

Chapter 04.09

Cholesky and LDL^T Decomposition

Introduction

Solving large (and sparse) system of simultaneous linear equations (SLE) has been (and continues to be) a major challenging problem for many real-world engineering/science applications [1-2]. In matrix notation, the SLE can be represented as:

$$[A][x] = [b] \quad (1)$$

where $[A]$ = known coefficient matrix, with dimension $n \times n$

$[b]$ = known right-hand-side (RHS) $n \times 1$ vector

$[x]$ = unknown $n \times 1$ vector.

Symmetrical Positive Definite (SPD) SLE

For many practical SLE, the coefficient matrix $[A]$ (see Equation (1)) is Symmetric Positive Definite (SPD). In this case, the efficient 3-step Cholesky algorithms [1-2] can be used.

A matrix $[A]_{n \times n}$ can be considered as SPD if either of the following conditions is satisfied:

(a) If each and every determinant of sub-matrix $A_{ii} (i = 1, 2, \dots, n)$ is positive, or..

(b) If $y^T A y > 0$, for any given vector $[y]_{n \times 1} \neq \vec{0}$

As a quick example, let us make a test a test to see if the given matrix

$$[A] = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \text{ is SPD?}$$

Based on criteria (a):

The given 3×3 matrix $[A]$ is symmetrical, because $a_{ij} = a_{ji}$. Furthermore, one has

$$\det[A]_{1 \times 1} = |2| = 2 > 0$$

$$\begin{aligned} \det[A]_{2 \times 2} &= \begin{vmatrix} 2 & -1 \\ -1 & 2 \end{vmatrix} \\ &= 3 > 0 \end{aligned}$$

$$\begin{aligned} \det[A]_{3 \times 3} &= \begin{vmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{vmatrix} \\ &= 1 > 0 \end{aligned}$$

Hence $[A]$ is SPD.

Based on criteria (b): For any given vector

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \neq \vec{0}, \text{ one computes}$$

$$\text{scalar} = y^T A y$$

$$\begin{aligned} &= \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \\ &= (2y_1^2 - 2y_1y_2 + 2y_2^2) + \{y_3^2 - 2y_2y_3\} \\ &= (y_1 - y_2)^2 + y_1^2 + y_2^2 + \{y_3^2 - 2y_2y_3\} \end{aligned}$$

$$\text{scalar} = (y_1 - y_2)^2 + y_1^2 + (y_2 - y_3)^2 > 0$$

Since the above scalar is always positive, hence matrix $[A]$ is SPD.

Step 1: Matrix Factorization phase

In this step, the coefficient matrix $[A]$ that is SPD can be decomposed (or factorized) into

$$[A] = [U]^T [U] \quad (2)$$

where $[U]$ is a $n \times n$ upper triangular matrix.

The following simple 3×3 matrix example will illustrate how to find the matrix $[U]$.

Various terms of the factorized matrix $[U]$ can be computed/derived as follows (see Equation (2)):

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} u_{11} & 0 & 0 \\ u_{12} & u_{22} & 0 \\ u_{13} & u_{23} & u_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \quad (3)$$

Multiplying two matrices on the right-hand-side (RHS) of Equation (3), then equating each upper-triangular RHS terms to the corresponding ones on the upper-triangular left-hand-side (LHS), one gets the following 6 equations for the 6 unknowns in the factorized matrix $[U]$.

$$u_{11} = \sqrt{a_{11}}; u_{12} = \frac{a_{12}}{u_{11}}; u_{13} = \frac{a_{13}}{u_{11}} \quad (4)$$

$$u_{22} = \left(a_{22} - u_{12}^2\right)^{\frac{1}{2}}; u_{23} = \frac{a_{23} - u_{12}u_{13}}{u_{22}}; u_{33} = \left(a_{33} - u_{13}^2 - u_{23}^2\right)^{\frac{1}{2}} \quad (5)$$

In general, for a $n \times n$ matrix, the diagonal and off-diagonal terms of the factorized matrix $[U]$ can be computed from the following formulas:

$$u_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} (u_{ki})^2 \right)^{\frac{1}{2}} \quad (6)$$

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj}}{u_{ii}} \quad (7)$$

It is noted that if $i = j$, then the numerator of Equation (7) becomes identical to the terms under the square root in Equation (6). In other words, to factorize a general term u_{ij} , one simply needs to do the following steps:

Step 1.1: Compute the numerator of Equation (7), such as

$$Sum = a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj}$$

Step 1.2 If u_{ij} is an off-diagonal term (say, $i < j$) then $u_{ij} = \frac{Sum}{u_{ii}}$ (See Equation (7)). Else, if u_{ij} is a diagonal term (that is, $i = j$), then $u_{ii} = \sqrt{Sum}$ (See Equation (6))

As a quick example, one computes:

$$u_{57} = \frac{a_{57} - u_{15}u_{17} - u_{25}u_{27} - u_{35}u_{37} - u_{45}u_{47}}{u_{55}} \quad (8)$$

Thus, for computing $u(i=5, j=7)$, one only needs to use the (already factorized) data in columns $\#i(=5)$, and $\#j(=7)$ of $[U]$, respectively.

