

第7章 連立一次方程式の解法1 — 直接法

数値計算は線形代数に依存するところが大きく、この高度に発展した数値線型代数という主題は、初めから数値解析の核心であった。安定性・良条件・後退誤差解析といった現在では標準的になっている概念は、数値線形代数という主題のもとで定義され、研ぎ澄まされてきた。こうした発展の中心人物は、1950年代からその死の1986年に至るまで、ウィルキンソン (Jim Wilkinson) であった。

L. N. Trefethen(岡田・三井 訳)
「数値解析の定義」(応用数理 Vol.3, No.2)

7.1 連立一次方程式とその数値計算法

後の章を通じて扱う連立一次方程式 (linear system of equations) を

$$\mathbf{Ax} = \mathbf{b} \quad (7.1)$$

と書くことにする。ここで

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \in M_n(\mathbb{R}), \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \in \mathbb{R}^n, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$$

であり、 A と \mathbf{b} がそれぞれ既知であるとき、(7.1) を満足する \mathbf{x} を求める問題である。これら連立一次方程式を構成するパーツを

A : 係数行列 — Coefficient Matrix

\mathbf{b} : 定数項, 定数ベクトル — Constant Vector

\mathbf{x} : 解, 解ベクトル — Solution

と呼ぶことにする。もし、 $A \in M_n(\mathbb{C})$ または $\mathbf{b} \in \mathbb{C}^n$ であれば、 $\mathbf{x} \in \mathbb{C}^n$ であることが期待されるが、本書では実数要素のみから成り立つ連立一次方程式を扱うことにする。

さて、線型代数の講義では(7.1)には次の3つのケースがあることを教えられたはずである。

- a) \mathbf{x} は一意に定まる
- b) \mathbf{x} は無数に存在する

c) (7.1) を満足する \mathbf{x} は存在しない

c) は (7.1) の設定が不適切ということになるので、a) と b) の場合のみを考える。このどちらになるのかは、正方行列 A の性質にのみ依存して決定されることもよくご存知であろう。具体的に言うと、 A の行列式 (determinant) $|A|$ が $|A| \neq 0$ であれば、逆行列 A^{-1} が存在する。この時、行列 A は正則 (行列) であると呼ぶ。逆行列とは元の行列との積を取ると単位行列 $I \in M_n(\mathbb{R})$ になるものをいう。

$$AA^{-1} = A^{-1}A = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} = I$$

この性質を利用すれば、(7.1) の両辺に A^{-1} を左から乗じて

$$\begin{aligned} A^{-1}A\mathbf{x} &= A^{-1}\mathbf{b} \\ \mathbf{x} &= A^{-1}\mathbf{b} \end{aligned} \tag{7.2}$$

と変形でき、一意な解 $A^{-1}\mathbf{b}$ が得られる。

もし $|A| = 0$ であれば、 A のランク $\text{rank}(A)$ が n 未満となり、 \mathbf{x} には $n - \text{rank}(A)$ 分の自由度が生まれて無数の解が存在することになる。これも数学的には意味があり、行列の固有値・固有ベクトルを求める際にはこのような連立一次方程式を解く必要が出てくる (第11章参照)。本章では a) の場合のみを考えることにする。

(7.2) 式は理論的表現としては良いが、実際に逆行列を求めて定数ベクトルとの積を取る、という方法を取ると、後述する LU 分解を経由する方法に比べて計算量が増えてしまうことが知られている。逆行列そのものを必要とする場合を除き、 $n \gg 1$ となる大次元の問題には用いるべきではない。

連立一次方程式はさまざまな場面で登場し、特に応用的には重要な常微分方程式 (第16, 18章)、偏微分方程式 (第19章) との関わりが深い。よって、行列の構造や問題、精度、コンピュータ環境に適した多種多様な解法が提案されている。大まかには次の3つのカテゴリに分類される。

- 直接法 (direct method) … A を直接変形して、有限回の演算で \mathbf{x} を求める方法。本章で取り扱う。
- 反復法 (iterative method) … (7.1) を変形してベクトル列を得る漸化式を導き、徐々に \mathbf{x} に収束させていく方法。第9章で扱う。
- Krylov 部分空間法 (Krylov subspace method) … Krylov 部分空間 $\subset \mathbb{C}^n$ or \mathbb{R}^n に属するベクトル列を生成しながら、「理論的には」有限回の反復を経て \mathbf{x} を求める方法。第10章で扱う。

7.2 LU 分解, LDU 分解

現在標準的な直接法の一つがこの LU 分解法である。考え方は Gauss の消去法 (掃き出し法ではない (後述)) と似ているが、非常に利用価値の高い考え方を提示してくれる方法である。

まず, LU 分解法の考え方を説明する。LU 分解法の L と U はそれぞれ下三角行列 (lower triangular matrix), 上三角行列 (upper triangular matrix) を意味する。具体的には

