

# LU分解\*

梅谷 信行

平成22年11月3日

## 目次

|          |                        |          |
|----------|------------------------|----------|
| <b>1</b> | <b>概要</b>              | <b>2</b> |
| <b>2</b> | <b>ブロックLU分解</b>        | <b>2</b> |
| 2.1      | ブロックLU分解               | 2        |
| 2.1.1    | ブロック下三角行列の逆行列          | 3        |
| 2.1.2    | 連立一次方程式のブロックごとの計算による求解 | 3        |
| 2.1.3    | ブロック三角行列の固有値           | 4        |
| 2.2      | ブロックLDU分解              | 4        |
| 2.3      | $LL^T$ 分解のブロックでの計算     | 5        |
| <b>3</b> | <b>LU分解</b>            | <b>6</b> |
| 3.1      | LDU分解                  | 6        |
| 3.1.1    | アルゴリズム                 | 7        |
| 3.1.2    | 成分の計算                  | 8        |
| 3.2      | LU分解                   | 9        |
| 3.2.1    | LU分解その1 (Lの対角が1な分解)    | 9        |
| 3.2.2    | LU分解その2 (Uの対角が1な分解)    | 9        |
| 3.3      | 連立一次方程式の求解             | 10       |
| 3.3.1    | $Lz=y$ を解いて $z$ を求める方法 | 11       |
| 3.3.2    | $Ux=z$ を解いて $x$ を求める方法 | 11       |

---

\*これは忘れっぽい著者が昔勉強したことを書き留めておく備忘録です。きっと間違いを多く含んでいます。すいません。ご指摘ご意見などありましたら教えてもらえると有り難いです。

|       |                                |    |
|-------|--------------------------------|----|
| 3.4   | 圧縮表示と計算の高速化 . . . . .          | 11 |
| 3.4.1 | LU分解 (Uの対角が1のLU分解, 対角が逆数の圧縮表示) | 12 |
| 3.4.2 | 前進代入 (Uの対角が1のLU分解, 対角が逆数の圧縮表示) | 13 |
| 3.4.3 | 後退代入 (Uの対角が1のLU分解, 対角が逆数の圧縮表示) | 13 |
| 3.4.4 | LU分解アルゴリズム図 (圧縮表示) . . . . .   | 13 |
| 4     | 参考にしたもの                        | 13 |

## 1 概要

行列を解くのに直接法の一つであるLU分解を使うことが多い。LU分解とは係数行列を下三角行列と上三角行列の積で表す方法である。LU分解の後は三角行列であることを利用して、前進・後退代入を使って高速に連立一次方程式を解くことができる。ここではLU分解の方法について簡単に述べる。LU分解を説明するにあたって、ブロックLU分解をまず説明する。LU分解はブロックLU分解の特殊な場合を何度も繰り返すことで得られる。

## 2 ブロックLU分解

### 2.1 ブロックLU分解

次のような行列を考えよう。

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{E} \end{bmatrix} \quad (1)$$

この行列は次のように分解できることが簡単にわかる。

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

これはブロックLU分解の一種(下行列の対角ブロックが単位行列となるもの)で、呼ぶ。ブロック上三角行列の右下に現れる項

$$\mathbf{E} - \mathbf{CA}^{-1}\mathbf{B} = \mathbf{S} \quad (3)$$

をシュール補元 (Shur Complement) と呼び、 $\mathbf{S}$  と書くこともある。

### 2.1.1 ブロック下三角行列の逆行列

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

が成り立つことから,

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

であることが分かる.

### 2.1.2 連立一次方程式のブロックごとの計算による求解

次のような連立一次方程式を考えよう.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{E} \end{bmatrix} \begin{Bmatrix} x_A \\ x_B \end{Bmatrix} = \begin{Bmatrix} y_A \\ y_B \end{Bmatrix} \quad (6)$$

係数行列についてブロックLU分解をすると, 次のようになる. 但し Shur Complement を  $\mathbf{S}$  と書いた.

$$\begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{CA}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{S} \end{bmatrix} \begin{Bmatrix} x_A \\ x_B \end{Bmatrix} = \begin{Bmatrix} y_A \\ y_B \end{Bmatrix} \quad (7)$$

