Factorizaciones de Cholesky, matrices definidas y semidefinidas positivas

Héctor Manuel Mora Escobar
Universidad Central, Bogotá
hectormora@yahoo.com

Junio de 2011

1 Introducción

Este documento presenta, con ejemplos, varios algoritmos ampliamente conocidos para la factorización de Cholesky incluyendo la adaptación para matrices semidefinidas positivas.

La factorización usual de Cholesky se puede usar únicamente para matrices definidas positivas. En optimización o en economía, se necesita saber frecuentemente si una matriz es semidefinida positiva, para aplicar condiciones de segundo orden o para averiguar si una función es convexa. Teóricamente esto se puede hacer mediante el cálculo de los valores propios, pero, numéricamente, este proceso es mucho más demorado que la factorización de Cholesky.

2 Definiciones y notación

Existen muchos artículos y textos que tratan la factorización de Cholesky, en este documento se usaron [AlK02], [Dem97], [GoV96], [Hig90], [Hig08], [Mor01], [Ste98], [TrB97].

Sea $A \in \mathbb{R}^{n\times n}$ una matriz simétrica. Se dice que A es *definida positiva* si

$$x^{\mathrm{T}}Ax > 0 \quad \text{para todo } x \text{ no nulo en } \mathbb{R}^{n\times 1}. \tag{1}$$

Se dice que A es *semidefinida positiva* si

$$x^{\mathrm{T}}Ax \geq 0 \quad \text{para todo } x \text{ en } \mathbb{R}^{n\times 1}. \tag{2}$$

Son usuales las notaciones $A \succ 0$ y $A \succcurlyeq 0$ para las matrices definidas positivas y semidefinidas postivas, respectivamente. Obviamente toda matriz definida positiva es semidefinida positiva. En este documento, mientras no se diga lo contrario, todas las matrices son reales y cuadradas de tamaño $n \times n$. Todos los vectores son vectores columna y se pueden escribir como una n-upla o explícitamente como un vector columna:

$$x = (x_1, x_2, ..., x_n) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$
$$x^{\mathrm{T}} = \begin{bmatrix} x_1 & x_2 & ... & x_n \end{bmatrix}.$$

Uma matriz simétrica tiene *factorización de Cholesky* si existe U triangular superior invertible tal que

$$A = U^{\mathrm{T}}U. \tag{3}$$

Si los elementos diagonales de U son positivos, la factorización es única. Por esto, se puede hablar de *la* factorización de Cholesky. Algunas veces se presenta la factorización de Cholesky como $A = L\,L^{\mathrm{T}}$ con L triangular inferior. Otras veces se presenta la factorización de Cholesky

$$A = V^{\mathrm{T}}D\,V$$

con D matriz diagonal invertible y V triangular superior con unos en la diagonal.

En este documento se utiliza algunas veces la siguiente notación de Matlab y Scilab, mezclada con la notación usual de matemáticas. La i-ésima fila de A se puede denotar por $A_{i\cdot}$ o por $A(i,:)$. De manera análoga para las columnas: $A_{\cdot j}$ o $A(:,j)$. $A(i:j,k:l)$ es la submatriz de A obtenida con las filas i, $i+1$, ..., j y con las columnas k, $k+1$, ..., l. Si p es un vector con enteros entre 1 y n, entonces $A(:,p)$ es la submatriz de A obtenida con las columnas cuyo índice está en p. El elemento (entrada) de A en la fila i y columna j se denotará indiferentemente por a_{ij} o por $A(i,j)$. La función `triu(A)` construye una matriz triangular superior con la parte triangular superior de A.

Sean λ_1, λ_2, ..., λ_n los valores propios de A. Si A es simétrica, todos son reales. Sea

$$\delta_i = \det(\,A(1:i,1:i)\,), \quad i = 1,...,n.$$

En particular, $\delta_1 = a_{11}$, $\delta_n = \det(A)$. Sea A simétrica. Una *permutación simétrica* de A se obtiene permutando filas de A y permutatndo de la misma manera las columnas. De manera matricial, B es una permutación simétrica de A si existe P, matriz de permutación, tal que

$$B = PAP^{\mathrm{T}}. \tag{4}$$

Aunque son conceptos completamente diferentes, aquí, un conjunto numérico finito es lo mismo que un vector cuyas entradas son justamente los elementos del conjunto.

