LU Decomposition *

Nobuyuki Umetani

March 23, 2021

Contents

1 Introduction **2**

2 Block LU Decomposition **2**
2.1 Block LU decomposition . . . 2
2.1.1 Inverse matrix of block lower triangular matrix . . . 3
2.1.2 Solving a linear system using the block LU decomposition . . . 3
2.1.3 Eigenvalues of block triangular matrix . . . 4
2.2 Block LDU decomposition . . . 4
2.3 Symmetric LU decomposition for a positive definite matrix . . . 4

3 LU decomposition **5**
3.1 LDU decomposition . . . 5
3.1.1 Algorithm . . . 7
3.1.2 Algorithm in detail . . . 7
3.2 LU decomposition . . . 8
3.2.1 LU decomposition part 1 (decomposition with 1 diagonal of L) . 8
3.2.2 LU decomposition part 2 (decomposition with one diagonal of U) 9
3.3 Solving a linear system . . . 10
3.3.1 How to find z by solving Lz = y . . . 10
3.3.2 How to obtain x by solving Ux = z . . . 10
3.4 Speed up compression display and calculation . . . 11
3.4.1 LU factorization (LU decomposition with U diagonal 1, compressed display with diagonal reciprocal) . . . 11

*This is a memorandum to write down what the forgetful author studied a long time ago. Surely it contains many mistakes. Excuse me. It is appreciated if you let me know if you have any comments or suggestions.

3.4.2 Forward substitution (LU decomposition with U diagonal 1, compressed display with diagonal reciprocal) 12
3.4.3 Back substitution (LU decomposition with U diagonal 1, compressed display with diagonal reciprocal) 12
3.4.4 LU decomposition algorithm diagram (compressed display) . . 12

1 Introduction

The LU decomposition is often used to solve a linear system. In this document, we take a look at how the LU decomposition is computed. LU factorization is a method of expressing a coefficient matrix by the product of a lower triangular matrix and an upper triangular matrix. By using the fact that there are only two triangular matrix after the LU decomposition, linear system can be solved very efficiently using forward / backward substitution after the decomposition. Here, we briefly describe how the LU decomposition works and how it is implemented. We first explain the block LU decomposition. The LU decomposition is obtained by repeating the special case of block LU decomposition many times.

2 Block LU Decomposition

2.1 Block LU decomposition

Consider the following square matrix

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{E} \end{bmatrix}, \tag{1}$$

where **A**,**B**,**C** and **E** are also matrices (we use **E** instead of **D** here to avoid confusion with a diagonal matrix). It is easy to check that this matrix can be written as the multiplication of two matrices

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{E} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{CA}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{E} - \mathbf{CA}^{-1}\mathbf{B} \end{bmatrix}. \tag{2}$$

The decomposition results in a block lower triangle matrix and a block upper triangle matrix. Thus, this decomposition is called block LU decomposition. To be more precise, there are several types of block LU decomposition and this decomposition is a special case where a diagonal block of the lower block triangle matrix becomes a unit matrix. The lower right entry of the right matrix is called *Shur complement* and sometimes write **S**.

$$\mathbf{E} - \mathbf{CA}^{-1}\mathbf{B} = \mathbf{S} \tag{3}$$

2.1.1 Inverse matrix of block lower triangular matrix

You can easily check the following equation by hand:

$$\begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{CA}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathbf{I} & 0 \\ -\mathbf{CA}^{-1} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \tag{4}$$

This mean that the inverse of the block lower triangle matrix is:

$$\begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{CA}^{-1} & I \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & 0 \\ -\mathbf{CA}^{-1} & \mathbf{I} \end{bmatrix} \tag{5}$$

2.1.2 Solving a linear system using the block LU decomposition

Consider the following linear system.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{E} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{Bmatrix} = \begin{Bmatrix} \mathbf{y}_A \\ \mathbf{y}_B \end{Bmatrix} \tag{6}$$

Let use have the block LU decomposition on the coefficient matrix.

$$\begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{CA}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{S} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{Bmatrix} = \begin{Bmatrix} \mathbf{y}_A \\ \mathbf{y}_B \end{Bmatrix} \tag{7}$$

where I wrote Shur Complement as **S**.