左から下三角行列の逆行列を掛けて,

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{S} \end{bmatrix} \begin{Bmatrix} x_A \\ x_B \end{Bmatrix} = \begin{Bmatrix} y_A \\ y_B \end{Bmatrix} \quad (8)$$

$$x_B = \begin{bmatrix} \mathbf{I} & 0 \\ -\mathbf{CA}^{-1} & I \end{bmatrix} \begin{Bmatrix} y_A \\ y_B \end{Bmatrix} \quad (9)$$

$$y_B \quad (10)$$

よって, 次のように解が求まる

$$\mathbf{x}_B = \mathbf{S}^{-1}\{-\mathbf{CAy}_A + \mathbf{y}_B\} \quad (11)$$

$$\mathbf{x}_A = \mathbf{A}^{-1}\{\mathbf{y}_A - \mathbf{Bx}_B\} \quad (12)$$

逆をとっているのは、 $\mathbf{A}$ と $\mathbf{S}$ である。Aの逆が簡単に取れるような状況（例えばAが対角行列）ではこのようなブロック分解における解法は解く行列の次数を減らすことができ、効率が良い。

### 2.1.3 ブロック三角行列の固有値

上のブロックLU分解の結果である、次のブロック上三角行列の固有値を求めよう。

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \phi = \lambda \phi \quad (13)$$

特性方程式は次のとおり

$$\det \begin{bmatrix} \mathbf{A} - \lambda \mathbf{I} & \mathbf{B} \\ \mathbf{0} & \mathbf{S} - \lambda \mathbf{I} \end{bmatrix} = 0 \quad (14)$$

$$\Leftrightarrow \det(\mathbf{A} - \lambda \mathbf{I}) \det(\mathbf{S} - \lambda \mathbf{I}) = 0 \quad (15)$$

よって、固有値 $\lambda$ は $\mathbf{A}$ の固有値、 $\mathbf{S}$ の固有値に等しくなる。**ブロック三角行列の固有値は対角に位置する行列の固有値に等しい**ことが分かる。また、対角行列が単位行列であるブロック三角行列の固有値は1であることがわかる。以上から、もとの行列の固有値は $\mathbf{A}$ と、 $\mathbf{S}$ の固有値に等しくなる Shur Complement  $\mathbf{S}$ は元の行列の性質を保持している、ので非常に重要である。

## 2.2 ブロックLDU分解

LU分解のブロック上三角行列をさらに次のように分解してみよう。

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{E} - \mathbf{CA}^{-1}\mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} - \mathbf{CA}^{-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (16)$$

これを用いると次のように対称に書ける.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{E} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{CA}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix} \quad (17)$$

### 2.3 $LL^T$ 分解のブロックでの計算

もとの行列が正定値対称行列である場合を考える. つまり,

$$\mathbf{A}^T = \mathbf{A} \quad (18)$$

$$\mathbf{C} = \mathbf{B}^T \quad (19)$$

$$\mathbf{E}^T = \mathbf{E} \quad (20)$$

もとの行列の固有値は  $\mathbf{A}$  と  $\mathbf{S}$  の固有値に等しいので, もとの行列が正定値であるということは  $\mathbf{A}$  も  $\mathbf{S}$  も正定値ということである. このとき次のような実行列  $L_A, L_S$  による分解が可能である.  $\mathbf{A} = L_A L_A^T$ ,  $\mathbf{S} = L_S L_S^T$  これを使うとLDU分解の対角ブロック行列は次のように分解できる.

$$\begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{S} \end{bmatrix} = \begin{bmatrix} L_A & 0 \\ 0 & L_S \end{bmatrix} \begin{bmatrix} L_A^T & 0 \\ 0 & L_S^T \end{bmatrix} \quad (21)$$

これを上に代入して,

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{E} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{B}^T \mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} L_A & 0 \\ 0 & L_S \end{bmatrix} \begin{bmatrix} L_A^T & 0 \\ 0 & L_S^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix} \quad (22)$$

$$= \begin{bmatrix} L_A & 0 \\ \mathbf{B}^T L_A^{-T} & L_S \end{bmatrix} \begin{bmatrix} L_A^T & L_A^{-T} \mathbf{B} \\ 0 & L_S^T \end{bmatrix} \quad (23)$$

$$= \begin{bmatrix} L_A & 0 \\ \mathbf{B}^T L_A^{-T} & L_S \end{bmatrix} \begin{bmatrix} L_A & 0 \\ \mathbf{B}^T L_A^{-T} & L_S \end{bmatrix}^T \quad (24)$$

よって, もとの行列を下三角行列とその転置の形で表すことができた.