Una matriz A es de *diagonal dominante por filas* si para cada fila i

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|. \tag{5}$$

La matriz es de *diagonal estrictamente dominante por filas* si todas las desigualdades son estrictas. Definiciones análogas se tienen por columnas. Para matrices simétricas se habla simplemente de matrices de diagonal dominante o de diagonal estrictamente dominante.

3 Algunos resultados

Existen varias caracterizaciones para las matrices definidas positivas y para las semidefinidas positivas, por medio de condiciones necesarias y suficientes, de condiciones necesarias y de condiciones suficientes. Algunas de las más conocidas son:

Proposicion 1. *Sea A simétrica. Las siguientes afirmaciones son equivalentes:*

1. $A \succ 0$.
2. *A tiene factorización de Cholesky.*
3. $\lambda_i > 0$ *para todo* i.
4. $\delta_i > 0$ *para todo* i
5. *Existe* $B \in \mathbb{R}^{q \times n}$, *de rango* n, *tal que* $A = B^T B$.

Proposicion 2. *Sea A simétrica. Las siguientes afirmaciones son equivalentes:*

1. $A \succcurlyeq 0$.
2. *Existe U triangular superior y P matriz de permutación tales que*
$$U^T U = PAP^T.$$
3. $\lambda_i \geq 0$ *para todo* i.
4. $\det(\, A(p,p)\,) \geq 0$ *para todo* $p \subseteq \{1, 2, ..., n\}$.
5. *Existe* $B \in \mathbb{R}^{q \times n}$ *tal que* $A = B^T B$.

Proposicion 3. *Si* $A \succ 0$, *entonces:*

1. $a_{ii} > 0$ *para todo* i.
2. $a_{ij}^2 < a_{ii} a_{jj}$ *para todo* $i \neq j$.
3. $2|a_{ij}| < a_{ii} + a_{jj}$ *para todo* $i \neq j$.
4. $\max\limits_i a_{ii} = \max\limits_{i,j} |a_{ij}|$.

Proposicion 4. *Si* $A \succcurlyeq 0$, *entonces:*

1. $a_{ii} \geq 0$ *para todo* i.

2. $a_{ij}^2 \leq a_{ii}a_{jj}$ *para todo* $i \neq j$.

3. $2|a_{ij}| \leq a_{ii} + a_{jj}$ *para todo* $i \neq j$.

4. $\max\limits_{i} a_{ii} = \max\limits_{i,j} |a_{ij}|$.

5. *Si* $a_{ii} = 0$, *entonces* $A(i,:) = 0$ *y* $A(:,i) = 0$.

6. *Si* $\max\limits_{i} a_{ii} = 0$, *entonces* $A = 0$.

Proposicion 5. *Si A es simétrica y de diagonal positiva y estrictamente dominante, entonces $A \succ 0$.*

4 Primera factorización

Si la factorización de Cholesky existe, entonces

$$\begin{bmatrix} u_{11} & 0 & 0 & & 0 \\ u_{12} & u_{22} & 0 & & \\ & & & & \\ u_{1i} & u_{2i} & & u_{ii} & 0 \\ & & & & \\ u_{1n} & u_{2n} & & & u_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{1i} & u_{1j} & u_{1n} \\ 0 & u_{22} & u_{2i} & u_{2j} & u_{2n} \\ & & & & \\ 0 & 0 & u_{ii} & u_{ij} & \\ & & & & \\ 0 & 0 & & & u_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & & a_{1j} & \\ & a_{22} & & a_{2j} & \\ & & & & \\ & & a_{ii} & a_{ij} & \\ & & & & \\ & & & & a_{nn} \end{bmatrix}$$

Al hacer el producto de la primera fila de U^{T} por la primera columna de U se obtiene:

$$\begin{aligned} u_{11}^2 &= a_{11} \\ u_{11} &= \sqrt{a_{11}} \end{aligned}$$

Al hacer el producto de la primera fila de U^{T} por la j-ésima columna de U se obtiene:

$$\begin{aligned} u_{11}u_{ij} &= a_{1j} \\ u_{1j} &= a_{1j}/u_{11} \end{aligned}$$