Multiplying from the left to the inverse of the lower triangular matrix,

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{S} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{Bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ -\mathbf{CA}^{-1} & I \end{bmatrix} \begin{Bmatrix} \mathbf{y}_A \\ \mathbf{y}_B \end{Bmatrix} \tag{8}$$

Therefore, the solution is obtained as follows

$$\mathbf{x}_B = \mathbf{S}^{-1}\{-\mathbf{CA}\mathbf{y}_A + \mathbf{y}_B\} \tag{9}$$

$$\mathbf{x}_A = \mathbf{A}^{-1}\{\mathbf{y}_A - \mathbf{B}\mathbf{x}_B\} \tag{10}$$

The opposite is **A** and **S**. For situations in which the inverse of A can be easily obtained (for example, A is a diagonal matrix), the solution in such block decomposition can reduce the order of the solved matrix and is efficient.

2.1.3 Eigenvalues of block triangular matrix

Let's find the eigenvalues of the next block upper triangular matrix which is the result of the above block LU decomposition.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{S} \end{bmatrix} \phi = \lambda \phi \tag{11}$$

The characteristic equation is as follows

$$\det \begin{bmatrix} \mathbf{A} - \lambda \mathbf{I} & \mathbf{B} \\ 0 & \mathbf{S} - \lambda \mathbf{I} \end{bmatrix} = 0 \tag{12}$$

$$\Leftrightarrow \quad \det(\mathbf{A} - \lambda \mathbf{I}) \det(\mathbf{S} - \lambda \mathbf{I}) = 0 \tag{13}$$

The eigenvalues of the block triangular matrix are equal to the eigenvalues of the diagonal block matrices. Therefore, the eigenvalue λ is either one of the eigenvalues of **A** or one of eigenvalues of **S**. Also, it can be seen that the eigenvalue of the block triangular matrix whose diagonal matrix is an identity matrix is 1. Shur Complement **S** is very important as it retains the properties of the original matrix.

2.2 Block LDU decomposition

Let us disassemble the block upper triangular matrix of LU decomposition as follows.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{E} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{E} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix} \tag{14}$$

Using this, it can be written symmetrically as follows.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{E} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{C}\mathbf{A}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix} \tag{15}$$

2.3 Symmetric LU decomposition for a positive definite matrix

Consider the case where the original matrix is a positive definite symmetric matrix. In other words,

$$\mathbf{A}^T = \mathbf{A} \tag{16}$$

$$\mathbf{C} = \mathbf{B}^T \tag{17}$$

$$\mathbf{E}^T = \mathbf{E} \tag{18}$$

Since the eigenvalues of the original matrix are equal to eigenvalues of **A** and **S**, the fact that the original matrix is positive definite means that both **A** and **S** are positive definite. At this time, it is possible to decompose by the real matrix L_A, L_S as follows. $\mathbf{A} = \mathbf{L}_A\mathbf{L}_A^T$, $\mathbf{S} = \mathbf{L}_S\mathbf{L}_S^T$ Using this, the diagonal block matrix of LDU decomposition can be decomposed as follows.

$$\begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{S} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_A & 0 \\ 0 & \mathbf{L}_S \end{bmatrix} \begin{bmatrix} \mathbf{L}_A^T & 0 \\ 0 & \mathbf{L}_S^T \end{bmatrix} \tag{19}$$

By assigning this to above,

$$\begin{aligned}
\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{D} \end{bmatrix} &= \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{B}^T\mathbf{A}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathbf{L}_A & 0 \\ 0 & \mathbf{L}_S \end{bmatrix} \begin{bmatrix} \mathbf{L}_A^T & 0 \\ 0 & \mathbf{L}_S^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix} &(20)\\
&= \begin{bmatrix} \mathbf{L}_A & 0 \\ \mathbf{B}^T\mathbf{L}_A^{-T} & \mathbf{L}_S \end{bmatrix} \begin{bmatrix} \mathbf{L}_A^T & \mathbf{L}_A^{-T}\mathbf{B} \\ 0 & \mathbf{L}_S^T \end{bmatrix} &(21)\\
&= \begin{bmatrix} \mathbf{L}_A & 0 \\ \mathbf{B}^T\mathbf{L}_A^{-T} & \mathbf{L}_S \end{bmatrix} \begin{bmatrix} \mathbf{L}_A & 0 \\ \mathbf{B}^T\mathbf{L}_A^{-T} & \mathbf{L}_S \end{bmatrix}^T &(22)
\end{aligned}$$

Therefore, the original matrix could be expressed in the form of a lower triangular matrix and its transposition.

