Analyse numérique

Résolution de systèmes linéaires
Méthodes directes

Alexandre Caboussat

Système linéaire

Système de N équations à N inconnues $x_1, x_2, \ldots, x_N$:

$$\left\{\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2N}x_N &= b_2, \\ &\vdots \\ a_{N1}x_1 + a_{N2}x_2 + \cdots + a_{NN}x_N &= b_N. \end{aligned}\right.$$

Définition du problème

$$A\vec{x} = \vec{b}.$$

avec

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}, \qquad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}, \qquad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}.$$

Le problème admet une solution si A est inversible!

Rappels : matrices

Matrices triangulaires

Définition

La matrice A est triangulaire supérieure (respectivement triangulaire inférieure) si $a_{ij} = 0$ pour i, j tel que $1 \leq j < i \leq N$ (resp. $1 \leq i < j \leq N$).

Définition

Si A est une matrice triangulaire supérieure (resp. triangulaire inférieure), le système linéaire est un système triangulaire supérieur (resp. triangulaire inférieur).

Résolution d'un système triangulaire

Supposons un instant que la matrice A soit triangulaire supérieure.

$$\det A = \prod_{i=1}^{N} a_{ii} \; \neq 0 \qquad \text{si } A \text{ est supposée régulière}$$

Donc on peut supposer:

$$a_{ii} \neq 0, \quad \forall i = 1, \ldots, N.$$

(Dans certains livres on suppose même $a_{ii} = 1$, $i = 1, 2, \ldots, N$.)

Algorithme

On calcule successivement les inconnues $x_N, x_{N-1}, \ldots, x_1$. En effet :

$$x_N = \frac{1}{a_{NN}} b_N$$

et pour $i = N-1, N-2, \ldots, 3, 2, 1$:

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^{N} a_{ij} x_j \right)$$

Comment transformer un système linéaire $A\vec{x} = \vec{b}$ en un système linéaire triangulaire équivalent? La méthode d'élimination de Gauss.

Méthode d'élimination de Gauss

Exemple avec $N = 3$

$$A = \begin{bmatrix} 4 & 8 & 12 \\ 3 & 8 & 13 \\ 2 & 9 & 18 \end{bmatrix}, \qquad \vec{b} = \begin{bmatrix} 4 \\ 5 \\ 11 \end{bmatrix}.$$

Algorithme (1)

Première étape Diviser la première équation du système par $a_{11} = 4$ (premier pivot) :

$$\left\{ \begin{array}{rcl} x_1 + 2x_2 + 3x_3 & = & 1 \\ 3x_1 + 8x_2 + 13x_3 & = & 5 \\ 2x_1 + 9x_2 + 18x_3 & = & 11 \end{array} \right.$$

Soustraire $3\times$ la première équation à la deuxième équation, et $2\times$ fois la première équation à la troisième équation:

$$\left\{ \begin{array}{rcl} x_1 + 2x_2 + 3x_3 & = & 1 \\ 2x_2 + 4x_3 & = & 2 \\ 5x_2 + 12x_3 & = & 9 \end{array} \right.$$

Algorithme (2)

Deuxième étape Diviser la première équation du système par 2 (deuxième pivot) :

$$\left\{\begin{array}{rcl} x_1 + 2x_2 + 3x_3 & = & 1 \\ x_2 + 2x_3 & = & 1 \\ 5x_2 + 12x_3 & = & 9 \end{array}\right.$$

Soustraire $5\times$ la deuxième équation à la troisième équation:

$$\left\{\begin{array}{rcl} x_1 + 2x_2 + 3x_3 & = & 1 \\ x_2 + 2x_3 & = & 1 \\ 2x_3 & = & 4 \end{array}\right.$$

Algorithme (3)

Dernière étape Diviser la troisième équation du système par 2 (troisième pivot) :

$$\left\{\begin{array}{rcl} x_1 + 2x_2 + 3x_3 & = & 1 \\ x_2 + 2x_3 & = & 1 \\ x_3 & = & 2 \end{array}\right.$$

Système triangulaire équivalent!