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ \vdots & \ddots & & \\ l_{n1} & \cdots & l_{nn} & \end{bmatrix} = [l_{ij}], l_{ij} = \begin{cases} 0 & (i < j) \\ l_{ij} & (i \geq j, l_{ii} \neq 0) \end{cases} \quad (7.3)$$

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix} = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ & \ddots & \vdots \\ & & u_{nn} \end{bmatrix} = [u_{ij}], u_{ij} = \begin{cases} u_{ij} & (i \leq j, u_{ii} \neq 0) \\ 0 & (i > j) \end{cases} \quad (7.4)$$

という形の行列である。

もし, 行列 A が L と U の積, つまり

$$A = LU$$

と分解できたとする。これを A の LU 分解と呼ぶことにする。さすれば (7.1) は

$$(LU)\mathbf{x} = \mathbf{b} \quad (7.5)$$

となる。これを 2 段階に分けて計算する。まず,

$$L\mathbf{y} = \mathbf{b} \Leftrightarrow \begin{bmatrix} l_{11} & & \\ \vdots & \ddots & \\ l_{n1} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad (7.6)$$

を y_1 から y_n の順に $\mathbf{y} (= U\mathbf{x})$ について解き, 次にこの \mathbf{y} を定数ベクトルとする連立一次方程式

$$U\mathbf{x} = \mathbf{y} \Leftrightarrow \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ & \ddots & \vdots \\ & & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad (7.7)$$

を x_n から x_1 の順に, \mathbf{x} について解く。(7.6) を解く過程を前進代入 (forward substitution), (7.7) を解く過程を後退代入 (backward substitution) と呼ぶ。よって, もし A が LU 分解できればたやすく (7.1) は解けることになる。

ではどのようにして LU 分解を行うのか。これは A を行列の基本変形によって上三角行列にする (上三角化), Gauss の消去法の考え方をを使うことで得ることが出来る。簡単のために, $A \in M_3(\mathbb{R})$, $\mathbf{x}, \mathbf{b} \in \mathbb{R}^3$ の場合に限ってこの方法を説明する。この時, (7.1) を書き下すと

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \Leftrightarrow \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

となる。この方程式を、解 \mathbf{x} が変化しないように、最終的には

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ & a_{22}^{(1)} & a_{23}^{(1)} \\ & & a_{33}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \end{bmatrix} \Leftrightarrow \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ & a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 = b_2^{(1)} \\ & & a_{33}^{(2)}x_3 = b_3^{(2)} \end{cases}$$

という形に変形する方法が、Gauss の消去法であり、これによって行列 A は上三角化される。

まず、第1列目を a_{11} で割り

$$\begin{bmatrix} 1 & \frac{1}{a_{11}} \cdot a_{12} & \frac{1}{a_{11}} \cdot a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{a_{11}} \cdot b_1 \\ b_2 \\ b_3 \end{bmatrix} \\ \Leftrightarrow \begin{cases} x_1 + \frac{1}{a_{11}} \cdot a_{12}x_2 + \frac{1}{a_{11}} \cdot a_{13}x_3 = \frac{1}{a_{11}} \cdot b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

と「変形したと仮定する」。この第1列目に a_{21} を乗じて第2列目から引けば

$$\begin{bmatrix} 1 & \frac{1}{a_{11}}a_{12} & \frac{1}{a_{11}}a_{13} \\ a_{21} - a_{21} \cdot 1 & a_{22} - \frac{a_{21}}{a_{11}} \cdot a_{12} & a_{23} - \frac{a_{21}}{a_{11}} \cdot a_{13} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{a_{11}} \cdot b_1 \\ b_2 - \frac{a_{21}}{a_{11}} \cdot b_1 \\ b_3 \end{bmatrix} \\ \Leftrightarrow \begin{cases} x_1 + \frac{a_{12}}{a_{11}}x_2 + \frac{a_{13}}{a_{11}}x_3 = \frac{b_1}{a_{11}} \\ (a_{21} - a_{21} \cdot 1)x_1 + \left(a_{22} - \frac{a_{21}}{a_{11}} \cdot a_{12}\right)x_2 + \left(a_{23} - \frac{a_{21}}{a_{11}} \cdot a_{13}\right)x_3 = b_2 - \frac{a_{21}}{a_{11}} \cdot b_1 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