Remarks

- In general, to find the (off-diagonal) factorized term u_{ij} , one only needs to utilize the “already factorized” columns $\#i$, and $\#j$ information (see Figure 1). For example, if $i=5$, and $j=7$, then Figure 1 will lead to the same formula as shown earlier in Equation (7), or in Equation (8). Similarly, to find the (diagonal) factorized term u_{ii} , one simply needs to utilize columns $\#i$, and $\#i$ (again!) information (see Figure 1). In this case, Figure 1 will lead to the same formula as shown earlier in Equation (6).

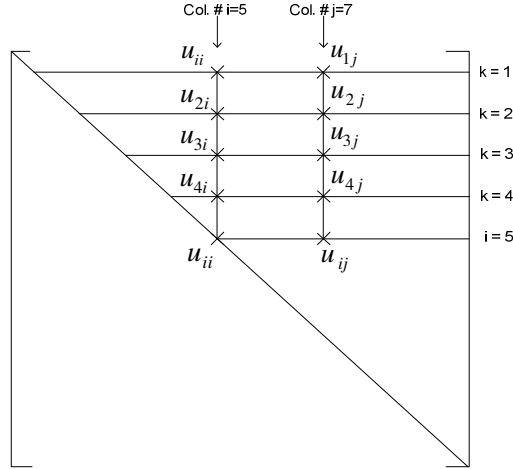


Figure 1 Cholesky Factorization for the term u_{ij}

- Since the square root operation involved during the Cholesky factorization phase (see Equation (6)), one must make sure the term under the square root is non-negative. This requirement is satisfied by $[A]$ being SPD.

Step 2: Forward Solution phase

Substituting Equation (2) into Equation (1), one gets:

$$[U]^T [U][x] = [b] \quad (9)$$

Let us define:

$$[U][x] \equiv [y] \quad (10)$$

Then, Equation (9) becomes:

$$[U]^T [y] = [b] \quad (11)$$

Since $[U]^T$ is a lower triangular matrix, Equation (11) can be efficiently solved for the

intermediate unknown vector $[y]$, according to the order $\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$, hence the name “forward

solution”.

As a quick example, one has (see Equation (11)):

$$\begin{bmatrix} u_{11} & 0 & 0 \\ u_{12} & u_{22} & 0 \\ u_{13} & u_{23} & u_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (12)$$

From the 1st row of Equation (12), one gets

$$\begin{aligned} u_{11}y_1 &= b_1 \\ y_1 &= \frac{b_1}{u_{11}} \end{aligned} \quad (13)$$

From the 2nd row of Equation (12), one gets

$$\begin{aligned} u_{12}y_1 + u_{22}y_2 &= b_2 \\ y_2 &= b_2 - \frac{u_{12}y_1}{u_{22}} \end{aligned} \quad (14)$$

Similarly

$$y_3 = \frac{b_3 - u_{13}y_1 - u_{23}y_2}{u_{33}} \quad (15)$$

In general, from the j^{th} row of Equation (12), one has

$$y_j = \frac{b_j - \sum_{i=1}^{j-1} u_{ij}y_i}{u_{jj}} \quad (16)$$

Step 3: Backward Solution phase

Since $[U]$ is an upper triangular matrix, Equation (10) can be efficiently solved for the original

unknown vector $[x]$, according to the order $\begin{bmatrix} x_n \\ x_{n-1} \\ x_{n-2} \\ \vdots \\ x_1 \end{bmatrix}$, hence the name “backward solution”.

As a quick example, one has (See Equation (10)):

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (17)$$

From the last (or $n^{th} = 3^{rd}$) row of Equation (17), one has

$$u_{33}x_3 = y_3.$$

hence

$$x_3 = \frac{y_3}{u_{33}} \quad (18)$$

Similarly:

$$x_2 = \frac{y_2 - u_{23}x_3}{u_{22}} \quad (19)$$

and

$$x_1 = \frac{y_1 - u_{12}x_2 - u_{13}x_3}{u_{11}} \quad (20)$$

In general, one has:

$$x_j = \frac{y_j - \sum_{i=j+1}^n u_{ji}x_i}{u_{jj}} \quad (21)$$

Remarks

(a) Amongst the above 3-step Cholesky algorithms, factorization phase in step 1 consumes about 95% of the total SLE solution time.

(b) If the coefficient matrix $[A]$ is symmetrical but not necessarily positive definite, then the above Cholesky algorithms will not be valid. In this case, the following LDL^T factorized algorithms can be employed:

$$[A] = [L][D][L]^T \quad (22)$$

For example,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \begin{bmatrix} 1 & l_{21} & l_{31} \\ 0 & 1 & l_{32} \\ 0 & 0 & 1 \end{bmatrix} \quad (23)$$

Multiplying the three matrices on the RHS of Equation (23), then equating the resulting upper-triangular RHS terms of Equation (23) to the corresponding ones on the LHS, one obtains the following formulas for the “diagonal” $[D]$, and “lower-triangular” $[L]$ matrices:

$$d_{jj} = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_{kk} \quad (24)$$

$$l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_{kk} l_{jk} \right) \times \left(\frac{1}{d_{jj}} \right) \quad (25)$$

Thus, the LDL^T algorithms can be summarized by the following step-by-step procedures.