Así se han calculado todos los elementos de la primera fila de U. Al hacer el producto de la segunda fila de U^{T} por la segunda columna de U se obtiene:

$$\begin{aligned} u_{12}^2 + u_{22}^2 &= a_{22} \\ u_{22} &= \sqrt{a_{22} - u_{12}^2} \end{aligned}$$

Segunda fila de U^{T} por columna j de A:

$$\begin{aligned} u_{12}u_{1j} + u_{22}u_{2j} &= a_{2j} \\ u_{2j} &= (a_{2j} - u_{12}u_{1j})/u_{22} \end{aligned}$$

De manera general:

$$\begin{aligned} t &= a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2\,, \quad i = 1, ..., n \\ u_{ii} &= \sqrt{t} \\ u_{ij} &= \big(a_{ij} - \sum_{k=1}^{i-1} u_{ki}u_{kj}\big)/u_{ii}\,, \quad i = 1, ..., n, \quad j = i+1, ..., n \end{aligned}$$

Para ahorrar espacio, se puede reescribir sobre la matriz A lo que se vaya obteniendo de U. Así, en el arreglo A, al principio estará la matriz A y, al final del proceso, estará la matriz U.

$$t = a_{ii} - \sum_{k=1}^{i-1} a_{ki}^2\,, \quad i = 1, ..., n \tag{6}$$

$$a_{ii} = \sqrt{t} \tag{7}$$

$$a_{ij} = \big(a_{ij} - \sum_{k=1}^{i-1} a_{ki}a_{kj}\big)/a_{ii}\,, \quad j = i+1, ..., n \tag{8}$$

Tanto el cálculo de t como de a_{ij} se pueden expresar por medio del producto de partes de columnas:

$$t \leftarrow A(i,i) - A(1:i-1,i)^{\mathrm{T}} A(1:i-1,i) \tag{9}$$

$$A(i,j) \leftarrow \big(\, A(i,j) - A(1:i-1,i)^{\mathrm{T}} A(1:i-1,j) \,\big)/A(i,i) \tag{10}$$

Primera factorización de Cholesky

datos: A
$A \leftarrow \texttt{triu}(A)$
para $i = 1, ..., n$
    $t \leftarrow A(i,i) - A(1:i-1,i)^{\mathrm{T}} A(1:i-1,i)$
    si $t \leq 0$
        `info` $\leftarrow 0$
        parar
    fin-si
    $A(i,i) \leftarrow \sqrt{t}$
    para $j = i+1, ..., n$
        $A(i,j) \leftarrow \big(\, A(i,j) - A(1:i-1,i)^{\mathrm{T}} A(1:i-1,j) \,\big)/A(i,i)$
    fin-para
fin-para
`info` $\leftarrow 1$

Si al final `info` vale 1, entonces la matriz inicial A es definida positiva y, al final, en A estará la matriz U. Si al final `info` vale 0, entonces A no era definida positiva.

El número de operaciones de punto flotante de la factorización de Cholesky, la anterior o cualquiera otra realizada eficientemente, es $n^3/3$ más una cantidad proporcional a n^2.

Ejemplo 1.

```
     4    -4     6    -6
    -4    20   -22    26
     6   -22    61   -59
    -6    26   -59   108
i= 1 --------------------
```

```
t=      4
     2    -2     3    -3
     0    20   -22    26
     0     0    61   -59
     0     0     0   108
i= 2 -------------------
t=     16
     2    -2     3    -3
     0     4    -4     5
     0     0    61   -59
     0     0     0   108
i= 3 -------------------
t=     36
     2    -2     3    -3
     0     4    -4     5
     0     0     6    -5
     0     0     0   108
i= 4 -------------------
t=     49
     2    -2     3    -3
     0     4    -4     5
     0     0     6    -5
     0     0     0     7
```

5 Factorización por columnas

El cálculo de la matriz U también se puede hacer por columnas, ver [Hig08]. Supongamos calculadas las columnas 1, 2, ... $j-1$ de U. Se desea calcular la columna j de U. Sea $1 \le i \le j$.