となる。同様にして、第 1 列目に a_{31} を乗じて第 3 列目から引くと、

$$\begin{aligned} & \begin{bmatrix} 1 & \frac{1}{a_{11}}a_{12} & \frac{1}{a_{11}}a_{13} \\ a_{21} - a_{21} \cdot 1 & a_{22} - \frac{a_{21}}{a_{11}} \cdot a_{12} & a_{23} - \frac{a_{21}}{a_{11}} \cdot a_{13} \\ a_{31} - a_{31} \cdot 1 & a_{32} - \frac{a_{31}}{a_{11}} \cdot a_{12} & a_{33} - \frac{a_{31}}{a_{11}} \cdot a_{13} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{a_{11}} \cdot b_1 \\ b_2 - \frac{a_{21}}{a_{11}} \cdot b_1 \\ b_3 - \frac{a_{31}}{a_{11}} \cdot b_1 \end{bmatrix} \\ \Leftrightarrow & \begin{cases} x_1 + \frac{a_{12}}{a_{11}}x_2 + \frac{a_{13}}{a_{11}}x_3 = \frac{b_1}{a_{11}} \\ (a_{21} - a_{21} \cdot 1)x_1 + \left(a_{22} - \frac{a_{21}}{a_{11}} \cdot a_{12}\right)x_2 + \left(a_{23} - \frac{a_{21}}{a_{11}} \cdot a_{13}\right)x_3 = b_2 - \frac{a_{21}}{a_{11}} \cdot b_1 \\ (a_{31} - a_{31} \cdot 1)x_1 + \left(a_{32} - \frac{a_{31}}{a_{11}} \cdot a_{12}\right)x_2 + \left(a_{33} - \frac{a_{31}}{a_{11}} \cdot a_{13}\right)x_3 = b_3 - \frac{a_{31}}{a_{11}} \cdot b_1 \end{cases} \end{aligned}$$

となる。煩雑なのでこれをまとめて

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{22}^{(1)} & a_{23}^{(1)} \\ a_{32}^{(1)} & a_{33}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(1)} \end{bmatrix} \Leftrightarrow \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 = b_2^{(1)} \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 = b_3^{(1)} \end{cases} \quad (7.8)$$

と書くことにする。第 1 列目は元に戻しても差し支えない。

同様にして、今度は第 2 列目を $a_{22}^{(1)}$ で割り、

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 1 & \frac{1}{a_{22}^{(1)}} \cdot a_{23}^{(1)} \\ a_{32}^{(1)} & a_{33}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ \frac{1}{a_{22}^{(1)}} \cdot b_2^{(1)} \\ b_3^{(1)} \end{bmatrix} \Leftrightarrow \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ x_2 + \frac{1}{a_{22}^{(1)}} \cdot a_{23}^{(1)}x_3 = \frac{1}{a_{22}^{(1)}} \cdot b_2^{(1)} \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 = b_3^{(1)} \end{cases}$$

と「変形したと仮定する」。この第 2 列目に $a_{32}^{(1)}$ を乗じて第 3 列目から引けば

$$\begin{aligned} & \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ & 1 & \frac{1}{a_{22}^{(1)}} \cdot a_{23}^{(1)} \\ a_{32}^{(1)} - a_{32}^{(1)} \cdot 1 & a_{33}^{(1)} - \frac{a_{32}^{(1)}}{a_{22}^{(1)}} \cdot a_{23}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ \frac{1}{a_{22}^{(1)}} \cdot b_2^{(1)} \\ b_3^{(1)} - \frac{a_{32}^{(1)}}{a_{22}^{(1)}} \cdot b_2^{(1)} \end{bmatrix} \\ \Leftrightarrow & \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ x_2 + \frac{1}{a_{22}^{(1)}} \cdot a_{23}^{(1)}x_3 = \frac{1}{a_{22}^{(1)}} \cdot b_2^{(1)} \\ (a_{33}^{(1)} - \frac{a_{32}^{(1)}}{a_{22}^{(1)}} \cdot a_{23}^{(1)})x_3 = b_3^{(1)} - \frac{a_{32}^{(1)}}{a_{22}^{(1)}} \cdot b_2^{(1)} \end{cases} \end{aligned}$$

となる。煩雑なのでこれをまとめ、第2列目を元に戻して、目的の

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ & a_{22}^{(1)} & a_{23}^{(1)} \\ & & a_{33}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \end{bmatrix} \Leftrightarrow \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ & a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 = b_2^{(1)} \\ & & a_{33}^{(2)}x_3 = b_3^{(2)} \end{cases} \quad (7.9)$$

を得る。

これを実際にプログラム化することを考えると、係数があらかじめ0, 1になることが分かっている箇所の計算をサボることが出来る。また、どうせ元に戻る部分の変形も実際には行う必要はない。つまり、(7.8)の計算においては $a_{21}/a_{11}(=a_{21}^{(1)})$ と書くことにすると $a_{31}/a_{11}(=a_{31}^{(1)})$ を、(7.9)の計算においては $a_{32}^{(1)}/a_{22}^{(1)}(=a_{32}^{(2)})$ を用いるが、これをどうせ0になる成分に格納しておくことで余計な変数を確保する必要もなく、全ての計算が実行できるようになる。つまり、コンピュータのメモリ内においては、行列Aは

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(2)} & a_{33}^{(2)} \end{bmatrix}$$