Step1: Factorization phase

$$[A] = [L][D][L]^T \quad (22, \text{ repeated})$$

Step 2: Forward solution and diagonal scaling phase

Substituting Equation (22) into Equation (1), one gets:

$$[L][D][L]^T [x] = [b] \quad (26)$$

Let us define:

$$[L]^T [x] = [y]$$

$$\begin{bmatrix} 1 & l_{21} & l_{31} \\ 0 & 1 & l_{32} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (27)$$

$$x_i = y_i - \sum_{k=i+1}^n l_{ki} x_k; \text{ for } i = n, n-1, \dots, 1 \quad (28)$$

Also, define:

$$[D][y] = [z]$$

$$\begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad (29)$$

$$y_i = \frac{z_i}{d_{ii}}, \text{ for } i = 1, 2, 3, \dots, n \quad (30)$$

Then Equation (26) becomes:

$$[L][z] = [b]$$

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (31)$$

$$z_i = b_i - \sum_{k=1}^{i-1} L_{ik} z_k \text{ for } i = 1, 2, 3, \dots, n \quad (32)$$

Equation (31) can be efficiently solved for the vector $[z]$, then Equation (29) can be conveniently (and trivially) solved for the vector $[y]$.

Step 3: Backward solution phase

In this step, Equation (27) can be efficiently solved for the original unknown vector $[x]$.

Numerical Example 1 (Cholesky algorithms)

Solve the following SLE system for the unknown vector $[x]$?

$$[A][x] = [b]$$

where

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

$$[b] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Solution:

The factorized, upper triangular matrix $[U]$ can be computed by either referring to Equations (6-7), or looking at Figure 1, as following:

$$\left. \begin{aligned} u_{11} &= \sqrt{a_{11}} \\ &= \sqrt{2} \\ &= 1.414 \\ u_{12} &= \frac{a_{12}}{u_{11}} \\ &= \frac{-1}{1.414} \\ &= -0.7071 \\ u_{13} &= \frac{a_{13}}{u_{11}} \\ &= \frac{0}{1.414} \\ &= 0 \end{aligned} \right\} \text{ row 1 of } [U]$$

$$\left. \begin{aligned}
u_{22} &= \left\{ a_{22} - \sum_{k=1}^{i-1=1} (u_{ki})^2 \right\}^{\frac{1}{2}} \\
&= \left\{ 2 - (u_{12})^2 \right\}^{\frac{1}{2}} \\
&= \sqrt{2 - (-0.7071)^2} \\
&= 1.225 \\
u_{23} &= \frac{a_{23} - \sum_{k=1}^{i-1=1} u_{ki} u_{kj}}{U_{22}} \\
&= \frac{-1 - u_{12} \times u_{13}}{1.225} \\
&= \frac{-1 - (-0.7071)(0)}{1.225} \\
&= -0.8165
\end{aligned} \right\} \text{row 2 of } [U]$$

$$\left. \begin{aligned}
u_{33} &= \left\{ a_{33} - \sum_{k=1}^{i-1=2} (u_{ki})^2 \right\}^{\frac{1}{2}} \\
&= \left\{ a_{33} - u_{13}^2 - u_{23}^2 \right\}^{\frac{1}{2}} \\
&= \sqrt{1 - (0)^2 - (-0.8165)^2} \\
&= 0.5774
\end{aligned} \right\} \text{row 3 of } [U]$$

Thus, the factorized matrix

$$[U] = \begin{bmatrix} 1.414 & -0.7071 & 0 \\ 0 & 1.225 & -0.8165 \\ 0 & 0 & 0.5774 \end{bmatrix}$$

The forward solution phase, shown in Equation (11), becomes:

$$[U]^T [y] = [b]$$

$$\begin{bmatrix} 1.414 & 0 & 0 \\ -0.7071 & 1.225 & 0 \\ 0 & -0.8165 & 0.5774 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Thus, Equation (16) can be used to solve:

$$y_1 = \frac{b_1}{u_{11}}$$

$$= \frac{1}{1.414}$$

$$= 0.7071$$

$$y_2 = \frac{b_2 - \sum_{i=1}^{j-1=1} u_{ij} y_i}{u_{jj}}$$

$$= \frac{0 - (u_{12} = -0.7071)(y_1 = 0.7071)}{(u_{22} = 1.225)}$$

$$= 0.4082$$

$$y_3 = \frac{b_3 - \sum_{i=1}^{j-1=2} u_{ij} y_i}{u_{jj}}$$

$$= \frac{0 - (u_{13} = 0)(y_1 = 0.7071) - (u_{23} = -0.8165)(y_2 = 0.4082)}{(u_{33} = 0.5774)}$$

$$= 0.5774$$

The backward solution phase, shown in Equation (10), becomes:

$$[U][x] = [y]$$

$$\begin{bmatrix} 1.414 & -0.7071 & 0 \\ 0 & 1.225 & -0.8165 \\ 0 & 0 & 0.5774 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.7071 \\ 0.4082 \\ 0.5774 \end{bmatrix}$$