$$
\begin{aligned}
U_{i\cdot}^{\mathrm{T}}\, U_{\cdot j} &= a_{ij} \\
\sum_{k=1}^{i} u_{ki} u_{kj} &= a_{ij} \\
\sum_{k=1}^{i-1} u_{ki} u_{kj} + u_{ii} u_{ij} &= a_{ij}
\end{aligned}
$$

$$
u_{ij} = \Big(a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj} \Big) / u_{ii}\,, \quad \text{si } i < j
$$

$$
u_{jj}^2 = a_{jj} - \sum_{k=1}^{j-1} u_{kj}^2\,, \quad \text{si } i = j
$$

Aquí también se puede reescribir la matriz U sobre la matriz A a medida que se calcula U. Las anteriores sumas se pueden expresar como productos de partes de columnas.

Factorización de Cholesky por columnas

datos: A
$A \leftarrow$ `triu`(A)
para $j = 1, ..., n$
    para $i = 1, ..., j-1$
        $A(i,j) \leftarrow \big(A(i,j) - A(1:i-1,i)^{\mathrm{T}} A(1:i-1,j) \big) / A(i,i)$
    fin-para
    $t \leftarrow A(j,j) - A(1:j-1,j)^{\mathrm{T}} A(1:j-1,j)$
    si $t \le 0$
        `info` $\leftarrow 0$
        parar
    fin-si
    $A(j,j) \leftarrow \sqrt{t}$
fin-para
`info` $\leftarrow 1$

Ejemplo 2.

```
A
     4    -4     6    -6
    -4    20   -22    26
     6   -22    61   -59
    -6    26   -59   108
triu
     4    -4     6    -6
     0    20   -22    26
     0     0    61   -59
     0     0     0   108
j= 1 --------------------
despues de calcular A(1:j-1,j)
     4    -4     6    -6
     0    20   -22    26
     0     0    61   -59
     0     0     0   108
t=      4
     2    -4     6    -6
     0    20   -22    26
     0     0    61   -59
     0     0     0   108
j= 2 --------------------
despues de calcular A(1:j-1,j)
     2    -2     6    -6
     0    20   -22    26
     0     0    61   -59
     0     0     0   108
t=     16
     2    -2     6    -6
     0     4   -22    26
     0     0    61   -59
     0     0     0   108
j= 3 --------------------
despues de calcular A(1:j-1,j)
     2    -2     3    -6
     0     4    -4    26
     0     0    61   -59
     0     0     0   108
```

```
t=     36
     2    -2     3    -6
     0     4    -4    26
     0     0     6   -59
     0     0     0   108
j= 4 --------------------
despues de calcular A(1:j-1,j)
     2    -2     3    -3
     0     4    -4     5
     0     0     6    -5
     0     0     0   108
t=     49
     2    -2     3    -3
     0     4    -4     5
     0     0     6    -5
     0     0     0     7

 U
     2    -2     3    -3
     0     4    -4     5
     0     0     6    -5
     0     0     0     7
```

6 Factorización recurrente

Sea A simétrica y definida positiva y, $A = U^{\mathrm{T}}U$, su factorización de Cholesky. La matriz A se puede descomponer en bloques, ver [GoV96], considerando la primera fila, la primera columna y el resto de la matriz. De manera análoga para U:

$$\begin{bmatrix} u_{11} & 0 \\ v & V^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} u_{11} & v^{\mathrm{T}} \\ 0 & V \end{bmatrix} = \begin{bmatrix} a_{11} & w^{\mathrm{T}} \\ w & \bar{A} \end{bmatrix}$$

Las matrices $\bar{A}$ y V están en $\mathbb{R}^{(n-1)\times(n-1)}$, $\bar{A}$ es simétrica y V es triangular superior invertible, los vectores w y v están en $\mathbb{R}^{(n-1)\times 1}$,

$$\bar{A} = \begin{bmatrix} a_{22} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n2} & \cdots & a_{nn} \end{bmatrix}$$
$$w^{\mathrm{T}} = \begin{bmatrix} w_2 & \cdots & w_n \end{bmatrix} = \begin{bmatrix} a_{12} & \cdots & a_{1n} \end{bmatrix}$$
$$v^{\mathrm{T}} = \begin{bmatrix} v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} u_{12} & \cdots & u_{1n} \end{bmatrix}$$