### 3 LU分解

前節ではブロックLU分解について解説した。ここではそれを用いてLU分解を説明する。LU分解を説明するためにまず、LDU分解について解説する。LDU分解は処理に対称性があるので、理解しやすい。LDU分解した後にDをUにかけたり、DをLにかけたりすることで、LU分解が得られる。LDU分解はブロックLDU分解の特殊な場合を繰り返した物として考えることができるので、ここではそれを使って解説する。

#### 3.1 LDU分解

LDU分解は行列を下三角行列、対角行列、上三角行列の積に分けることである。ブロックLDU分解は、もとの行列を大きくブロックに分けて、ブロックごとの計算で全体の分解を行った。ブロックの分け方を代えて、サイズが1のブロックと、n-1のブロックに分けるようなブロックLDU分解を考える。これを繰り返すことでLDU分解が得られる。

$$\mathbf{S}_0 = \begin{bmatrix} a_0 & \mathbf{b}_0^T \\ \mathbf{c}_0 & \mathbf{E}_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{a_0} \mathbf{c}_0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} a_0 & 0 \\ 0 & \mathbf{E}_0 - \frac{1}{a_0} \mathbf{c}_0 \mathbf{b}_0^T \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{a_0} \mathbf{b}_0^T \\ 0 & \mathbf{I} \end{bmatrix} = \mathbf{L}_1 \mathbf{D}_1 \mathbf{U}_1 \quad (25)$$

さて、

$$\mathbf{D}_1 = \begin{bmatrix} a_0 & 0 \\ 0 & \mathbf{S}_1 \end{bmatrix}, \quad \mathbf{S}_1 = \mathbf{E}_0 - \frac{1}{a_0} \mathbf{c}_0 \mathbf{b}_0^T \quad (26)$$

であった。この右下の(n-1)次の正方行列である $\mathbf{S}_1 = \mathbf{E}_0 - \frac{1}{a_0} \mathbf{c}_0 \mathbf{b}_0^T$ が次のようにLDU分解されるとする。

$$\mathbf{S}_1 = \begin{bmatrix} a_1 & \mathbf{b}_1^T \\ \mathbf{c}_1 & \mathbf{E}_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{a_1} \mathbf{c}_1 & \mathbf{I} \end{bmatrix} \begin{bmatrix} a_1 & 0 \\ 0 & \mathbf{E}_1 - \frac{1}{a_1} \mathbf{c}_1 \mathbf{b}_1^T \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{a_1} \mathbf{b}_1^T \\ 0 & \mathbf{I} \end{bmatrix} = \tilde{\mathbf{L}}_2 \tilde{\mathbf{D}}_2 \tilde{\mathbf{U}}_2 \quad (27)$$

これを使うと、もとの行列 $\mathbf{D}_1$ は次のように書ける。

$$\mathbf{D}_1 = \begin{bmatrix} a_0 & 0 \\ 0 & \tilde{\mathbf{L}}_2 \tilde{\mathbf{D}}_2 \tilde{\mathbf{U}}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{\mathbf{L}}_2 \end{bmatrix} \begin{bmatrix} a_0 & 0 \\ 0 & \tilde{\mathbf{D}}_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{\mathbf{U}}_2 \end{bmatrix} = \tilde{\mathbf{L}}_2 \mathbf{D}_2 \tilde{\mathbf{U}}_2 \quad (28)$$

これを代入すると，もとの行列  $\mathbf{S}_0$  は次のように書ける

$$\mathbf{S}_0 = (\tilde{\mathbf{L}}_1 \tilde{\mathbf{L}}_2) \mathbf{D}_2 (\tilde{\mathbf{U}}_2 \tilde{\mathbf{U}}_1) \quad (29)$$

ここで， $\mathbf{L}_2 = \tilde{\mathbf{L}}_1 \tilde{\mathbf{L}}_2$  について具体的に計算してみよう．

$$\mathbf{L}_2 = \tilde{\mathbf{L}}_1 \tilde{\mathbf{L}}_2 = \begin{bmatrix} 1 & 0 \\ \frac{1}{a_0} \mathbf{c}_0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{\mathbf{L}}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{a_0} \mathbf{c}_0 & \tilde{\mathbf{L}}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{a_0} \mathbf{c}_0 & \begin{bmatrix} 1 & 0 \\ \frac{1}{a_1} \mathbf{c}_1 & \mathbf{I} \end{bmatrix} \end{bmatrix} \quad (30)$$