Thus, Equation (21) can be used to solve:

$$\begin{aligned}
x_3 &= \frac{y_j}{u_{jj}} \\
&= \frac{y_3}{u_{33}} \\
&= \frac{0.5774}{0.5774} \\
&= 1 \\
x_2 &= \frac{y_j - \sum_{i=j+1}^{N=3} u_{ji}x_i}{u_{jj}} \\
&= \frac{y_2 - u_{23}x_3}{u_{22}} \\
&= \frac{0.4082 - (-0.8165)(1)}{1.225} \\
&= 1 \\
x_1 &= \frac{y_j - \sum_{i=j+1}^{N=3} u_{ji}x_i}{u_{jj}} \\
&= \frac{y_1 - u_{12}x_2 - u_{13}x_3}{u_{11}} \\
&= \frac{0.7071 - (-0.7071)(1) - (0)(1)}{1.414} \\
&= 1
\end{aligned}$$

Hence

$$[x] = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Numerical Example 2 (LDL^T Algorithms)

Using the same data given in Numerical Example 1, find the unknown vector $[x]$ by LDL^T algorithms?

Solution:

The factorized matrices $[D]$ and $[L]$ can be computed from Equation (24), and Equation (25), respectively.

$$\left. \begin{aligned}
d_{11} &= a_{11} - \sum_{k=1}^{j-1=0} l_{jk}^2 d_{kk} \\
&= a_{11} \\
&= 2 \\
l_{11} &= 1 \text{ (always !)} \\
l_{21} &= \frac{a_{21} - \sum_{k=1}^{j-1=0} l_{ik} d_{kk} l_{jk}}{d_{jj}} \\
&= \frac{a_{21}}{d_{11}} \\
&= \frac{-1}{2} \\
&= -0.5 \\
l_{31} &= \frac{a_{31}}{d_{11}} \\
&= \frac{0}{2} \\
&= 0
\end{aligned} \right\} \text{Column 1 of matrices } [D] \text{ and } [L]$$

$$\left. \begin{aligned}
d_{22} &= a_{22} - \sum_{k=1}^{j-1=1} l_{jk}^2 d_{kk} \\
&= 2 - l_{21}^2 d_{11} \\
&= 2 - (-0.5)^2 (2) \\
&= 1.5 \\
l_{22} &= 1 \text{ (always !)} \\
l_{32} &= \frac{a_{32} - \sum_{k=1}^{j-1=1} l_{31} d_{11} l_{21}}{d_{22}} \\
&= \frac{-1 - (0)(2)(-0.5)}{1.5} \\
&= -0.6667
\end{aligned} \right\} \text{Column 2 of matrices } [D] \text{ and } [L]$$

$$\left. \begin{aligned}
d_{33} &= a_{33} - \sum_{k=1}^{j-1=2} l_{jk}^2 d_{kk} \\
&= 1 - l_{31}^2 d_{11} - l_{32}^2 d_{22} \\
&= 1 - (0)^2 (2) - (-0.6667)^2 (1.5) \\
&= 0.3333
\end{aligned} \right\} \text{Column 3 of matrices } [D] \text{ and } [L]$$

Hence

$$[D] = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0.3333 \end{bmatrix}$$

and

$$[L] = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0 & -0.6667 & 1 \end{bmatrix}$$

The forward solution shown in Equation (31), becomes:

$$[L][z] = [b]$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0 & -0.6667 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

or,

$$z_i = b_i - \sum_{k=1}^{i-1} l_{ik} z_k \quad (32, \text{ repeated})$$

Hence

$$z_1 = b_1 = 1$$

$$\begin{aligned} z_2 &= b_2 - L_{21}z_1 \\ &= 0 - (-0.5)(1) \\ &= 0.5 \end{aligned}$$

$$\begin{aligned} z_3 &= b_3 - L_{31}z_1 - L_{32}z_2 \\ &= 0 - (0)(1) - (-0.6667)(0.5) \\ &= 0.3333 \end{aligned}$$

The diagonal scaling phase, shown in Equation (29), becomes

$$[D][y] = [z]$$

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0.3333 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 0.3333 \end{bmatrix}$$

or

$$y_i = \frac{z_i}{d_{ii}}$$

Hence

$$\begin{aligned} y_1 &= \frac{z_1}{d_{11}} \\ &= \frac{1}{2} \\ &= 0.5 \end{aligned}$$

$$\begin{aligned} y_2 &= \frac{z_2}{d_{22}} \\ &= \frac{0.5}{1.5} \\ &= 0.3333 \end{aligned}$$

$$\begin{aligned} y_3 &= \frac{z_3}{d_{33}} \\ &= \frac{0.3333}{0.3333} \\ &= 1 \end{aligned}$$