3 LU decomposition

In the previous section, we explained the block LU decomposition. In this section, we explain LU decomposition as an extension of the block LU decomposition.

In order to explain LU decomposition, we first explain LDU decomposition. LDU decomposition is easier than the LU decomposition because he operation is symmetry. Then, LU decomposition can be obtained by applying D to U after LDU decomposition or by applying D to L. Since LDU decomposition can be seen as repetitively applying the special case of block LDU decomposition, we explain it using it here.

3.1 LDU decomposition

LDU decomposition is to divide the matrix into products of lower triangular matrix, diagonal matrix, upper triangular matrix. In the block LDU decomposition, the original matrix is largely divided into blocks, and the entire decomposition is performed by calculation for each block. Consider block LDU decomposition that divides into blocks of size 1 and n-1 blocks instead of dividing blocks. By repeating this, LDU decomposition is obtained.

$$\mathbf{S}_0 = \begin{bmatrix} a_0 & \mathbf{b}_0^T \\ \mathbf{c}_0 & \mathbf{E}_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{a_0}\mathbf{c}_0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} a_0 & 0 \\ 0 & \mathbf{E}_0 - \frac{1}{a}\mathbf{c}_0\mathbf{b}_0^T \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{a_0}\mathbf{b}_0^T \\ 0 & \mathbf{I} \end{bmatrix} = \mathbf{L}_1\mathbf{D}_1\mathbf{U}_1 \tag{23}$$

Now,

$$\mathbf{D}_1 = \begin{bmatrix} a_0 & 0 \\ 0 & \mathbf{S}_1 \end{bmatrix}, \quad \mathbf{S}_1 = \mathbf{E}_0 - \frac{1}{a_0}\mathbf{c_0}\mathbf{b_0}^T \tag{24}$$

Met. Suppose $\mathbf{S}_1 = \mathbf{E}_0 - \frac{1}{a_0}\mathbf{c_0}\mathbf{b_0}^T$, which is the (n - 1) -th order square matrix at the bottom right, is LDU decomposed as follows.

$$\mathbf{S}_1 = \begin{bmatrix} a_1 & \mathbf{b}_1^T \\ \mathbf{c}_1 & \mathbf{E}_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{a_1}\mathbf{c}_1 & \mathbf{I} \end{bmatrix} \begin{bmatrix} a_1 & 0 \\ 0 & \mathbf{E}_1 - \frac{1}{a_1}\mathbf{c}_1\mathbf{b}_1^T \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{a_1}\mathbf{b}_1^T \\ 0 & \mathbf{I} \end{bmatrix} = \bar{\mathbf{L}}_2\bar{\mathbf{D}}_2\bar{\mathbf{U}}_2 \tag{25}$$

Using this, the original matrix $\mathbf{D}_1$ can be written as

$$\mathbf{D}_1 = \begin{bmatrix} a_0 & 0 \\ 0 & \bar{\mathbf{L}}_2\bar{\mathbf{D}}_2\bar{\mathbf{U}}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \bar{\mathbf{L}}_2 \end{bmatrix} \begin{bmatrix} a_0 & 0 \\ 0 & \bar{\mathbf{D}}_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \bar{\mathbf{U}}_2 \end{bmatrix} = \tilde{\mathbf{L}}_2\mathbf{D}_2\tilde{\mathbf{U}}_2 \tag{26}$$

Substituting this, the original matrix $\mathbf{S}_0$ can be written as

$$\mathbf{S}_0 = (\tilde{\mathbf{L}}_1\tilde{\mathbf{L}}_2)\mathbf{D}_2(\tilde{\mathbf{U}}_2\tilde{\mathbf{U}}_1) \tag{27}$$

Let's calculate concretely about $\mathbf{L}_2 = \tilde{\mathbf{L}}_1\tilde{\mathbf{L}}_2$.