これは単位行列の前2列の対角より下が埋まった下三角行列になっていることがわかる．同様にして， $\mathbf{D}_2 \mathbf{U}_2$  について具体的に書くと次のとおり，

$$\mathbf{D}_2 = \begin{bmatrix} a_0 & 0 & 0 \\ 0 & a_1 & 0 \\ 0 & 0 & \mathbf{E}_1 - \frac{1}{a_1} \mathbf{c}_1 \mathbf{b}_1^T \end{bmatrix} \quad (31)$$

$$\mathbf{U}_2 = \tilde{\mathbf{U}}_2 \tilde{\mathbf{U}}_1 = \begin{bmatrix} 1 & \frac{1}{a_0} \mathbf{b}_0^T \\ 0 & \begin{bmatrix} 1 & \frac{1}{a_1} \mathbf{b}_1^T \\ 0 & \mathbf{I} \end{bmatrix} \end{bmatrix} \quad (32)$$

$\mathbf{D}_2$  は対角ブロック行列で，始めの2つは対角行列で，以降はブロック行列となっている． $\mathbf{U}_2$  については単位行列の前2行の対角より右が埋まった上三角行列になっていることがわかる．

### 3.1.1 アルゴリズム

これらの作業をもとの行列の次数である  $n$  回繰り返せば， $\mathbf{D}_n$  が対角行列になる． $n$  回繰り返した時に  $\mathbf{L D U}$  分解が得られる．もとの行列を  $\mathbf{S}_0$  とするとき，順番にこれらを求めるアルゴリズムは次のとおり

1.  $i=0$  とする
2.  $S_i$  は  $(n-i)$  次の正方行列である． $S_i$  の1行1列目の成分を  $a_i$  とする． $S_i$  の1行目で2列目以降の成分をベクトル表記して  $\mathbf{b}_i^T$  とする．同様に  $S_i$  の1列で2行目以降の成分をベクトル表記して  $\mathbf{c}_i$  とする． $\mathbf{b}_i, \mathbf{c}_i$  は  $(n-i-1)$  次のベクトルである．また  $S_i$  の2行目以降，2列目以降からなる  $(n-i-1)$  次の正方行列を  $\mathbf{E}_i$  とする． $S_{i+1} = \mathbf{E}_i - \frac{1}{a_i} \mathbf{c}_i \mathbf{b}_i^T$  とする．

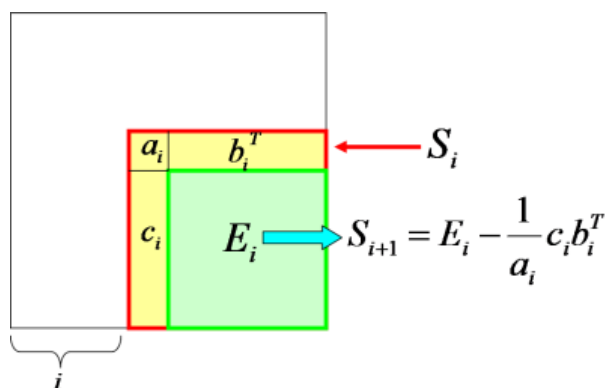


図 1: LDU分解のアルゴリズム

3.  $a_i$  を  $\mathbf{D}$  の  $i$  番目の対角に代入.  $\frac{1}{a_i} \mathbf{c}_i$  を  $\mathbf{L}$  の  $i$  列目の対角の下に並べる.  $\frac{1}{a_i} \mathbf{b}_i^T$  を  $\mathbf{U}$  の  $i$  行目の対角から右に並べる
4.  $i < n$  なら  $i$  に 1 を加えて 2 へ戻る.

これを図示すると下の図のようになる. 元の行列  $\mathbf{S}_0$  から始まり, 右下へ向けて逐次的に計算を進めていく.