The backward solution phase can be found by referring to Equation (27)

$$[L]^T [x] = [y]$$

$$\begin{bmatrix} 1 & -0.5 & 0 \\ 0 & 1 & -0.667 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.333 \\ 1 \end{bmatrix}$$

$$x_i = y_i - \sum_{k=i+1}^N l_{ki} x_k \quad (28, \text{ repeated})$$

Hence

$$\begin{aligned}
 x_3 &= y_3 \\
 &= 1 \\
 x_2 &= y_2 - l_{32}x_3 \\
 &= 0.3333 - (-0.6667) \times 1 \\
 x_2 &= 1 \\
 x_1 &= y_1 - l_{21}x_2 - l_{31}x_3 \\
 x_1 &= 0.5 - (-0.5)(1) - (0)(1) \\
 &= 1
 \end{aligned}$$

Hence

$$[x] = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Remarks

Through this numerical example, one clearly sees that the “square root operations” have NOT been involved during the entire LDL^T algorithms. Thus, the coefficient matrix $[A]$, shown in Equation (1), is NOT required to be SPD.

Re-ordering Algorithms For Minimizing Fill-in Terms [1,2].

During the factorization phase (of Cholesky, or LDL^T algorithms), many “zero” terms in the original/given matrix $[A]$ will become “non-zero” terms in the factored matrix $[U]$. These new non-zero terms are often called as “fill-in” terms (indicated by the symbol F). It is, therefore, highly desirable to minimize these fill-in terms, so that both computational time/effort and computer memory requirements can be substantially reduced. For example, the following matrix $[A]$ and vector $[b]$ are given:

$$[A] = \begin{bmatrix} 112 & 7 & 0 & 0 & 0 & 2 \\ 7 & 110 & 5 & 4 & 3 & 0 \\ 0 & 5 & 88 & 0 & 0 & 1 \\ 0 & 4 & 0 & 66 & 0 & 0 \\ 0 & 3 & 0 & 0 & 44 & 0 \\ 2 & 0 & 1 & 0 & 0 & 11 \end{bmatrix} \quad (33)$$

$$[b] = \begin{bmatrix} 121 \\ 129 \\ 94 \\ 70 \\ 47 \\ 14 \end{bmatrix} \quad (34)$$

The Cholesky factorization matrix $[U]$, based on the original matrix $[A]$ (see Equation 33) and Equations (6-7), or Figure 1, can be symbolically computed as:

$$[U] = \begin{bmatrix} \times & \times & 0 & 0 & 0 & \times \\ 0 & \times & \times & \times & \times & F \\ 0 & 0 & \times & F & F & \times \\ 0 & 0 & 0 & \times & F & F \\ 0 & 0 & 0 & 0 & \times & F \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix} \quad (35)$$

In Equation (35), the symbols \times , and F represents the “non-zero” and “fill-in” terms, respectively.

In practical applications, however, it is always a necessary step to rearrange the original matrix $[A]$ through re-ordering algorithms (or subroutines) [Refs 1-2] and produce the following integer mapping array

$$\text{IPERM}(\text{new equation \#}) = \{\text{old equation \#}\} \quad (36)$$

such as, for this particular example:

$$\text{IPERM} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix} \quad (37)$$

Using the above results (see Equation 37), one will be able to construct the following re-arranged matrices:

$$[A^*] = \begin{bmatrix} 11 & 0 & 0 & 1 & 0 & 2 \\ 0 & 44 & 0 & 0 & 3 & 0 \\ 0 & 0 & 66 & 0 & 4 & 0 \\ 1 & 0 & 0 & 88 & 5 & 0 \\ 0 & 3 & 4 & 5 & 110 & 7 \\ 2 & 0 & 0 & 0 & 7 & 112 \end{bmatrix} \quad (38)$$

and

$$[b^*] = \begin{bmatrix} 14 \\ 47 \\ 70 \\ 94 \\ 129 \\ 121 \end{bmatrix} \quad (39)$$

Remarks:

- In the original matrix A (shown in Equation 33), the nonzero term A (old row 1, old column 2) = 7 will move to new location of the new matrix A^* (new row 6, new column 5) = 7, etc.
- The non zero term A (old row 3, old column 3) = 88 will move to A^* (new row 4, new column 4) = 88, etc.
- The value of b (old row 4) = 70 will be moved to (or located at) b^* (new row 3) = 70, etc.

Now, one would like to solve the following modified system of linear equations (SLE) for $[x^*]$,

$$[A^*][x^*] = [b^*] \quad (40)$$

rather than to solve the original SLE (see Equation 1). The original unknown vector $\{x\}$ can be easily recovered from $[x^*]$ and $[IPERM]$, shown in Equation (37).

The factorized matrix $[U^*]$ can be “symbolically” computed from $[A^*]$ as (by referring to either Figure 1, or Equations 6-7):

$$[U^*] = \begin{bmatrix} \times & 0 & 0 & \times & 0 & \times \\ 0 & \times & 0 & 0 & \times & 0 \\ 0 & 0 & \times & 0 & \times & 0 \\ 0 & 0 & 0 & \times & \times & F \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix} \quad (41)$$

You can clearly see the big benefits of solving the SLE shown in Equation (40), instead of solving the original Equation (1), since the factorized matrix $[U^*]$ has only 1 fill-in term (see the symbol “ F ” in Equation 41), as compared to six fill-in-terms occurred in the factorized matrix $[U]$ (shown in Equation 35)!