$$\mathbf{L}_2 = \tilde{\mathbf{L}}_1\tilde{\mathbf{L}}_2 = \begin{bmatrix} 1 & 0 \\ \frac{1}{a_0}\mathbf{c}_0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \bar{\mathbf{L}}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{a_0}\mathbf{c}_0 & \bar{\mathbf{L}}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{a_0}\mathbf{c}_0 & \begin{bmatrix} 1 & 0 \\ \frac{1}{a_1}\mathbf{c}_1 & \mathbf{I} \end{bmatrix} \end{bmatrix} \tag{28}$$

It can be seen that this is a lower triangular matrix filled below the diagonal of the first two columns of the identity matrix. Likewise, concretely writing about $\mathbf{D}_2$ and U_2 as follows,

$$\mathbf{D}_2 = \begin{bmatrix} a_0 & 0 & 0 \\ 0 & a_1 & 0 \\ 0 & 0 & \mathbf{E}_1 - \frac{1}{a_1}\mathbf{c}_1\mathbf{b}_1^T \end{bmatrix} \tag{29}$$

$$\mathbf{U}_2 = \tilde{\mathbf{U}}_2\tilde{\mathbf{U}}_1 = \begin{bmatrix} 1 & \frac{1}{a_0}\mathbf{b}_0^T \\ 0 & \begin{bmatrix} 1 & \frac{1}{a_1}\mathbf{b}_1^T \\ 0 & \mathbf{I} \end{bmatrix} \end{bmatrix} \tag{30}$$

$\mathbf{D}_2$ is a diagonal block matrix, the first two are diagonal matrices, and the rest are block matrices. For $\mathbf{U}_2$, it turns out that it is an upper triangular matrix filled to the right from the diagonal of the two previous rows of the identity matrix.

3.1.1 Algorithm

Repeating these operations n times, which is the order of the original matrix, $\mathbf{D}_n$ becomes a diagonal matrix. When repeating n times, LDU decomposition is obtained. When the original matrix is S_0, the algorithm for obtaining these in order is as follows

1. Let i = 0
2. $\mathbf{S}_i$ is a square matrix of (n - i). Let a_i be the component of the first row and first column of $\mathbf{S}_i$. In the first row of $\mathbf{S}_i$, denote the components of the second and subsequent columns as $\mathbf{b}_i^T$. Likewise, $\mathbf{c}_i$ is expressed by vectorizing the components of the second and subsequent rows in one column of S_i. $\mathbf{b}_i, \mathbf{c}_i$ is the $(n-i-1)$ next vector. Also, $(n-i-1)$ consisting of the second and subsequent rows of S_i and the second and subsequent rows shall be $\mathbf{E}_i$ as the following square matrix. Let's be $S_{i+1} = \mathbf{E}_i - \frac{1}{a_i}\mathbf{c}_i\mathbf{b}_i^T$.
3. Assign a_i to the i-th diagonal of $\mathbf{D}$. Arrange $\frac{1}{a_i}\mathbf{c}_i$ below the diagonal of $\mathbf{L}$ in the i-th column. Arrange $\frac{1}{a_i}\mathbf{b}_i^T$ to the right from the diagonal of the $\mathbf{U}$ i-th line.
4. If $i < n$, add 1 to i and go back to 2.

This is shown in the figure below. Beginning with the original matrix $\mathbf{S}_0$, we proceed sequentially to the lower right.

3.1.2 Algorithm in detail

Let's write the above algorithm for each component. As the original matrix $\mathbf{A}$, the components of the LDU decomposed matrix are as follows. However, suppose that you write ij components of A, L, D, U in a lower-case letter as $a_{ij}, l_{il}, d_{ij}, u_{ij}$ respectively.

$$d_{ij} = \begin{cases} a_{ii} - \sum_{k=0}^{i-1} l_{ik}u_{ki}d_{kk} & (i = j) \\ 0 & (i \neq j) \end{cases} \quad (i = 0, 1, \ldots, n-1) \tag{31}$$