と変化していけば良い。最後の行列が実はAのLU分解になっている。具体的には

$$L = \begin{bmatrix} 1 & & & \\ a_{21}^{(1)} & 1 & & \\ a_{31}^{(1)} & a_{32}^{(2)} & 1 & \end{bmatrix}, U = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ & a_{22}^{(1)} & a_{23}^{(1)} \\ & & a_{33}^{(2)} \end{bmatrix} \quad (7.10)$$

と格納されていると見ることが出来る。

以上をまとめると、LU分解(LU decomposition)法のアルゴリズムは次のようになる。

アルゴリズム 5 (LU 分解法)**(I) LU 分解**

1. $a_{ij}^{(0)} = a_{ij}$ とする。
2. $i = 1, 2, \dots, n-1$ に対し, 以下を計算する。
 - (a) $a_{ii}^{(i-1)} = 0$ ならば終了する。
 - (b) $j = i+1, \dots, n$ に対して以下の計算をする。
 - i. $a_{ji}^{(i)} := a_{ji}^{(i-1)} / a_{ii}^{(i-1)}$
 - ii. $k = i+1, \dots, n$ に対して,

$$a_{jk}^{(i)} := a_{jk}^{(i-1)} - a_{ji}^{(i)} \cdot a_{ik}^{(i-1)}$$

(II) 前進代入と後退代入

1. 以下を計算する。

$$\begin{aligned} y_1 &:= b_1 \\ y_2 &:= b_2 - a_{21}^{(1)} y_1 \\ &\vdots \\ y_{n-1} &:= b_{n-1} - \sum_{j=1}^{n-2} a_{n-1,j}^{(j)} y_j \\ y_n &:= b_n - \sum_{j=1}^{n-1} a_{nj}^{(j)} y_j \end{aligned}$$

2. 以下を計算する。

$$\begin{aligned} x_n &:= \frac{y_n}{a_{nn}^{(n-1)}} \\ x_{n-1} &:= \frac{y_{n-1} - a_{n-1,n}^{(n-2)} x_n}{a_{n-1,n-1}^{(n-2)}} \\ &\vdots \\ x_2 &:= \frac{y_2 - \sum_{j=3}^n a_{2j}^{(1)} x_j}{a_{22}^{(1)}} \\ x_1 &:= \frac{y_1 - \sum_{j=2}^n a_{1j}^{(0)} x_j}{a_{11}^{(0)}} \end{aligned}$$

このアルゴリズムを実行すると、任意の正則行列 $A = A^{(0)} \in M_n(\mathbb{R})$ は

$$A^{(n-1)} = \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(2)} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(2)} & a_{n3}^{(3)} & \cdots & a_{nn}^{(n-1)} \end{bmatrix}$$

となっている。このうち上三角部分 U と下三角部分 L を

$$L = \begin{bmatrix} 1 & & & & \\ a_{21}^{(1)} & \ddots & & & \\ \vdots & \ddots & \ddots & & \\ a_{n1}^{(1)} & \cdots & a_{n,n-1}^{(n-1)} & 1 & \end{bmatrix}, \quad U = \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{1n}^{(0)} \\ & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n-1)} \end{bmatrix}$$

とすれば

$$LU = A^{(0)}$$

という関係がある。

例題 7.2.1

3次正方行列 A を

$$A = \begin{bmatrix} 3 & -1 & 2 \\ 5 & 1 & -2 \\ 2 & 1 & -1 \end{bmatrix}$$

とする。この時、 A をLU分解すると

$$L = \begin{bmatrix} 1 & & \\ 5/3 & 1 & \\ 2/3 & 5/8 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 3 & -1 & 2 \\ & 8/3 & -16/3 \\ & & 1 \end{bmatrix}$$

となる。

LU分解のアルゴリズムを次のように変更したものをLDU分解 (LDU decomposition) という。

アルゴリズム 6 (LDU 分解)

1. $a_{ij}^{(0)} = a_{ij}$ とする。
2. $i = 1, 2, \dots, n-1$ に対して以下の計算をする。
 - (a) $a_{ii}^{(i-1)} = 0$ ならば終了する。
 - (b) $j = i+1, \dots, n$ に対して以下を計算する。

$$a_{ij}^{(i-1)} := \frac{a_{ij}^{(i-1)}}{a_{ii}^{(i-1)}}$$

- (c) $j = i+1, \dots, n$ に対して以下を計算する。
 - i. $k = i+1, \dots, n$ に対して,

$$a_{jk}^{(i)} := a_{jk}^{(i-1)} - a_{ji}^{(i-1)} \cdot a_{ik}^{(i-1)}$$

ii.