On-Line Chess-Like Game For Reordering/Factorized Phase [4].

Based on the discussions presented in the previous section 2 (about factorization phase), and section 3 (about reordering phase), one can easily see the similar operations between the symbolic, numerical factorization and reordering (to minimize the number of fill-in terms) phases of sparse SLE.

In practical computer implementation for the solution of SLE, the reordering phase is usually conducted first (to produce the mapping between “old \leftrightarrow new” equation numbers, as indicated in the integer array IPERM(-), see Equations 36-37).

Then, the sparse “symbolic” factorization phase is followed by using either Cholesky Equations 6-7, or the LDL^T Equations 24-25 (without requiring the actual/numerical values to be computed). The reason is because during the “symbolic factorization” phase, one only wishes to find the number (and the location) of non-zero “fill-in terms”. This “symbolic” factorization process is necessary for allocating the “computer memory” requirement for the “numerical factorization” phase which will actually compute the exact numerical values of $[U^*]$, based on the same Cholesky Equations (6-7) (or the LDL^T Equations (24-25)).

In this work, a chess-like game (shown in Figure 2, Ref. [4]) has been designed with the following objectives:

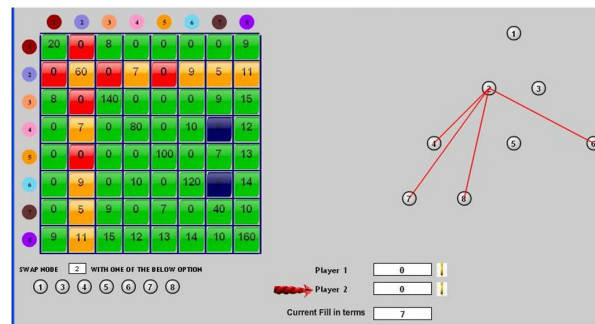


Figure 2 A Chess-Like Game For Learning to Solve SLE.

(A) Teaching undergraduate/HS students the process how to use the reordering output

IPERM(-), see Equations (36-37) for converting the original/given matrix $[A]$, see Equation (33), into the new/modified matrix $[A^*]$, see Equation (38). This step is reflected in Figure 2, when the “Game Player” decides to swap node (or equation) “ i ” (say $i = 2$) with another node (or equation) “ j ”, and click the “CONFIRM” icon!

Since node “ $i = 2$ ” is currently connected to nodes $j = 4, 6, 7, 8$; hence swapping node $i = 2$ with the above nodes “ j ” will “NOT” change the number/pattern of “Fill-in” terms. However, if node $i = 2$ is swapped with node $j = 1, \text{ or } 3, \text{ or } 5$, then the fill-in terms pattern may change (for better or worse)!

(B) Helping undergraduate/HS students to understand the “symbolic” factorization” phase, by symbolically utilizing the Cholesky factorized Equations (6-7). This step is illustrated in Figure 2, for which the “game player” will see (and also hear the computer animated sound, and human

voice), the non-zero terms (including fill-in terms) of the original matrix $[A]$ to move to the new locations in the new/modified matrix $[A^*]$.

(C) Helping undergraduate/HS students to understand the “numerical factorization” phase, by numerically utilizing the same Cholesky factorized Equations (6-7).

(D) Teaching undergraduate engineering/science students and even high-school (HS) students to “understand existing reordering concepts”, or even to “discover new reordering algorithms”

Further Explanation On The Developed Game

1. In the above Chess-Like Game, which is available on-line [4], powerful features of FLASH computer environments [3], such as animated sound, human voice, motions, graphical colors etc... have all been incorporated and programmed into the developed game-software for more appealing to game players/learners.

2. In the developed “Chess-Like Game”, fictitious monetary (or any kind of ‘scoring system’) is rewarded (and broadcasted by computer animated human voice) to game players, based on how he/she swaps the node (or equation) numbers, and consequently based on how many fill-in “ F ” terms occurred. In general, less fill-in terms introduced will result in more rewards!

3. Based on the original/given matrix $[A]$, and existing re-ordering algorithms (such as the Reverse Cuthill-Mckee, or RCM algorithms [1-2]) the number of fill-in (“ F ”) terms can be computed (using RCM algorithms). This internally generated information will be used to judge how good the players/learners are, and/or broadcast “congratulations message” to a particular player who discovers new “chess-like move” (or, swapping node) strategies which are even better than RCM algorithms!

4. Initially, the player(s) will select the matrix size (8×8 , or larger is recommended), and the percentage (50%, or larger is suggested) of zero-terms (or sparsity of the matrix). Then, “START Game” icon will be clicked by the player.