Al efectuar el producto se obtiene:

$$u_{11}^2 = a_{11} \tag{11}$$
$$u_{11}v^{\mathrm{T}} = w^{\mathrm{T}} \tag{12}$$
$$v\,v^{\mathrm{T}} + V^{\mathrm{T}}V = \bar{A} \tag{13}$$

De las dos primeras igualdades se obtienen las conocidas fórmulas

$$u_{11} = \sqrt{a_{11}} \tag{14}$$
$$u_{1j} = a_{1j}/u_{11}, \quad j = 2, ..., n. \tag{15}$$

De (13) se obtiene

$$V^{\mathrm{T}}V = \bar{A} - v\,v^{\mathrm{T}}.$$

Es decir, V se obtiene por medio de la factorización de Cholesky de la matriz $\bar{A}' = \bar{A} - v\,v^{\mathrm{T}}$. Como $\bar{A}'$ también es simétrica, basta con calcular los elementos de la parte triangular superior:

$$a'_{kj} = a_{kj} - v_k v_j\,, \qquad k = 2, ..., n, \;\; j = k, ..., n$$
$$a'_{kj} = a_{kj} - u_{1k}\,u_{1j}$$

Escribiendo esta última igualdad para toda la parte de la fila k se obtiene:

$$A'(k, k:n) = A(k, k:n) - U(1, k)U(1, k:n)$$

Teniendo en cuenta que se puede sobreescribir la matriz U donde estaba la matriz A y aplicando las fórmulas anteriores, inicialmente para la primera fila y después para la primera fila de la matriz restante se obtiene el siguiente algoritmo:

FACTORIZACIÓN RECURRENTE DE CHOLESKY

datos: A
$A \leftarrow \texttt{triu}(A)$
para $i = 1, ..., n$
    si $A(i,i) \leq 0$
        $\texttt{info} \leftarrow 0$
        parar
    fin-si
    $A(i,i) \leftarrow \sqrt{A(i,i)}$
    $A(i, i+1:n) \leftarrow A(i, i+1:n)/A(i,i)$
    para $k = i+1, ..., n$
        $A(k, k:n) \leftarrow A(k, k:n) - A(i,k)\,A(i, k:n)$
    fin-para
fin-para
$\texttt{info} \leftarrow 1$

Ejemplo 3.

```
A
     4    -4     6    -6
    -4    20   -22    26
     6   -22    61   -59
    -6    26   -59   108
triu
     4    -4     6    -6
     0    20   -22    26
     0     0    61   -59
     0     0     0   108
i= 1 --------------------
     2    -2     3    -3
     0    16   -16    20
     0     0    52   -50
     0     0     0    99
i= 2 --------------------
     2    -2     3    -3
     0     4    -4     5
     0     0    36   -30
```

```
     0     0     0    74
i= 3 --------------------
     2    -2     3    -3
     0     4    -4     5
     0     0     6    -5
     0     0     0    49
i= 4 --------------------
     2    -2     3    -3
     0     4    -4     5
     0     0     6    -5
     0     0     0     7
```

7 Factorización de Cholesky con pivoteo

Está basada en la factorización recurrente, ver [GoV96]. Cuando se calcula el elemento u_{ii}, éste va a ser utilizado como divisor. Entre más grande sea, menor será el error de redondeo. Así, se busca el elemento diagonal más grande, entre la posición (i,i) y la posición (n,n). Sea a_{qq} el más grande. Entonces se permutan las filas i y q y las columnas i y q, para guardar la simetría. Este intercambio simétrico no requiere operaciones de punto flotante, pero se hace basado en información que está en el arreglo `A`. Hay valores que no están de manera explícita en `A`.