Résoudre dans l'ordre inverse:

$$x_3 = 2, \qquad x_2 = -3, \qquad x_1 = 1.$$

Algorithme général (i-ème étape) (1)

Soient $A^{(i)}$ la matrice et $\vec{b}^{(i)}$ le second membre obtenus avant la i-ième étape de l'élimination.

$$A^{(i)} = \begin{bmatrix} 1 & a_{12}^{(i)} & a_{13}^{(i)} & a_{14}^{(i)} & \cdots & & a_{1i}^{(i)} & \cdots & a_{1N}^{(i)} \\ 0 & 1 & a_{23}^{(i)} & a_{24}^{(i)} & \cdots & & a_{2i}^{(i)} & \cdots & a_{2N}^{(i)} \\ 0 & 0 & 1 & a_{34}^{(i)} & \cdots & & a_{3i}^{(i)} & \cdots & a_{3N}^{(i)} \\ 0 & 0 & 0 & 1 & \cdots & & a_{4i}^{(i)} & \cdots & a_{4N}^{(i)} \\ & & & & \ddots & & \vdots & & \vdots \\ & & & & & 1 & & & \\ & & 0 & & & & a_{ii}^{(i)} & \cdots & a_{iN}^{(i)} \\ & & & & & & \vdots & & \vdots \\ & & & & & & a_{Ni}^{(i)} & \cdots & a_{NN}^{(i)} \end{bmatrix}.$$

Algorithme général (i-ème étape) (2)

$$(A^{(i)}, \vec{b}^{(i)}) \longmapsto (A^{(i+1)}, \vec{b}^{(i+1)})$$

Normalisation i-ième ligne:

$$a_{ij}^{(i+1)} = \frac{a_{ij}^{(i)}}{a_{ii}^{(i)}}, \qquad j = i+1, i+2, \ldots, N \quad \text{et} \quad b_i^{(i+1)} = \frac{b_i^{(i)}}{a_{ii}^{(i)}}$$

Elimination : Soustraire $a_{ki}^{(i)} \times$ la i-ième ligne de la k-ième ligne (de $A^{(i)}$ et $b_k^{(i)}$):

$$a_{kj}^{(i+1)} = a_{kj}^{(i)} - a_{ki}^{(i)} \times a_{ij}^{(i+1)}, \qquad j = i+1, i+2, \ldots, N \quad \text{et} \quad b_k^{(i+1)} = b_k^{(i)} - a_{ki}^{(i)} \times b_i^{(i+1)}$$

Remarque (importante!)

Apès la i-ème étape, les nombres $a_{kj}^{(i)}$, $i+1 \leq k, j \leq N$, ne sont plus utiles. On peut donc économiser de la place mémoire en écrivant :

$$a_{ij} := \frac{a_{ij}}{a_{ii}}, \qquad j = i+1, i+2, \ldots, N,$$

(idem pour les autres relations)

Algorithme d'élimination de Gauss

entrées : a_{ij}, $1 \leq i, j \leq N$ et b_j, $1 \leq j \leq N$ représentant la matrice A et le second membre $\vec{b}$ du système linéaire originel **sorties** : a_{ij}, $1 \leq i < j \leq N$ et b_j, $1 \leq j \leq N$ représentant la partie surdiagonale du système triangulaire supérieur et le nouveau second membre $\vec{b}$	
Algorithme	**Commentaires**
Faire $i = 1$ à $N-1$	Elimination de l'inconnue x_i
$p := 1/a_{ii}$	Inverse du i-ième pivot
Faire $j = i+1$ à N $a_{ij} := p \times a_{ij}$	Division de la i-ième ligne par le i-ième pivot (termes surdiagonaux uniquement)
$b_i := p \times b_i$	Division de b_i par le i-ième pivot
Faire $k = i+1$ à N	Elimination dans la k-ième équation
Faire $j = i+1$ à N $a_{kj} := a_{kj} - a_{ki} \times a_{ij}$	Soustraction de a_{ki} fois la nouvelle i-ième ligne à la k-ième ligne
$b_k := b_k - a_{ki} \times b_i$	Soustraction de a_{ki} fois b_i à b_k
$p := 1/a_{NN}$	Inverse du N-ième pivot
$b_N := p \times b_N$	Division de b_N par le N-ième pivot

Systèmes mal conditionnés

Exemple

$$\begin{cases} 4.218613x_1 & + & 6.327917x_2 & = & 10.546530 \\ 3.141592x_1 & + & 4.712390x_2 & = & 7.853982 \end{cases}$$

Nous vérifions aisément que ce système est régulier (déterminant de A non nul) et que la solution est donnée par $x_1 = x_2 = 1$. Considérons maintenant un système d'équations voisin:

$$\begin{cases} 4.218611\ x_1 & + & 6.327917\ x_2 & = & 10.546530 \\ 3.141594\ x_1 & + & 4.712390\ x_2 & = & 7.853980 \end{cases}$$

Nous vérifions encore que ce système est régulier, mais cette fois la solution est donnée par $x_1 = -5$, et $x_2 = +5$.