### 3.1.2 成分の計算

上のアルゴリズムを成分毎に書いてみよう. 元の行列  $\mathbf{A}$  として, LDU分解した行列の成分は次のようになる. 但し,  $\mathbf{A}, \mathbf{L}, \mathbf{D}, \mathbf{U}$  の  $ij$  成分をそれぞれ  $a_{ij}$  のように書くとする.

$$d_{ij} = \begin{cases} a_{ii} - \sum_{k=0}^{i-1} l_{ik} u_{ki} d_{kk} & (i = j) \\ 0 & (i \neq j) \end{cases} \quad (i = 0, 1, \dots, n-1) \quad (33)$$

$$l_{ij} = \begin{cases} (a_{ij} - \sum_{k=0}^{j-1} l_{ik} u_{kj} d_{kk}) / d_{jj} & (i > j) \\ 1 & (i = j) \\ 0 & (i < j) \end{cases} \quad (i, j = 0, 1, \dots, n-1) \quad (34)$$

$$u_{ij} = \begin{cases} (a_{ij} - \sum_{k=0}^{i-1} l_{ik} u_{kj} d_{kk}) / d_{ii} & (i < j) \\ 1 & (i = j) \\ 0 & (i > j) \end{cases} \quad (i, j = 0, 1, \dots, n-1) \quad (35)$$



## 3.2 LU分解

と下三角行列と上三角行列の積になる。これをLU分解という。Aが非対称な場合はクラウト分解，対称の場合はコレスキー分解と呼ばれる。

### 3.2.1 LU分解その1 (Lの対角が1な分解)

前節でLDU分解を用いて分解した行列を  $\mathbf{A} = \mathbf{L}\bar{\mathbf{D}}\bar{\mathbf{U}}$  とする。

$$\mathbf{U} = \bar{\mathbf{D}}\bar{\mathbf{U}} \quad (36)$$

とすると，Uは上三角行列となるので，

$$\mathbf{A} = \mathbf{LU} \quad (37)$$

LDU分解を参考にして上のアルゴリズムを成分毎に書いてみよう。元の行列  $\mathbf{A} = \mathbf{L}\bar{\mathbf{D}}\bar{\mathbf{U}}$  として，LDU分解した行列の成分は次のようになる。  $\mathbf{U} = \bar{\mathbf{D}}\bar{\mathbf{U}}$  から成分ごとに得られる次の関係式  $\bar{d}_{ii} = u_{ii}$ ，  $u_{ij} = \bar{u}_{ij}\bar{d}_{ij}$  を代入して，次が得られる。

$$l_{ij} = \begin{cases} (a_{ij} - \sum_{k=0}^{j-1} l_{ik}u_{kj})/u_{jj} & (i > j) \\ 1 & (i = j) \\ 0 & (i < j) \end{cases} \quad (i, j = 0, 1, \dots, n-1) \quad (38)$$

$$u_{ij} = \begin{cases} a_{ij} - \sum_{k=0}^{i-1} l_{ik}u_{kj} & (i \leq j) \\ 0 & (i > j) \end{cases} \quad (i, j = 0, 1, \dots, n-1) \quad (39)$$

### 3.2.2 LU分解その2 (Uの対角が1な分解)

前節でLDU分解を用いて分解した行列を  $\mathbf{A} = \bar{\mathbf{L}}\bar{\mathbf{D}}\mathbf{U}$  とする。

$$\mathbf{L} = \bar{\mathbf{L}}\bar{\mathbf{D}} \quad (40)$$

とすると，Lは上三角行列となるので，

$$\mathbf{A} = \mathbf{LU} \quad (41)$$

LDU 分解を参考にして上のアルゴリズムを成分毎に書いてみよう。元の行列  $\mathbf{A} = \bar{\mathbf{L}}\bar{\mathbf{D}}\mathbf{U}$  として、LDU 分解した行列の成分は次のようになる。 $\mathbf{L} = \bar{\mathbf{L}}\bar{\mathbf{D}}$  から成分ごとに得られる次の関係式  $\bar{d}_{ii} = u_{ii}$ ,  $u_{ij} = \bar{u}_{ij}\bar{d}_{ij}$  を代入して、次が得られる。

$$l_{ij} = \begin{cases} a_{ij} - \sum_{k=0}^{j-1} l_{ik}u_{kj} & (i \geq j) \\ 0 & (i < j) \end{cases} \quad (i, j = 0, 1, \dots, n-1) \quad (42)$$

$$u_{ij} = \begin{cases} (a_{ij} - \sum_{k=0}^{i-1} l_{ik}u_{kj})/l_{ii} & (i < j) \\ 1 & (i = j) \\ 0 & (i > j) \end{cases} \quad (i, j = 0, 1, \dots, n-1) \quad (43)$$