$$a_{ji}^{(i)} := \frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}}$$

このアルゴリズムで得た行列を L' (下三角行列), U' (上三角行列), D' (対角行列) で表わすと,

$$L' = \begin{bmatrix} 1 & & & & \\ a_{21}^{(1)} & \ddots & & & \\ \vdots & \ddots & \ddots & & \\ a_{n1}^{(1)} & \cdots & a_{n,n-1}^{(n-1)} & & 1 \end{bmatrix}, D' = \begin{bmatrix} a_{11}^{(0)} & & & & \\ & a_{22}^{(1)} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & a_{nn}^{(n-1)} \end{bmatrix}, U' = \begin{bmatrix} 1 & a_{12}^{(0)} & \cdots & a_{1n}^{(0)} \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,n}^{(n-2)} \\ & & & 1 \end{bmatrix}$$

となる。先と同様に

$$L'D'U' = A^{(0)} \quad (7.11)$$

という関係がある。更に LU 分解との関係では

$$L' = L, D'U' = U \quad (7.12)$$

となる。

例題 7.2.2

先の 3 次正方行列 A

$$A = \begin{bmatrix} 3 & -1 & 2 \\ 5 & 1 & -2 \\ 2 & 1 & -1 \end{bmatrix}$$

を LDU 分解すると,

$$L' = \begin{bmatrix} 1 & & \\ 5/3 & 1 & \\ 2/3 & 5/8 & 1 \end{bmatrix}, D' = \begin{bmatrix} 3 & & \\ & 8/3 & \\ & & 1 \end{bmatrix}, U' = \begin{bmatrix} 1 & -1/3 & 2/2 \\ & 1 & -2 \\ & & 1 \end{bmatrix}$$

となる。

LU分解やLDU分解を経て連立一次方程式を解くメリットは、定数項の計算を分離できることにある。つまり行列 A が同じで、定数項 \mathbf{b} だけが異なる問題を個々に解くとき、行列を一度だけLU(またはLDU)分解しておけば、計算しなければいけないのは前進代入、後退代入のみとなる。このようなシチュエーションは固有値問題(第11章)やNewton法(第12章)で出現する。

問題 7.2.1

係数行列のLDU分解が与えられている時、解 \mathbf{x} を求める手順を考えよ。

7.3 Gaussの消去法

線型代数の初歩で、必ず教えられるアルゴリズムであるが、本によっては掃き出し法もしくは Gauss-Jordan法との混同が見られる。ここでは以下に示すアルゴリズムを Gaussの消去法と呼ぶことにする。

アルゴリズム 7 (Gaussの消去法 (Gaussian Elimination))

ここでは便宜上

$$A^{(0)} := A = \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{1n}^{(0)} \\ a_{21}^{(0)} & a_{22}^{(0)} & \cdots & a_{2n}^{(0)} \\ \vdots & \vdots & & \vdots \\ a_{n1}^{(0)} & a_{n2}^{(0)} & \cdots & a_{nn}^{(0)} \end{bmatrix}, \mathbf{b}^{(0)} := \mathbf{b} = \begin{bmatrix} b_1^{(0)} \\ b_2^{(0)} \\ \vdots \\ b_n^{(0)} \end{bmatrix}$$

と、元の係数行列 A の各要素の上に添字 (0) を付けておくことにする。この添字は計算により値が更新されるごとに増えていくものとする。

I. 前進消去 (forward elimination)

1. $i = 1, 2, \dots, n-1$ に対し以下を計算する。

- (a) $a_{i-1}^{(i-1)} = 0$ ならば終了する。
- (b) $j = i+1, i+2, \dots, n$ に対して以下を計算する。
 - i. $k = i+1, i+2, \dots, n$ に対して

$$a_{jk}^{(i)} := a_{jk}^{(i-1)} - \frac{a_{ji}^{(i-1)} \cdot a_{ik}^{(i-1)}}{a_{ii}^{(i-1)}}$$

ii.

$$b_j^{(i)} := b_j^{(i-1)} - \frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}} \cdot b_i^{(i-1)}$$

II. 後退代入 (backward substitution)

1.

$$x_n := \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}} \circ$$

2. $i = n-1, \dots, 1$ に対して以下を計算する。

(a) $j = i+1, i+2, \dots, n$ に対して

$$b_i^{(i-1)} := b_i^{(i-1)} - a_{ij}^{(i-1)} \cdot x_j$$

(b)

$$x_i := \frac{b_i^{(i-1)}}{a_{ii}^{(i-1)}}$$

上記のような様式ではどのような操作を行っているか、イメージが湧きづらい。もう少し説明を加える。

まず前進消去で行列は

$$A^{(n-1)} = \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{1n}^{(0)} \\ & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n-1)} \end{bmatrix}$$

と上三角行列になる。従ってこの操作を**上三角化**ともいう。これを順に追ってみよう。まず、

$$A^{(0)} = \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{1n}^{(0)} \\ a_{21}^{(0)} & a_{22}^{(0)} & \cdots & a_{2n}^{(0)} \\ \vdots & \vdots & & \vdots \\ a_{n1}^{(0)} & a_{n2}^{(0)} & \cdots & a_{nn}^{(0)} \end{bmatrix}$$