Interprétation

Le système est formé de deux équations qui, dans le plan Ox_1, x_2, décrivent deux droites presque parallèles. Si on perturbe un tout petit peu deux droites presque parallèles, alors le point d'intersection est fortement modifié !

Résoudre un problème mal conditionné avec un ordinateur peut être une affaire délicate si l'ordinateur calcule avec trop peu de chiffres significatifs.

Décomposition LU

Théorème

Soit A une $N \times N$ matrice dont toutes les sous-matrices principales sont régulières, alors

$$A = LU$$

où L est une matrice triangulaire inférieure et U est une matrice triangulaire supérieure avec des valeurs 1 dans sa diagonale.
La décomposition est unique.
La matrice U est celle obtenue par l'algorithme d'élimination de Gauss.

Construction de l'algorithme : exemple

Soit une matrice A composée de 3 lignes et 3 colonnes.

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}}_{U}.$$

Algorithme de décomposition LU

entrées : a_{ij}, $1 \leq i, j \leq N$ sont les coefficients de la matrice A

sorties : a_{ij}, $1 \leq j \leq i \leq N$ sont les coefficients de la matrice L

et a_{ij}, $1 \leq i < j < N$ sont les coefficients surdiagonaux de la matrice U

Algorithme	**Commentaires**
Faire $i = 2$ à N $a_{1i} := a_{1i}/a_{11}$	Construction de la première ligne de U (la $1^{\text{ère}}$ col. de L est la $1^{\text{ère}}$ col. de A)
Faire $k = 2$ à $N-1$	Colonnes de L et lignes de U
$a_{kk} := a_{kk} - \sum_{j=1}^{k-1} a_{kj} * a_{jk}$	Construction du pivot ℓ_{kk}
Faire $i = k+1$ à N	
$a_{ik} := a_{ik} - \sum_{j=1}^{k-1} a_{ij} * a_{jk}$	Construction de la k-ième colonne de L
$a_{ki} := \dfrac{1}{a_{kk}}\left(a_{ki} - \sum_{j=1}^{k-1} a_{kj} * a_{ji}\right)$	Construction de la k-ième ligne de U
$a_{NN} := a_{NN} - \sum_{j=1}^{N-1} a_{Nj} * a_{jN}$	Construction du pivot ℓ_{NN}

Nombre d’opérations

Comme l’algorithme d’élimination de Gauss, le nombre d’opérations de l’algorithme de décomposition LU se comporte comme N^3 lorsque N est grand.

Plusieurs systèmes linéaires?

Nous voulons résoudre:

$$A\vec{x}^{(\ell)} = \vec{b}^{(\ell)}, \qquad \ell = 1, 2, \ldots, m,$$

où, par exemple $\vec{b}^{(\ell)}$ dépend de $\vec{x}^{(\ell-1)}$.

Définition

Une $N \times N$ matrice A est dite symétrique définie positive si :

i) $A = A^T$ (A est symétrique),

ii) $\vec{y}^T A \vec{y} \geq 0$ pour tout N-vecteur $\vec{y}$,

iii) $\vec{y}^T A \vec{y} = 0$ si et seulement si $\vec{y} = 0$.

Si A est une $N \times N$ matrice symétrique définie positive, alors elle est régulière.

Résultat

Si A est une $N \times N$ matrice symétrique définie positive, alors toutes ses sous-matrices principales sont symétriques définies positives et sont donc régulières.

Démonstration:
Soit A_k la sous-matrice principale d'ordre k de A (A_k est symétrique). En considérant:

$$\vec{y} = \begin{bmatrix} \vec{z} \\ 0 \end{bmatrix} \begin{array}{l} \} \, k \text{ premières composantes} \\ \} \, (N-k) \text{ composantes nulles,} \end{array}$$

On a

$$\vec{z}^T A_k \vec{z} = \vec{y}^T A \vec{y}.$$

On déduit que A_k est symétrique définie positive.

Théorème

Théorème de Cholesky

Si A est une matrice symétrique définie positive, il existe une et une seule matrice triangulaire inférieure à valeurs diagonales positives notée L telle que

$$A = LL^T.$$

Construction de l'algorithme : exemple

Soit une matrice A composée de 3 lignes et 3 colonnes.