これは行列のデータ構造として、CRS データ形式を用いる場合に良く使われる。

### 3.3 連立一次方程式の求解

次のような連立一次方程式を解くとする。

$$\mathbf{Ax} = \mathbf{y} \quad (44)$$

ここで係数が次のように LU 分解されているとすると、

$$\mathbf{A} = \mathbf{LU} \quad (45)$$

これを解く際には、次のようにベクトル  $\mathbf{z}$  を導入して、まず下三角行列について解き、次に上三角行列について順番に解くことで得られる。下三角行列や、上三角行列について解くときには、後に述べるように前進代入、後退代入によって解が得られる。

$$\mathbf{z} = \mathbf{L}^{-1}\mathbf{y} \quad (46)$$

$$\mathbf{x} = \mathbf{U}^{-1}\mathbf{z} \quad (47)$$

### 3.3.1 $Lz=y$ を解いて $z$ を求める方法

前進代入を用いてできる

$$\mathbf{Lz} = \mathbf{y} \quad (48)$$

$$l_{00}z_0 = y_0 \Leftrightarrow z_0 = y_0/l_{00} \quad (49)$$

$$l_{10}z_0 + l_{11}z_1 = y_1 \Leftrightarrow z_1 = \{y_1 - l_{10}z_0\}/l_{11} \quad (50)$$

$$l_{20}z_0 + l_{21}z_1 + l_{22}z_2 = y_2 \Leftrightarrow z_2 = \{y_2 - l_{20}z_0 - l_{21}z_1\}/l_{22} \quad (51)$$

$$\sum_{i=0}^k l_{ki}z_i = y_k \Leftrightarrow z_k = \{y_k - \sum_{i=0}^{k-1} l_{ki}z_i\}/l_{kk} \quad (k = 0, 1, \dots, n-1) \quad (52)$$

### 3.3.2 $Ux=z$ を解いて $x$ を求める方法

後退消去を用いて求めることができる。

$$\mathbf{Ux} = \mathbf{z} \quad (53)$$

$x$  の最後の要素から、順番に最初の要素に向けて求めていく

$$\begin{aligned} u_{(n-1,n-1)}x_{n-1} &= z_{n-1} \\ \Leftrightarrow x_{n-1} &= z_{n-1}/u_{(n-1,n-1)} \end{aligned} \quad (54)$$

$$\begin{aligned} u_{(n-2,n-2)}x_{n-2} + u_{(n-2,n-1)}x_{n-1} &= z_{n-2} \\ \Leftrightarrow x_{n-2} &= \{z_{n-2} - u_{(n-2,n-1)}x_{n-1}\}/u_{(n-2,n-2)} \end{aligned} \quad (55)$$

$$\begin{aligned} u_{(n-3,n-3)}x_{n-3} + u_{(n-3,n-2)}x_{n-2} + u_{(n-3,n-1)}x_{n-1} &= z_{n-3} \\ \Leftrightarrow x_{n-3} &= \{z_{n-3} - u_{(n-3,n-1)}x_{n-1} - u_{(n-3,n-2)}x_{n-2}\}/u_{(n-3,n-3)} \end{aligned} \quad (56)$$

$$\begin{aligned} \sum_{i=1}^k u_{(n-k,n-i)}x_{n-i} &= z_{n-k} \\ \Leftrightarrow x_{n-k} &= \left\{ \sum_{i=1}^k z_{n-k} - u_{(n-k,n-i)}x_{n-i} \right\} / u_{(n-k,n-k)} \quad (k = 1, 2, \dots, n) \end{aligned} \quad (57)$$

## 3.4 圧縮表示と計算の高速化

分解する前のもとの行列  $\mathbf{A}$  に加え、 $\mathbf{L}$  と  $\mathbf{U}$  を表現するためのメモリを別に確保することは効率が悪い。 $\mathbf{L}$  は下三角行列で、上半分は 0 であるし、 $\mathbf{U}$  は下三角行

列で下半分は0である。もとの行列を変形して、上半分がUになるようにして、下半分がLになるようにすると、これらを1つの行列の中に表示することができる。いまここで、Uの対角が1になるようにLU分解を行う場合を考える。つまり  $A = (\bar{L}\bar{D})U = LU$  のとき  $A$  を変形して次のような行列  $A'$  になるようにする。但し、 $A'$ ,  $L$ ,  $U$  の  $ij$  成分をそれぞれ  $a'_{ij}$ ,  $l_{ij}$ ,  $u_{ij}$  とする。

$$a'_{ij} = \begin{cases} l_{ij} & (i > j) \\ 1/l_{ii} & (i = j) \\ u_{ij} & (i < j) \end{cases} \quad (58)$$