Consideremos el siguiente ejemplo. La matriz muestra el resultado después del cálculo de la primera fila de U:

$$\begin{bmatrix} 2 & -1 & 1 & -2 & 3 \\ 0 & 9 & 6 & -12 & 15 \\ 0 & 0 & 5 & -9 & 14 \\ 0 & 0 & 0 & 117 & -104 \\ 0 & 0 & 0 & 0 & 109 \end{bmatrix}$$

En la primera fila del arreglo anterior está la primera fila de U. En las otras filas, está la parte triangular superior de la modificación de la submatriz 4×4 de A. La información completa de la submatriz 4×4 es:

$$\begin{bmatrix} 2 & -1 & 1 & -2 & 3 \\ 0 & 9 & 6 & -12 & 15 \\ 0 & 6 & 5 & -9 & 14 \\ 0 & -12 & -9 & 117 & -104 \\ 0 & 15 & 14 & -104 & 109 \end{bmatrix}$$

Supongamos que se desea intercambiar de manera simétrica filas y columnas 2 y 4. Primero se intercambian las filas 2 y 4. Se obtiene:

$$\begin{bmatrix} 2 & -1 & 1 & -2 & 3 \\ 0 & -12 & -9 & 117 & -104 \\ 0 & 6 & 5 & -9 & 14 \\ 0 & 9 & 6 & -12 & 15 \\ 0 & 15 & 14 & -104 & 109 \end{bmatrix}$$

La submatriz 4×4 no es simétrica. Ahora se intercambian columnas 2 y 4. El intercambio de columnas también se hace sobre la parte de U ya calculada (en este ejemplo, en la primera fila). Se obtiene:

$$\begin{bmatrix} 2 & -2 & 1 & -1 & 3 \\ 0 & 117 & -9 & -12 & -104 \\ 0 & -9 & 5 & 6 & 14 \\ 0 & -12 & 6 & 9 & 15 \\ 0 & -104 & 14 & 15 & 109 \end{bmatrix}$$

Como se trabaja con la parte triangular superior, se tendría:

$$\begin{bmatrix} 2 & -2 & 1 & -1 & 3 \\ 0 & 117 & -9 & -12 & -104 \\ 0 & 0 & 5 & 6 & 14 \\ 0 & 0 & 0 & 9 & 15 \\ 0 & 0 & 0 & 0 & 109 \end{bmatrix}$$

En resumen:

$$\begin{bmatrix} 2 & -1 & 1 & -2 & 3 \\ 0 & 9 & 6 & -12 & 15 \\ 0 & 0 & 5 & -9 & 14 \\ 0 & 0 & 0 & 117 & -104 \\ 0 & 0 & 0 & 0 & 109 \end{bmatrix} \xrightarrow{\texttt{intercSim}(A,2,4)} \begin{bmatrix} 2 & -2 & 1 & -1 & 3 \\ 0 & 117 & -9 & -12 & -104 \\ 0 & 0 & 5 & 6 & 14 \\ 0 & 0 & 0 & 9 & 15 \\ 0 & 0 & 0 & 0 & 109 \end{bmatrix}$$

En la descripción de este ejemplo y en el algoritmo completo se utiliza la función `intercSim`(A, i, q). En el arreglo A se han calculado la filas 1, 2, ..., $i-1$ de U y se hace intercambio simétrico con las filas y columnas $i < q$. Este intercambio debe hacerse de manera eficiente sin los pasos explicativos mostrados anteriormente.

La matriz U que se obtiene al final no cumple con $A = U^{\mathrm{T}}U$, pero en cambio

$$U^{\mathrm{T}}U = P\,A\,P^{\mathrm{T}} \tag{16}$$

donde P es un matriz de permutación y U es una matriz tringular superior invertible. Para reconstruir P, es necesario tener memoria de los intercambios realizados. Para esto se utiliza un vector de n enteros, que incialmente será (1, 2, ..., n).

Factorización de Cholesky con pivoteo

datos: A
$A \leftarrow$ `triu`(A)
$p = [1, 2, ..., n]$
para $i = 1 : n$
    obtener q tal que $A(q,q) = \max\{A(i,i), ..., A(n,n)\}$
    si $A(q,q) \leq 0$
        `info` $\leftarrow 0$
        parar
    fin-si
    si $i < q$
        `intercSim`(A, i, q)
        $p(i) \longleftrightarrow p(q)$
    fin-si
    $A(i,i) \leftarrow \sqrt{A(i,i)}$
    $A(i, i+1:n) \leftarrow A(i, i+1:n)/A(i,i)$
    para $k = i+1, ..., n$
        $A(k, k:n) \leftarrow A(k, k:n) - A(i,k)\,A(i, k:n)$
    fin-para
fin-para
`info` $\leftarrow 1$