に対して、

$$\begin{aligned} a_{21}^{(1)} &:= a_{21}^{(0)} - \frac{a_{21}^{(0)}}{a_{11}^{(0)}} a_{11}^{(0)} = 0 \\ a_{22}^{(1)} &:= a_{22}^{(0)} - \frac{a_{21}^{(0)}}{a_{11}^{(0)}} a_{12}^{(0)} \\ &\vdots \\ a_{2n}^{(1)} &:= a_{2n}^{(0)} - \frac{a_{21}^{(0)}}{a_{11}^{(0)}} a_{1n}^{(0)} \end{aligned}$$

と計算される。以下、 $j = 3, 4, \dots, n$ に対して

$$\begin{aligned} a_{j1}^{(1)} &:= a_{j1}^{(0)} - \frac{a_{j1}^{(0)}}{a_{11}^{(0)}} a_{11}^{(0)} = 0 \\ a_{j2}^{(1)} &:= a_{j2}^{(0)} - \frac{a_{j1}^{(0)}}{a_{11}^{(0)}} a_{12}^{(0)} \\ &\vdots \\ a_{jn}^{(1)} &:= a_{jn}^{(0)} - \frac{a_{j1}^{(0)}}{a_{11}^{(0)}} a_{1n}^{(0)} \end{aligned}$$

まで解くことができる。こうして解 \mathbf{x} の全成分を得ることができる。

ここで重要なことは2つある。

1. 前進消去の過程で $a_{ii}^{(i-1)} \neq 0$ を仮定したこと。実際にこのアルゴリズム通りにプログラムを組んでしまうと、 A が正則であるにもかかわらず解くことができなくなる可能性がある。例えば

$$A = \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix}$$

となる場合への対応をどうするか。

2. 丸め誤差を小さくするための工夫を行う必要はないか？[5]

このような事態を避けるため、枢軸選択 (pivoting strategy) という技法が考案されており、以下の2種類に分類されている。

アルゴリズム 8 (部分枢軸選択)

アルゴリズム 7 の前進消去において

「 $a_{i-1}^{(i-1)} = 0$ ならば終了する。」という部分を次のように改変する。

「 $w := \max_{i+1 \leq j \leq n} |a_{ij}^{(i-1)}|$ とするとき、 $w \neq |a_{ii}^{(i-1)}|$ ならば w のある行と第 i 行を入れ替える。 $w = 0$ ならば終了する。」

アルゴリズム 9 (完全枢軸選択)

アルゴリズム 7 の前進消去において

「 $a_{i-1}^{(i-1)} = 0$ ならば終了する。」という部分を次のように改変する。

「 $w := \max_{i+1 \leq j, k \leq n} |a_{jk}^{(i-1)}|$ とするとき、 $w \neq |a_{ii}^{(i-1)}|$ ならば w のある行と第 i 行を入れ替える。 $w = 0$ ならば終了する。」

簡単に言えば、部分枢軸選択は行方向だけ、完全枢軸選択は行、列方向に絶対値最大成分を探索する手法である。従って、前者の方が比較する要素の数が少ない分、素早く実行できる。

但し、誤差解析の条件としては完全枢軸選択の方が都合良い。以下計算される成分中、絶対値最大のものが必ず対角要素となるからである。Wilkinson[41] はこれを利用して丸め誤差の解析を行っている。

重要なことの2つ目は、消去法では行列要素の計算と定数ベクトルの計算が独立に行われていることである。つまり、前進消去過程で定数ベクトルの計算に使用される $a_{ji}^{(i-1)}/a_{ii}^{(i-1)}$ さえあれば、行列部分とは別個に計算できることになる。これを実現したのが前節の LU 分解である。

Gauss の消去法の変形として Gauss-Jordan 法がある。これは連立一次方程式の解法というよりは、逆行列の計算法として用いられる。

アルゴリズム 10 (Gauss-Jordan 法 (Sweeping-out Method))

1. $i = 1, 2, \dots, n$ に対して以下を計算する。

(a) $a_{ii}^{(i-1)} = 0$ ならば終了する。

(b) $j = 1, 2, \dots, i-1, i+1, \dots, n$ に対して以下を計算する。

i. $k = i+1, i+2, \dots, n$ に対して

$$a_{jk}^{(i)} := a_{jk}^{(i-1)} - \frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}} a_{ik}^{(i-1)}$$

ii. $b_j^{(i)} := b_j^{(i-1)} - \frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}} b_i^{(i-1)}$

2. $i = 1, 2, \dots, n$ に対して

$$x_i := \frac{b_i^{(n)}}{a_{ii}^{(i-1)}}$$

まず1. で行列を

$$A^{(n)} = \begin{bmatrix} a_{11}^{(0)} & & & \\ & a_{22}^{(1)} & & \\ & & \ddots & \\ & & & a_{nn}^{(n-1)} \end{bmatrix}$$