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} \ell_{11} & \ell_{21} & \ell_{31} \\ 0 & \ell_{22} & \ell_{32} \\ 0 & 0 & \ell_{33} \end{bmatrix}}_{L^T}.$$

Algorithme de Cholesky

entrées : $(a_{ij})_{1\le j\le i\le N}$ représentant la partie triangulaire inférieure de A;
(A est symétrique définie positive)
sorties : $(a_{ij})_{1\le j\le i\le N}$ représentant L qui satisfait $A = LL^T$

Algorithme	Commentaires
$a_{11} := \sqrt{a_{11}}$	Construction de ℓ_{11}
Faire $i = 2$ à N $\quad a_{i1} := a_{i1}/a_{11}$	Construction de la première colonne de L
Faire $k = 2$ à $N-1$	Parcours des colonnes de L
$\quad a_{kk} := \left(a_{kk} - \sum_{j=1}^{k-1} a_{kj}^2\right)^{1/2}$	Construction de ℓ_{kk}
$\quad$ Faire $i = k+1$ à N $\quad\quad a_{ik} := \frac{1}{a_{kk}}\left(a_{ik} - \sum_{j=1}^{k-1} a_{ij} * a_{kj}\right)$	Construction de la k-ième col. de L
$a_{NN} := \left(a_{NN} - \sum_{j=1}^{N-1} a_{Nj}^2\right)^{1/2}$	Construction de ℓ_{NN}

Définition

Soit A une $N \times N$ matrice de coefficients a_{ij}, $1 \leq i, j \leq N$ et soit ℓ un entier positif inférieur à N. On dira que A est une matrice de bande de demi-largeur ℓ si on a $a_{ij} = 0$ pour tout i, j satisfaisant $1 \leq i, j \leq N$ et $\mid i - j \mid \geq \ell$.

Résultat

La décomposition LU (ou LL^T) de A donne lieu à des matrices triangulaires qui sont aussi de bande de demi-largeur ℓ. Le nombre d'opérations pour faire la décomposition LU ou LL^T est de l'ordre de $N\ell^2$ lorsque N est grand.

$$A = \begin{bmatrix} * & * & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & * & * & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & * & * & * & 0 & 0 & 0 & 0 \\ * & * & * & * & * & * & * & * & 0 & 0 & 0 \\ * & * & * & * & * & * & * & * & * & 0 & 0 \\ 0 & * & * & * & * & * & * & * & * & * & 0 \\ 0 & 0 & * & * & * & * & * & * & * & * & * \\ 0 & 0 & 0 & * & * & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & * & * & * & * & * \end{bmatrix}$$

Exemple (classique)

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}.$$

$$L = \begin{bmatrix} d_1 & & & & \\ e_1 & d_2 & & & \\ & \ddots & \ddots & & \\ & & \ddots & d_{N-1} & \\ & & & e_{N-1} & d_N \end{bmatrix}.$$

Exemple (classique) (2)

$$L = \begin{bmatrix} d_1 & & & & \\ e_1 & d_2 & & & \\ & \ddots & \ddots & & \\ & & \ddots & d_{N-1} & \\ & & & e_{N-1} & d_N \end{bmatrix}.$$

Exemple (classique) (3)

Dans ce cas l'algorithme de décomposition LL^T de A se met sous la forme suivante :

$$
\begin{array}{l}
d_1 = \sqrt{2} \\
\left[\begin{array}{l}
\text{Faire } j = 1 \text{ à } N-1 \\
e_j := -1/d_j \\
d_{j+1} := \sqrt{2 - e_j^2}.
\end{array}\right.
\end{array}
$$

Application: Méthode des moindres carrés

Problème

Soit A une matrice $M \times N$ matrice avec $M > N$, et $\vec{b}$ un vecteur de taille M. On cherche à résoudre:

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

- Plus d'équations (M équations) que d'inconnues (N inconnues)
- Système surdéterminé (certaines fois sans solutions)

Trouver un N-vecteur $\vec{x}$ tel que pour tout N-vecteur $\vec{y}$ on ait :

$$\|A\vec{x} - \vec{b}\| \leq \|A\vec{y} - \vec{b}\|.$$

On dira dans ce cas que l'on cherche une solution de $A\vec{x} = \vec{b}$ au sens des moindres carrés.

Théorème

Supposons que A soit une $M \times N$ matrice ($M \geq N$) de rang N. Alors il existe un et un seul vecteur $\vec{x}$ tel que

$$\|A\vec{x} - \vec{b}\| \leq \|A\vec{y} - \vec{b}\|, \quad \forall \vec{y}.$$

De plus, $\vec{x}$ est solution de

$$A^T A\vec{x} = A^T \vec{b}.$$

Démonstration