5. The player will then CLICK one of the selected node “ i ” (or equation) numbers appearing on the computer screen. The player will see those nodes “ j ” which are connected to node “ i ” (based on the given/generated matrix $[A]$). The player then has to decide to swap node “ i ” with one of the possible node “ j ”. After confirming the player’s decision, the outcomes/results will be announced by the computer animated human voice, and the monetary-award will (or will NOT) be given to the players/learners, accordingly. In this software, a maximum of \$1,000,000 can be earned by the player, and the “exact dollar amount” will be INVERSELY proportional to the number of fill-in terms occurred (as a consequence of the player’s decision on how to swap node “ i ” with another node “ j ”).

6. The next player will continue to play, with his/her move (meaning to swap the i^{th} node with the j^{th} node) based on the current best non-zero terms pattern of the matrix.

Multiple Choice Tests

Cholesky Algorithms

For a given SLE:

$$[A][x] = [b]$$

where

$$[A] = \begin{bmatrix} 2 & -1 & 0 & 0.5 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0.5 & 0 & -1 & 1 \end{bmatrix}$$

and

$$[b] = \begin{bmatrix} -0.5 \\ 5 \\ -5 \\ 1.5 \end{bmatrix}$$

1. The Cholesky factorized matrix $[U]$ can be computed as

$$(a) \begin{bmatrix} 1.414 & 0.7071 & 0 & 0.3536 \\ 0 & 1.225 & -0.8165 & 0.2041 \\ 0 & 0 & 1.155 & -0.7217 \\ 0 & 0 & 0 & 0.5590 \end{bmatrix}$$

$$(b) \begin{bmatrix} 1.414 & -0.7071 & 0 & 0.3536 \\ 0 & 1.225 & -0.8165 & 0.2041 \\ 0 & 0 & 1.155 & -0.7217 \\ 0 & 0 & 0 & 0.5590 \end{bmatrix}$$

$$(c) \begin{bmatrix} 1.414 & 0.7071 & 0 & -0.3536 \\ 0 & 1.225 & -0.8165 & 0.2041 \\ 0 & 0 & 1.155 & -0.7217 \\ 0 & 0 & 0 & 0.5590 \end{bmatrix}$$

$$(d) \begin{bmatrix} 1.414 & 0.7071 & 0 & 0.3536 \\ 0 & 1.225 & -0.8165 & 0.2041 \\ 0 & 0 & -1.155 & -0.7217 \\ 0 & 0 & 0 & 0.5590 \end{bmatrix}$$

2. The forward solution vector $[y]$ can be computed as

- (a) $\vec{y}^T = \{0.5363, 38.784, -15.877, 0.5590\}$
- (b) $\vec{y}^T = \{0.5363, -15.877, 38.784, 0.5590\}$
- (c) $\vec{y}^T = \{-3.536, -1.5877, 3.878, 0.5590\}$
- (d) $\vec{y}^T = \{-0.3536, 3.8784, -1.5877, -0.5590\}$

3. The backward solution vector $[x]$ can be computed as

- (a) $\vec{x}^T = \{1, 2, -2, -1\}$
- (b) $\vec{x}^T = \{1, 2, 2, -1\}$
- (c) $\vec{x}^T = \{-1, 2, -2, 1\}$
- (d) $\vec{x}^T = \{1, 2, 2, 1\}$

4. The determinant of $[A]_{4 \times 4}$ can be computed as:

- (a) -5
- (b) 5
- (c) -50
- (d) 1.25

5. Based on the given matrix $[A]$, and assuming the reordering algorithm will produce the

following mapping $IPERM(\text{new equation \#}) = \{\text{old equation \#}\}$, such as $IPERM \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix}$. The

non zero off-diagonal term $A(\text{old row 4, old column 1}) = 0.5$ will move to the following new location of the new matrix $[A^*]$

- (a) $A^*(\text{new row 3, new column 1})$
- (b) $A^*(\text{new row 1, new column 3})$
- (c) $A^*(\text{new row 3, new column 2})$

(d) A^* (new row 2, new column 2)

6. Based on the given matrix $[A]$, and the given reordering mapping $IPERM \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix}$, the non-

zero diagonal term $A(4,4) = 1$ will move to the following new location of the new matrix $[A^*]$

(a) $A^*(1,1) = 1$

(b) $A^*(2,2) = 1$

(c) $A^*(3,3) = 1$

(d) $A^*(4,4) = 1$

LDL^T Algorithms

For the given SLE:

$$[A][x] = [b]$$

where

$$[A] = \begin{bmatrix} 2 & -1 & 0 & 0.5 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0.5 & 0 & -1 & 1 \end{bmatrix}$$

and

$$[b] = \begin{bmatrix} 2 \\ -1 \\ 1 \\ 0.5 \end{bmatrix}$$

7. The lower triangular matrix $[L]$ can be computed as

$$(a) [L] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.5 & 1 & 0 & 0 \\ 0 & -0.6667 & 1 & 0 \\ 0.25 & 0.1667 & -0.625 & 1 \end{bmatrix}$$

$$(b) [L] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.5 & 1 & 0 & 0 \\ 0 & 0.6667 & 1 & 0 \\ 0.25 & 0.1667 & -0.625 & 1 \end{bmatrix}$$

$$(c) [L] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.5 & 1 & 0 & 0 \\ 0 & -0.6667 & 1 & 0 \\ 0.25 & 0.1667 & 0.625 & 1 \end{bmatrix}$$

$$(d) [L] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0 & -0.6667 & 1 & 0 \\ 0.25 & 0.1667 & -0.625 & 1 \end{bmatrix}$$

