x \cdot x}$ désigne la norme euclidienne sur \mathbb{R}^n . La fonction f ainsi définie est évidemment positive, et s'il existe x qui annule f , alors x est solution du système (1.45). Comme on l'a dit, un tel x n'existe pas forcément, et on cherche alors un vecteur x qui vérifie (1.45) "au mieux", au sens où $f(x)$ soit le plus proche de 0. On cherche donc $x \in \mathbb{R}^n$ satisfaisant (1.45) en minimisant f , c.à.d. en cherchant $x \in \mathbb{R}^n$ solution du problème d'optimisation :

$$f(x) \leq f(y) \quad \forall y \in \mathbb{R}^n \quad (1.46)$$

On peut réécrire f sous la forme : $f(x) = A^t A x \cdot x - 2b \cdot Ax + b \cdot b$. On montrera au chapitre III que s'il existe une solution au problème (1.46), elle est donnée par la résolution du système linéaire suivant :

$$AA^t x = A^t b \in \mathbb{R}^n, \quad (1.47)$$

qu'on appelle équations normales du problème de minimisation. La résolution approchée du problème (1.45) par cette procédure est appelée méthode des moindres carrés. La matrice AA^t étant symétrique, on peut alors employer la méthode de Choleski pour la résolution du système (1.47).