Factorisation de Cholesky explicite

Jean-François Burnol, mars 2017

1 Factorisation de Cholesky

Théorème 1. *Soit* A *une matrice* $n \times n$ *hermitienne définie positive. Alors* A *admet la factorisation*

$$A = LL^*$$

avec L *une matrice triangulaire inférieure avec des coefficients strictement positifs sur la diagonale. Cette décomposition est donnée par la formule explicite :*

$$L = \begin{pmatrix} a(1) & & & & & & \\ \dots & a(2) & & & & & \\ \vdots & \vdots & \ddots & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ \dots & \dots & a(j-1;i,j) & \dots & a(j) & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\ \dots & \dots & \dots & \dots & \dots & \dots & a(n) \end{pmatrix} \cdot \operatorname{diag}\Big(\frac{1}{\sqrt{a(0)a(1)}}, \frac{1}{\sqrt{a(1)a(2)}}, \dots, \frac{1}{\sqrt{a(n-1)a(n)}}\Big)$$

On a noté $a(0) = 1$ *et* $a(j)$ *est le* j^e *mineur principal de* A *et, pour* $i > j$,

$$a(j-1;i,j)$$

est le mineur de A *de taille* j *formé en ne conservant que les* j *premières colonnes, les* $j-1$ *premières lignes et la* i^e *ligne.*

Preuve. C'est une conséquence immédiate du Théorème 2 de :

http://jf.burnol.free.fr/agreg170330LU.pdf

qui donne une factorisation $A = LEU$ avec des formules explicites et clairement $U = L^*$, et notre matrice diagonale est obtenue par racine carrée de la matrice diagonale positive E. □

2 Le cas de la dimension 3

$$A = \begin{pmatrix} a & u^* & v^* \\ u & b & w^* \\ v & w & c \end{pmatrix}$$

$$L = \begin{pmatrix} \sqrt{a} & 0 & 0 \\ \frac{u}{\sqrt{a}} & \sqrt{\frac{ab-uu^*}{a}} & 0 \\ \frac{v}{\sqrt{a}} & \frac{aw-vu^*}{\sqrt{a(ab-uu^*)}} & \sqrt{\frac{\det A}{ab-uu^*}} \end{pmatrix}$$

$$A = L \cdot L^*$$

3 Lien avec la décomposition LU

Compte tenu de la discussion de la décomposition LU faite dans le document mis en référence, on voit qu'il suffit de suivre l'algorithme du pivot de Gauss pour mettre A sous la forme L_1U_1 avec L_1 triangulaire inférieure avec des 1 sur la diagonale, puis de prendre de U_1 les coefficients diagonaux, d'en extraire les racines carrées (ce sont nécessairement des réels positifs) et l'on obtient L en multipliant L_1 sur sa droite par la matrice diagonale formée avec ces racines carrées. C'est-à-dire,on multiplie chaque colonne de L_1 par la racine carrée du coefficient diagonal de U_1 qui se trouve dans la même colonne. Si l'on rencontre un pivot négatif pendant le déroulement de l'algorithme on conclut que la matrice hermitienne initiale n'était pas définie positive.

On peut gagner un facteur deux car il y a (environ) deux fois moins de coefficients dans une matrice symétrique que dans une matrice générale. Considérons la première étape. Le coefficient $a_{i,j}$, pour $i > 1$ est devenu 0 dans la première colonne ($j = 1$), et, pour $j > 1$, il est devenu $b_{i,j} = a_{i,j} - \frac{a_{i,1}}{a_{1,1}}a_{1,j} = \frac{a_{i,j}a_{1,1}-a_{i,1}a_{1,j}}{a_{1,1}}$. Que vaut son conjugué complexe ? $\overline{b_{i,j}} = (a_{j,i}a_{1,1} - a_{1,i}a_{j,1})/a_{1,1} = b_{j,i}$. Donc la partie de A_1 sans la première colonne (qui a le pivot et des zéros en-dessous) et sans la première ligne (qui est celle de A) est **encore hermitienne** (ou symétrique dans le cas réel). On peut donc gagner un facteur deux en ne s'occupant lors de la soustraction de la ligne pivot que de la modification des termes sous-diagonaux. Ensuite par symétrie hermitienne on met à jour les termes sur-diagonaux.

Dans la pratique on procède un peu différemment. Revenons à la formule

$$a_{i,j} \leftarrow \frac{a_{i,j}a_{k,k} - a_{i,k}a_{k,j}}{a_{k,k}}$$

qui montre comment transformer à la k^e étape les coefficients d'indices $i > k$, $j > k$. Attention qu'ici les $a_{i,j}$ ne sont plus les coefficients d'origine de A mais ceux après les étapes précédentes.

Le coefficient diagonal de L, selon les explications qui précèdent, sera $L_{k,k} = \sqrt{a_{k,k}}$, et ceux dans la même colonne en-dessous ($i > k$) seront $L_{i,k} = a_{i,k}/\sqrt{a_{k,k}}$. Donc le nouveau coefficient en position (i, j), $i \geqslant j > k$ sera

$$a_{i,j} \leftarrow a_{i,j} - L_{i,k}\overline{L_{j,k}}$$

Ah ! il en résulte donc qu'après k étapes, les coefficients sous-diagonaux d'origine $a_{i,j}$, $i > j > k$, sont devenus [1]

$$a_{i,j} - \sum_{p=1}^{k} L_{i,p}\overline{L_{j,p}}$$

ce qui permet, d'une part de trouver

$$L_{k+1,k+1} = \sqrt{a_{k+1,k+1} - \sum_{p=1}^{k} |L_{k+1,p}|^2}$$

puis d'affirmer que

$$i > k+1 \implies L_{i,k+1} = \frac{a_{i,k+1} - \sum_{p=1}^{k} L_{i,p}\overline{L_{k+1,p}}}{L_{k+1,k+1}}$$

Ces formules permettent donc de construire la matrice triangulaire inférieure à coeffecients diagonaux positifs L avec $A = LL^*$. Ce sont elles que l'on associe habituellement à l'algorithme de Cholesky.

On justifie souvent ces formules par une autre approche :
https://fr.wikipedia.org/wiki/Factorisation_de_Cholesky

1. Dorénavant les $a_{i,j}$ sont bien les coefficients d'origine de A. Hmm, j'espère être clair...

Mais la seule différence réelle avec la procédure LU décrite précédemment est le fait d'aboutir d'un seul coup à $L_{i,j}$ en soustrayant à $a_{i,j}$ une somme de produits de coefficients déjà connus de L. Alors que dans LU on y arrive en faisant les soustractions les unes après les autres au fil des étapes. Mais il n'y aucun différence réelle à part cela. Ce ne sont pas les formules de Cholesky qui apportent un gain, mais le facteur deux déjà expliqué, car il y a (un peu plus de) deux fois moins de coefficients à mettre à jour à chaque étape.

En ce qui concerne la stabilité numérique, il y a les mêmes problèmes qu'avec LU pour les mêmes raisons : des mineurs principaux qui peuvent être très proches de zéro.