8. The diagonal matrix $[D]$ can be computed as:

$$(a) [D] = \begin{bmatrix} -2 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 \\ 0 & 0 & 1.3333 & 0 \\ 0 & 0 & 0 & 0.3125 \end{bmatrix}$$

$$(b) [D] = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & -1.5 & 0 & 0 \\ 0 & 0 & 1.3333 & 0 \\ 0 & 0 & 0 & 0.3125 \end{bmatrix}$$

$$(c) [D] = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 \\ 0 & 0 & 1.3333 & 0 \\ 0 & 0 & 0 & -0.3125 \end{bmatrix}$$

$$(d) [D] = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 \\ 0 & 0 & 1.3333 & 0 \\ 0 & 0 & 0 & 0.3125 \end{bmatrix}$$

9. The forward solution for the unknown vector $[z]$, in $[L][z] = [b]$ can be computed as:

(a) $\{z\}^T = \{-2, 0, 1, 0.625\}$

(b) $\{z\}^T = \{2, 0, 1, 0.625\}$

(c) $\{z\}^T = \{2, 0, -1, 0.625\}$

(d) $\{z\}^T = \{2, 0, 1, -0.625\}$

10. The diagonal scaling solution for the unknown vector $[y]$ in $[D][y] = [z]$ can be computed as:

(a) $\{y\}^T = \{-1, 0, 0.75, 2\}$

(b) $\{y\}^T = \{1, 0, -0.75, 2\}$

(c) $\{y\}^T = \{1, 0, 0.75, -2\}$

(d) $\{y\}^T = \{1, 0, 0.75, 2\}$

11. The backward solution for the original unknown vector $[x]$, in $[L]^T[x] = [y]$, can be computed as:

(a) $\{x\}^T = \{1, 1, 2, 2\}$

(b) $\{x\}^T = \{2, 1, 2, 1\}$

(c) $\{x\}^T = \{1, 1, 2, 1\}$

(d) $\{x\}^T = \{2, 2, 2, 1\}$

12. Given the following 6x6 matrix $[A]$, which is assumed to be SPD:

$$[A] = \begin{bmatrix} \times & 0 & \times & 0 & \times & 0 \\ & \times & 0 & \times & 0 & 0 \\ & & \times & 0 & \times & \times \\ \text{Sym} & & & \times & 0 & 0 \\ & & & & \times & 0 \\ & & & & & \times \end{bmatrix}$$

where \times = a nonzero value (given)

0 = a zero value (given)

Based on the numerically factorized formulas (shown in Equations 6-7), or even more helpful information as indicated in Figure 1, the symbolically factorized (upper-triangular) matrix can be obtained as:

$$(a) [U] = \begin{bmatrix} * & 0 & * & 0 & * & 0 \\ & * & 0 & * & 0 & 0 \\ & & * & 0 & * & * \\ & & & * & 0 & 0 \\ & & & & * & F \\ & & & & & * \end{bmatrix}$$

where

$*$ = a nonzero value (computed, at the same location as the original nonzero value of $[A]$)

0 = a zero value

F = a nonzero Fill-in-Term (computed)

Note

$$U(5,6) = F$$

$$A(5,6) = 0$$

$$(b) [U] = \begin{bmatrix} * & 0 & * & 0 & * & 0 \\ & * & 0 & F & 0 & 0 \\ & & * & 0 & * & * \\ & & & * & 0 & 0 \\ & & & & * & F \\ & & & & & * \end{bmatrix}$$

$$(c) [U] = \begin{bmatrix} * & 0 & * & 0 & * & 0 \\ & * & 0 & F & 0 & 0 \\ & & * & 0 & F & * \\ & & & * & 0 & 0 \\ & & & & * & F \\ & & & & & * \end{bmatrix}$$

$$(d) [U] = \begin{bmatrix} * & 0 & * & 0 & * & 0 \\ & * & 0 & F & 0 & 0 \\ & & * & 0 & F & F \\ & & & * & 0 & 0 \\ & & & & * & F \\ & & & & & * \end{bmatrix}$$

References

- [1] Duc T. Nguyen, "Finite Element Methods: Parallel-Sparse Statics and Eigen-Solutions", Springer Publisher (2006).
- [2] Duc T. Nguyen, "Parallel-vector Equation Solvers for Finite Element Engineering Applications", Kluwer Academic/Plenum Publishers (2002).
- [3] www.brothersoft.com/downloads/flash-animation-software.html.
- [4] <http://www.lions.odu.edu/~amoha006/Fillinterms/FILLINTERMS.html>

CHOLESKY AND LDL^T DECOMPOSITION

Topic	Cholesky and LDL^T Decomposition
Summary	Textbook notes on Cholesky and LDL^T Decomposition
Major	General Engineering
Authors	Duc Nguyen
Date	September 16, 2010
Web Site	http://numericalmethods.eng.usf.edu