Ejemplo 4.

```
A =
   400    40   -60    80
    40   104   -56    68
   -60   -56   178    54
    80    68    54   165

triu:
   400    40   -60    80
     0   104   -56    68
     0     0   178    54
     0     0     0   165

i = 1  --------------------
q = 1
a(q,q) =    400
despues de operaciones:
    20     2    -3     4
     0   100   -50    60
     0     0   169    66
     0     0     0   149

i = 2  --------------------
q = 3
a(q,q) =    169
despues de intercSim  :
    20    -3     2     4
     0   169   -50    66
     0     0   100    60
     0     0     0   149

p :       1     3     2     4

despues de operaciones:
    20    -3     2            4
     0    13    -3.846154     5.076923
     0     0    85.207101    79.526627
     0     0     0           123.224852
```

```
i = 3  --------------------
q = 4
a(q,q) =    123.224852
despues de intercSim  :
    20    -3     4            2
     0    13     5.076923    -3.846154
     0     0   123.224852    79.526627
     0     0     0           85.207101

p :       1     3     4     2

despues de operaciones:
    20    -3     4            2
     0    13     5.076923    -3.846154
     0     0    11.100669     7.164129
     0     0     0           33.882353

i = 4  --------------------
q = 4
a(q,q) =     33.882353
despues de operaciones:
    20    -3     4            2
     0    13     5.076923    -3.846154
     0     0    11.100669     7.164129
     0     0     0            5.820855


U =
    20    -3     4            2
     0    13     5.076923    -3.846154
     0     0    11.100669     7.164129
     0     0     0            5.820855

UT U =
   400   -60    80    40
   -60   178    54   -56
    80    54   165    68
    40   -56    68   104
```

```
p :      1      3      4      2

P =
     1      0      0      0
     0      0      1      0
     0      0      0      1
     0      1      0      0

P A PT =
   400    -60     80     40
   -60    178     54    -56
    80     54    165     68
    40    -56     68    104
```

8 Factorización de Cholesky para matrices semidefinidas positivas

Las factorizaciones anteriores sirven únicamente para matrices definidas positivas. Sin embargo, la factorización de Cholesky con pivoteo se puede adaptar para obtener la factorización de Cholesky normal cuando A es definida positiva o la siguiente factorización cuando A es semidefinida positiva pero no es definida positiva:

$$U^{\mathrm{T}}U = PAP^{\mathrm{T}}\,, \tag{17}$$

P es matriz de permutación,

U es triangular superior no necesariamente invertible.

En el siguiente algoritmo si al final `info` vale 1, A es definida positiva y se obtiene la factorización usual de Cholesky. Si `info` vale 0, A no es definida positiva pero es semidefinida positiva y se obtiene la factorización (17). En este caso el vector p permite reconstruir la matriz P. Si `info` vale -1, A no es semidefinida positiva.

Factorización de Cholesky para matrices semidefinidas pos.

```
datos: A
A ← triu(A)
p = [1, 2, ..., n]
info ← 1
para i = 1 : n
    si A(i,i) > 0
        "paso usual de Cholesky"
    sino
        si A(i,i) < 0
            info ← −1
            parar
        sino
            "A(i,i) es nulo"
            info ← 0
            buscar q tal que A(q,q) = max{A(i,i), ..., A(n,n)}
            si A(q,q) = 0
                si A(i:n, i:n) ≠ 0
                    info ← −1
                fin-si
                parar
            sino
                "A(q,q) es positivo"
                intercSim(A, i, q)
                p(i) ⟷ p(q)
            fin-si
        fin-si
    fin-si
    A(i,i) ← sqrt(A(i,i))
    A(i, i+1:n) ← A(i, i+1:n)/A(i,i)
    para k = i+1, ..., n
        A(k, k:n) ← A(k, k:n) − A(i,k) A(i, k:n)
    fin-para
fin-para
```

Durante el algoritmo `info` puede tomar tres valores 1, 0 ó -1. Si `info` vale 1, entonces A puede ser definida positiva. Si `info` vale 0, entonces A no es definida positiva pero puede ser semidefinida positiva. Si `info` vale -1, entonces A no es semidefinida positiva.