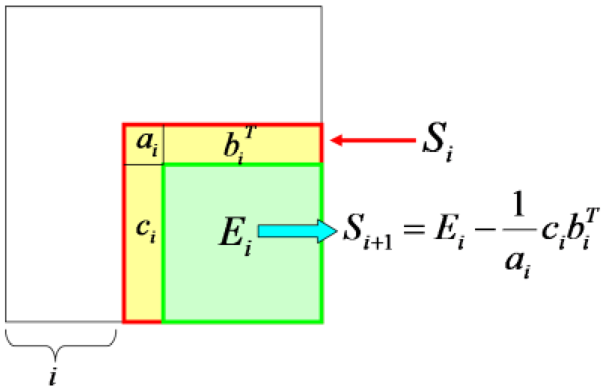

Figure 1: LDU Factorization Algorithm

$$l_{ij} = \begin{cases} \left(a_{ij} - \sum_{k=0}^{j-1} l_{ik} u_{kj} d_{kk}\right) / d_{jj} & (i > j) \\ 1 & (i = j) \\ 0 & (i < j) \end{cases} \qquad (i, j = 0, 1, \ldots, n-1) \tag{32}$$

$$u_{ij} = \begin{cases} \left(a_{ij} - \sum_{k=0}^{i-1} l_{ik} u_{kj} d_{kk}\right) / d_{ii} & (i < j) \\ 1 & (i = j) \\ 0 & (i > j) \end{cases} \qquad (i, j = 0, 1, \ldots, n-1) \tag{33}$$

3.2 LU decomposition

And the lower triangular matrix and the upper triangular matrix. This is called LU factorization. When **A** is asymmetric, it is called the Craut's decomposition. In the case of symmetry it is called the Cholesky decomposition.

3.2.1 LU decomposition part 1 (decomposition with 1 diagonal of L)

Let $\mathbf{A} = \mathbf{L}\bar{D}\bar{U}$ be the matrix decomposed using LDU decomposition in the previous section.

$$\mathbf{U} = \bar{D}\bar{U} \tag{34}$$

Then, since **U** is an upper triangular matrix,

$$\mathbf{A} = \mathbf{LU} \tag{35}$$

Let's write the above algorithm for each component with reference to LDU decomposition. As the original matrix $\mathbf{A} = \mathbf{L}\bar{\mathbf{D}}\bar{\mathbf{U}}$, the components of the LDU decomposed matrix are as follows. The following relational expression obtained for each component from $\mathbf{U} = \bar{\mathbf{D}}\bar{\mathbf{U}}$ $\bar{d}_{ii} = u_{ii}$, $u_{ij} = \bar{u}_{ij}\bar{d}_{ij}$ Then, the following is obtained.

$$l_{ij} = \begin{cases} \left(a_{ij} - \sum_{k=0}^{j-1} l_{ik}u_{kj}\right)/u_{jj} & (i > j) \\ 1 & (i = j) \\ 0 & (i < j) \end{cases} \quad (i, j = 0, 1, \ldots, n-1) \tag{36}$$

$$u_{ij} = \begin{cases} a_{ij} - \sum_{k=0}^{i-1} l_{ik}u_{kj} & (i \leq j) \\ 0 & (i > j) \end{cases} \quad (i, j = 0, 1, \ldots, n-1) \tag{37}$$

3.2.2 LU decomposition part 2 (decomposition with one diagonal of U)

Let $\mathbf{A} = \bar{\mathbf{L}}\bar{\mathbf{D}}\mathbf{U}$ be the matrix decomposed using LDU decomposition in the previous section.

$$\mathbf{L} = \bar{\mathbf{L}}\bar{\mathbf{D}} \tag{38}$$

Then, since $\mathbf{U}$ is an upper triangular matrix,

$$\mathbf{A} = \mathbf{L}\mathbf{U} \tag{39}$$

Let's write the above algorithm for each component with reference to LDU decomposition. As the original matrix $\mathbf{A} = \bar{\mathbf{L}}\bar{\mathbf{D}}\mathbf{U}$, the components of the LDU decomposed matrix are as follows. The following relational expression obtained for each component from $\mathbf{L} = \bar{\mathbf{L}}\bar{\mathbf{D}}$ $\bar{d}_{ii} = u_{ii}$, $u_{ij} = \bar{u}_{ij}\bar{d}_{ij}$ Then, the following is obtained.