という形にする。つまり対角成分以外を全て0クリアしてしまうのである。このときはもちろん定数項も適宜計算され、 $\mathbf{b}^{(n)} = [b_1^{(n)} \ b_2^{(n)} \ \dots \ b_n^{(n)}]^T$ となっている。

あとは

$$x_i := \frac{b_i^{(n)}}{a_{ii}^{(i-1)}} \quad (i = 1, 2, \dots, n)$$

として順に \mathbf{x} の全成分を得ることができる。

Gauss の消去法, Gauss-Jordan 法は計算量の面でかなりの相違がある事を、日本では二宮 [29] がかなり以前から繰り返し繰り返し指摘している。以下の補題でそれを確認しよう。

補題 7.3.1

Gauss の消去法の計算量は

乗除算 : $\frac{n}{6}(n-1)(2n+5) + \frac{n}{2}(n+1)$ 回

加減算 : $\frac{n}{3}(n^2-1) + \frac{n}{2}(n-1)$ 回

である。

(証)

地道に計算する以外にない。

(1) 前進消去

$$a_{jk}^{(i)} := a_{jk}^{(i-1)} - \frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}} a_{ik}^{(i-1)}, \quad b_j^{(i)} := b_j^{(i-1)} - \frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}} b_i^{(i-1)}$$

下線部分は中間変数 w に入れておくと、

$$a_{jk}^{(i)} := a_{jk}^{(i-1)} - w \cdot a_{ik}^{(i-1)}, \quad b_j^{(i)} := b_j^{(i-1)} - w \cdot b_i^{(i-1)}$$

となり、乗除算1回、加減算1回である。これが $k = i+1, i+2, \dots, n$ まで続くから

乗除算 : $n - (i + 1) + 1 = n - i + 1$ 回

加減算 : $n - (i + 1) + 1 = n - i + 1$ 回

である。これに j が変わるたびに計算される中間変数 w の分を加えて、乗除算は $n - i + 2$ となる。

これが $j = i + 1, i + 2, \dots, n$ まで続くから

乗除算 : $(n - i + 2)(n - i)$ 回

加減算 : $(n - i + 1)(n - i)$ 回。

よって合計で

乗除算 : $\sum_{i=1}^{n-1} (n - i + 2)(n - i) = \frac{n}{6}(n - 1)(2n + 5)$ 回

加減算 : $\sum_{i=1}^{n-1} (n - i + 1)(n - i) = \frac{n}{3}(n^2 - 1)$ 回

となる。

(2) 後退代入

$$x_i := \frac{b_i^{(i-1)} - \sum_{j=i+1}^n a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}}$$

より

乗除算 : $n - i + 1$ 回

加減算 : $n - i$ 回。

よって

乗除算 : $\sum_{i=1}^n (n - i + 1) = \sum_{i=1}^n i = \frac{n}{2}(n + 1)$ 回

加減算 : $\sum_{i=1}^n (n - i) = \sum_{i=1}^{n-1} i = \frac{n}{2}(n - 1)$ 回

を得る。

以上を合計して、計算量を得た。

(証明終)

では Gauss-Jordan 法はどうか。

補題 7.3.2

Gauss-Jordan 法の計算量は

乗除算 : $\frac{n(n^2 + 2n - 1)}{2}$ 回

加減算 : $\frac{n(n^2 - 1)}{2}$ 回

となる。

(証)

乗除算 消去法と同様、中間変数 $w := a_{ji}^{(i-1)} / a_{ii}^{(i-1)}$ を使用すると一行につき $n - i + 1$ 回の乗除算が必要になる。これに定数項の分を加えて $n - i + 2$ 回となる。

加減算 定数項分を合わせて $n - i + 1$ 回。

よって合計、

乗除算 : $(n - 1) \sum_{i=1}^n (n - i + 2) + n = \frac{n(n^2 + 2n - 1)}{2}$ 回

加減算 : $(n - 1) \sum_{i=1}^n (n - i + 1) = \frac{n(n^2 - 1)}{2}$ 回

となる。

(証明終)

上の補題から、消去法の方が Gauss-Jordan 法よりも約 1.5 倍近く計算量が少ないことがわかる。前に Gauss-Jordan 法は逆行列の計算で用いられることが多いことを述べた。ここでそのアルゴリズムをみることにする。

アルゴリズム 11 (逆行列計算用の Gauss-Jordan 法)