Este algoritmo se puede modificar para hacer pivoteo siempre, es decir, aún si $a_{ii} > 0$. Si la matriz es definida positiva de todas maneras se pivotea y se obtiene (16).

Ejemplo 5.

```
A =
     0     2     0     0
     2     0     0     0
     0     0     4    -6
     0     0    -6    25

triu
     0     2     0     0
     0     0     0     0
     0     0     4    -6
     0     0     0    25

i = 1  ------------------------
A(i,i) =      0
q = 4
A(q,q) =     25
A(q,q) > 0.
despues de intercSim:
A =
    25     0    -6     0
     0     0     0     2
     0     0     4     0
     0     0     0     0

p :      4     2     3     1

despues de operaciones:
     5     0    -1.200000     0
     0     0     0            2
     0     0     2.560000     0
```

```
     0     0     0            0

i = 2  -----------------------
A(i,i) =      0
q = 3
A(q,q) =      2.560000
A(q,q) > 0.
despues de intercSim:
A =
     5    -1.200000     0     0
     0     2.560000     0     0
     0     0            0     2
     0     0            0     0

p :     4     3     2     1

despues de operaciones:
     5    -1.200000     0     0
     0     1.600000     0     0
     0     0            0     2
     0     0            0     0

i = 3  -----------------------
A(i,i) =      0
q = 3
A(q,q) =      0
A(i:n,i:n)  no nula.

La matriz  NO  es semidef. positiva.
```

Ejemplo 6.

```
A =
     0     0     0     0
     0     0     0     0
     0     0     4    -6
     0     0    -6    25
```

```
triu
     0     0     0     0
     0     0     0     0
     0     0     4    -6
     0     0     0    25

i = 1  -----------------------
A(i,i) =      0
q = 4
A(q,q) =     25
A(q,q) > 0.
despues de intercSim:
A =
    25     0    -6     0
     0     0     0     0
     0     0     4     0
     0     0     0     0

p :     4     2     3     1

despues de operaciones:
     5     0    -1.200000     0
     0     0     0            0
     0     0     2.560000     0
     0     0     0            0

i = 2  -----------------------
A(i,i) =      0
q = 3
A(q,q) =      2.560000
A(q,q) > 0.
despues de intercSim:
A =
     5    -1.200000     0     0
     0     2.560000     0     0
     0     0            0     0
     0     0            0     0
```

```
p :      4     3     2     1

despues de operaciones:
     5    -1.200000     0     0
     0     1.600000     0     0
     0     0            0     0
     0     0            0     0

i = 3  -----------------------
A(i,i) =      0
q = 3
A(q,q) =      0
A(i:n,i:n) = 0

info = 0
Matriz semidefinida positiva

U =
     5    -1.200000     0     0
     0     1.600000     0     0
     0     0            0     0
     0     0            0     0

UT U =
    25    -6     0     0
    -6     4     0     0
     0     0     0     0
     0     0     0     0

p :     4     3     2     1

P =
     0     0     0     1
     0     0     1     0
     0     1     0     0
     1     0     0     0
```

```
P A PT =
    25    -6     0     0
    -6     4     0     0
     0     0     0     0
     0     0     0     0
```

Referencias

[AlK02] Allaire G., Kaber S.M., *Algèbre linéaire numérique, Cours et exercises*, Ellipses, Paris, 2002.

[Dem97] Demmel J.W., *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[GoV96] Golub G.H., Van Loan C., *Matrix Computations*, 3rd ed., The Johns Hopkins Univ. Press, Baltimore, 1996.

[Hig90] Higham N.J., *Analysis of the Cholesky decomposition of a semidefinite matrix*, en Cox M.G. y Hammarling S.J. eds. *Reliable Numerical Computation*, p.161-185, Oxford University Press, 1990.

[Hig08] Higham N.J., *Cholesky Factorization*, University of Manchester, `http://www.man.ac.uc/~higham`, 2008.

[Mor01] Mora H., *Optiomización no lineal y dinámica* Facultad de Ciencias, Universidad Nacional, Bogotá, 2001.

[Ste98] Stewart G.W., *Matrix Algortihms, Volume I: Basic Decompositions*, SIAM, Philadelphia, 1998.

[TrB97] Trefethen L.N., Bau D., *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.