$$l_{ij} = \begin{cases} a_{ij} - \sum_{k=0}^{j-1} l_{ik}u_{kj} & (i \geq j) \\ 0 & (i < j) \end{cases} \quad (i, j = 0, 1, \ldots, n-1) \tag{40}$$

$$u_{ij} = \begin{cases} \left(a_{ij} - \sum_{k=0}^{i-1} l_{ik}u_{kj}\right)/l_{ii} & (i < j) \\ 1 & (i = j) \\ 0 & (i > j) \end{cases} \quad (i, j = 0, 1, \ldots, n-1) \tag{41}$$

This is often used as a data structure of a matrix when CRS data format is used.

3.3 Solving a linear system

Let's solve a linear system as follows.

$$\mathbf{A}\mathbf{x} = \mathbf{y} \tag{42}$$

Here, if the coefficient matrix **A** is LU decomposed as

$$\mathbf{A} = \mathbf{L}\mathbf{U} \tag{43}$$

To solve this, we introduce the vector **z** as follows, first solve for the lower triangular matrix, then solve for the upper triangular matrix in order. When solving for the lower triangular matrix and the upper triangular matrix, solutions are obtained by forward substitution and backward substitution as described later.

$$\mathbf{z} = \mathbf{L}^{-1}\mathbf{y} \tag{44}$$

$$\mathbf{x} = \mathbf{U}^{-1}\mathbf{z} \tag{45}$$

3.3.1 How to find z by solving Lz = y

Can be done using forward substitution

$$\begin{aligned}
\mathbf{L}\mathbf{z} &= \mathbf{y} && (46)\\
l_{00}z_0 &= y_0 \quad\Leftrightarrow\quad z_0 = y_0/l_{00} && (47)\\
l_{10}z_0 + l_{11}z_1 &= y_1 \quad\Leftrightarrow\quad z_1 = \{y_1 - l_{10}z_0\}/l_{11} && (48)\\
l_{20}z_0 + l_{21}z_1 + l_{22}z_2 &= y_2 \quad\Leftrightarrow\quad z_2 = \{y_2 - l_{20}z_0 - l_{21}z_1\}/l_{22} && (49)\\
\sum_{i=0}^{k} l_{ki}z_i &= y_k \quad\Leftrightarrow\quad z_k = \{y_k - \sum_{i=0}^{k-1} l_{ki}z_i\}/l_{kk} \quad (k = 0, 1, \ldots, n-1) && (50)
\end{aligned}$$

3.3.2 How to obtain x by solving Ux = z

Can be obtained using backward elimination.

$$\mathbf{U}\mathbf{x} = \mathbf{z} \tag{51}$$

From the last element of x, we will seek to the first element in order

$$
\begin{aligned}
&u_{(n-1,n-1)}x_{n-1} = z_{n-1} \\
&\quad\Leftrightarrow \quad x_{n-1} = z_{n-1}/u_{(n-1,n-1)} && (52)\\
&u_{(n-2,n-2)}x_{n-2} + u_{(n-2,n-1)}x_{n-1} = z_{n-2} \\
&\quad\Leftrightarrow \quad x_{n-2} = \{z_{n-2} - u_{(n-2,n-1)}x_{n-1}\}/u_{(n-2,n-2)} && (53)\\
&u_{(n-3,n-3)}x_{n-3} + u_{(n-3,n-2)}x_{n-2} + u_{(n-3,n-1)}x_{n-1} = z_{n-3} \\
&\quad\Leftrightarrow \quad x_{n-3} = \{z_{n-3} - u_{(n-3,n-1)}x_{n-1} - u_{(n-3,n-2)}x_{n-2}\}/u_{(n-3,n-3)} && (54)\\
&\sum_{i=1}^{k} u_{(n-k,n-i)}x_{n-i} = z_{n-k} \\
&\quad\Leftrightarrow \quad x_{n-k} = \{\sum_{i=1}^{k} z_{n-k} - u_{(n-k,n-i)}x_{n-k}\}/u_{(n-k,n-k)} \quad (k = 1, 2, \ldots, n) && (55)
\end{aligned}
$$