$A'$  の対角成分が  $L$  の対角成分の逆になっていることに注目されたい。これは  $A'$  の対角成分はLU分解や対角成分は前進代入の際に使われるが、常に逆の値として使われるので、逆の値を保存しておくことで高速化が期待できる。さて、Uの対角が1になるようなLU分解は行ごとの分解に向いている。これはCRSのような行ごとのデータ構造にとって、最も適した分解方法だと言える。このような行列  $A'$  を求めるアルゴリズムは次のとおりとなる。

### 3.4.1 LU分解 (Uの対角が1のLU分解, 対角が逆数の圧縮表示)

1. *for*  $i = 0, \dots, n-1$ 
  - (a) *for*  $j = 0, \dots, n-1$ 
    - $a'_{ij} = a_{ij} - \sum_{k=0}^{j-1} a'_{ik} a'_{kj}$
  - (b) *end for*
  - (c)  $a'_{ii} = 1/a'_{ii}$  (下三角行列の対角成分の逆数を計算して格納)
  - (d) *for*  $j = i+1, \dots, n-1$ 
    - $a'_{ij} = a'_{ij} \times a'_{ii}$  (下三角行列の対角成分を掛けて上三角行列を作る)
  - (e) *end for*
2. *end for*

さて、このように変形した行列に対する前進消去、後退代入は次のようになる。

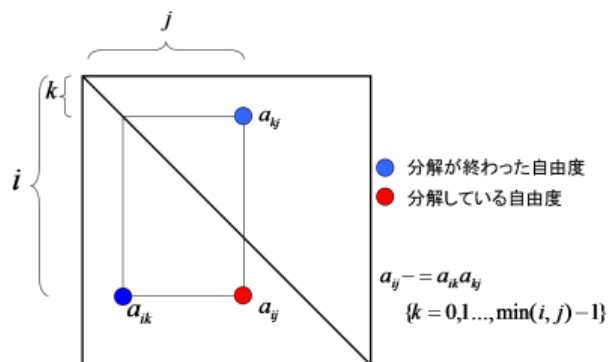


図 2: L と U の行列を圧縮表示した時の LU 分解の図

### 3.4.2 前進代入 (U の対角が 1 の LU 分解, 対角が逆数の圧縮表示)

$$z_k = \{y_k - \sum_{i=0}^{k-1} a'_{ki} z_i\} a'_{kk} \quad (k = 0, 1, \dots, n-1) \quad (59)$$

### 3.4.3 後退代入 (U の対角が 1 の LU 分解, 対角が逆数の圧縮表示)

$$x_{n-k} = \sum_{i=1}^k z_{n-k} - a'_{(n-k, n-i)} x_{n-i} \quad (k = 1, 2, \dots, n) \quad (60)$$

### 3.4.4 LU 分解アルゴリズム図 (圧縮表示)

図 2 は、圧縮表示をした場合の LU 分解アルゴリズムを示している。LU 分解の種類や分解の手順に依存せず、だいたい次のような図のとおりになる。つまり、分解する要素  $a_{ij}$  があった場合に、対角要素  $a_{kk}$  と  $a_{ij}$  が四角形を作れるように、その上  $a_{kj}$  や左  $a_{ik}$  の分解が終わった要素全てを参照する。

## 4 参考にしたもの

[1],[2],[3],[4]

## 参考文献

- [1] 村田健郎, J.J. ドンガラ, 小国力, 三好俊郎, 長谷川秀彦: 行列計算ソフトウェア - WS、スーパーコン、並列計算機, 丸善 (1991).
- [2] Saad, Y.: *Iterative Methods for Sparse Linear Systems, Second Edition*, Society for Industrial and Applied Mathematics, 2 edition (2003).
- [3] Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C. and Vorst, van der H.: *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods (Miscellaneous Titles in Applied Mathematics Series No 43)*, Society for Industrial Mathematics (1987).
- [4] 渡辺浩志: 非線形有限要素法特論, <http://www.sml.k.u-tokyo.ac.jp/members/nabe/>.