3.4 Speed up compression display and calculation

In addition to the original matrix **A** before decomposition it is inefficient to reserve separate memory to express L and U. L is a lower triangular matrix, the upper half is 0, U is the lower triangular matrix, and the lower half is 0. By transforming the original matrix so that the upper half is U and the lower half is L, you can display them in one matrix. Let us now consider the case where LU factorization is performed so that the diagonal of U is 1. That is, $A = \left(\bar{L}\bar{D}\right)\mathbf{U} = LU$ transforms **A** at this time so that it becomes a matrix $\mathbf{A}'$ like the following. However, let izz components of $\mathbf{A}'$, **L**, **U** be a'_{ij}, l_{ij}, u_{ij} respectively.

$$
a'_{ij} = \begin{cases} l_{ij} & (i > j) \\ 1/l_{ii} & (i = j) \\ u_{ij} & (i < j) \end{cases} \qquad (56)
$$

Notice that the diagonal elements of A' are the inverse of the diagonal elements of **L**. This is because the diagonal elements of $\mathbf{A}'$ are used for LU decomposition and diagonal elements for forward substitution, but they are always used as opposite values, so you can expect high speed by storing the opposite value. Now, LU decomposition such that the diagonal of U is 1 is suitable for line by line decomposition. This can be said to be the most suitable decomposition method for row-wise data structures like CRS. The algorithm for finding such a matrix $\mathbf{A}'$ is as follows.

3.4.1 LU factorization (LU decomposition with U diagonal 1, compressed display with diagonal reciprocal)

1. $for \;\; i = 0, \ldots, n-1$

(a) $for \;\; j = 0, \ldots, n-1$

- $a'_{ij} = a_{ij} - \sum_{k=0}^{j-1} a'_{ik} a'_{kj}$

(b) $end \; for$

(c) $a'_{ii} = 1/a'_{ii}$ (Calculate and store the reciprocal of diagonal elements of lower triangular matrix)

(d) $for \;\; j = i+1, \ldots, n-1$

- $a'_{ij} = a'_{ij} \times a'_{ii}$ (Multiply diagonal elements of lower triangular matrix to create upper triangular matrix)

(e) $end \; for$

2. $end \; for$

Now, the forward elimination and the backward substitution for the deformed matrix are as follows.

3.4.2 Forward substitution (LU decomposition with U diagonal 1, compressed display with diagonal reciprocal)

$$z_k = \{y_k - \sum_{i=0}^{k-1} a'_{ki} z_i\} a'_{kk} \quad (k = 0, 1, \ldots, n-1) \tag{57}$$

3.4.3 Back substitution (LU decomposition with U diagonal 1, compressed display with diagonal reciprocal)

$$x_{n-k} = \sum_{i=1}^{k} z_{n-k} - a'_{(n-k,n-i)} x_{n-i} \quad (k = 1, 2, \ldots, n) \tag{58}$$

3.4.4 LU decomposition algorithm diagram (compressed display)

FIG. 2 shows an LU decomposition algorithm when compressed display is performed. It does not depend on the type of LU decomposition or the decomposition procedure, and it is roughly as shown in the following figure. In other words, if there is an element a_{ij} to decompose, refer to all the elements of textbf decomposed of a_{kj} and left a_{ik} so that the diagonal elements a_{kk} and a_{ij} can make a rectangle.

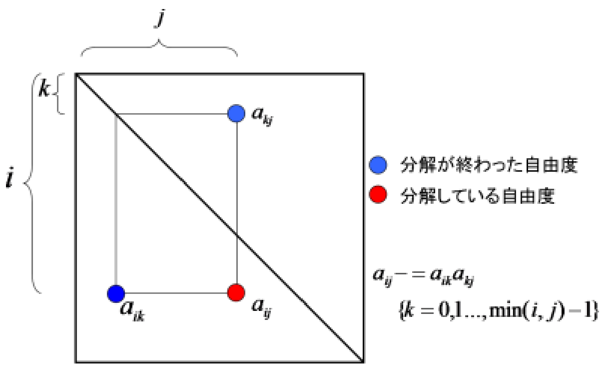

Figure 2: illustragtion of LU factorization where L and U are put in the same matrix.