1. $i = 1, 2, \dots, n$ に対して以下の計算を行う。

(a) $a_{ii}^{(i-1)} := 1/a_{ii}^{(i-1)}$

(b) $j = 1, 2, \dots, i-1, i+1, \dots, n$ に対して

$$a_{ji}^{(i-1)} := a_{ji}^{(i-1)} \cdot a_{ii}^{(i-1)}$$

(c) $j = 1, 2, \dots, i-1, i+1, \dots, n$ に対して以下の計算を行う。

i. $k = 1, 2, \dots, i-1, i+1, \dots, n$ に対して

$$a_{jk}^{(k-1)} := a_{jk}^{(k-2)} - a_{ji}^{(i-1)} \cdot a_{ik}^{(k-2)}$$

(d) $j = 1, 2, \dots, i-1, i+1, \dots, n$ に対して

$$a_{ji}^{(i-1)} := -a_{ji}^{(i-1)} \cdot a_{ii}^{(i-1)}$$

例題 7.3.3

$n = 4$ として具体的に検証してみよう。

普通は

$$\left[\begin{array}{cccc|cccc} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & a_{14}^{(0)} & 1 & 0 & 0 & 0 \\ a_{21}^{(0)} & a_{22}^{(0)} & a_{23}^{(0)} & a_{24}^{(0)} & 0 & 1 & 0 & 0 \\ a_{31}^{(0)} & a_{32}^{(0)} & a_{33}^{(0)} & a_{34}^{(0)} & 0 & 0 & 1 & 0 \\ a_{41}^{(0)} & a_{42}^{(0)} & a_{43}^{(0)} & a_{44}^{(0)} & 0 & 0 & 0 & 1 \end{array} \right] = [A^{(0)}|B^{(0)}]$$

というように単位行列を並べて書いておく。

まず第一列を $a_{11}^{(0)}$ で割り

$$\left[\begin{array}{cccc|cccc} 1 & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} & 1/a_{11}^{(0)} & 0 & 0 & 0 \\ a_{21}^{(0)} & a_{22}^{(0)} & a_{23}^{(0)} & a_{24}^{(0)} & 0 & 1 & 0 & 0 \\ a_{31}^{(0)} & a_{32}^{(0)} & a_{33}^{(0)} & a_{34}^{(0)} & 0 & 0 & 1 & 0 \\ a_{41}^{(0)} & a_{42}^{(0)} & a_{43}^{(0)} & a_{44}^{(0)} & 0 & 0 & 0 & 1 \end{array} \right]$$

という形にする。ここで $a_{1j}^{(1)} = a_{1j}^{(0)}/a_{11}^{(0)}$ である。

右の行列で変化したのは (1, 1) 成分だけだから、

$$\left[\begin{array}{cccc} 1/a_{11}^{(0)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ a_{21}^{(0)} & a_{22}^{(0)} & a_{23}^{(0)} & a_{24}^{(0)} \\ a_{31}^{(0)} & a_{32}^{(0)} & a_{33}^{(0)} & a_{34}^{(0)} \\ a_{41}^{(0)} & a_{42}^{(0)} & a_{43}^{(0)} & a_{44}^{(0)} \end{array} \right]$$

とできる。

次に $a_{ij}^{(1)} := a_{ij}^{(0)} - a_{1j}^{(1)} \cdot a_{i1}^{(0)}$ とすることで

$$\left[\begin{array}{cccc|cccc} 1 & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} & 1/a_{11}^{(0)} & 0 & 0 & 0 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & -a_{21}^{(0)}/a_{11}^{(0)} & 1 & 0 & 0 \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & a_{34}^{(1)} & -a_{31}^{(0)}/a_{11}^{(0)} & 0 & 1 & 0 \\ 0 & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} & -a_{41}^{(0)}/a_{11}^{(0)} & 0 & 0 & 1 \end{array} \right]$$

となる。右側の行列で変化している成分は 1 列目だけだから

$$\left[\begin{array}{cccc} 1/a_{11}^{(0)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ -a_{21}^{(0)}/a_{11}^{(0)} & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} \\ -a_{31}^{(0)}/a_{11}^{(0)} & a_{32}^{(1)} & a_{33}^{(1)} & a_{34}^{(1)} \\ -a_{41}^{(0)}/a_{11}^{(0)} & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} \end{array} \right]$$

としてよい。こうして左側の枢軸列に対応する右側の列を入れておけば、右側の行列は必要ない。

こうして結局

$$\left[\begin{array}{cccc} 1/a_{11}^{(0)} & -a_{12}^{(1)}/a_{22}^{(1)} & -a_{13}^{(2)}/a_{33}^{(2)} & -a_{14}^{(3)}/a_{44}^{(3)} \\ -a_{21}^{(0)}/a_{11}^{(0)} & 1/a_{22}^{(1)} & -a_{23}^{(2)}/a_{33}^{(2)} & -a_{24}^{(3)}/a_{44}^{(3)} \\ -a_{31}^{(0)}/a_{11}^{(0)} & -a_{32}^{(1)}/a_{22}^{(1)} & 1/a_{33}^{(2)} & -a_{34}^{(3)}/a_{44}^{(3)} \\ -a_{41}^{(0)}/a_{11}^{(0)} & -a_{42}^{(1)}/a_{22}^{(1)} & -a_{43}^{(2)}/a_{33}^{(2)} & 1/a_{44}^{(3)} \end{array} \right] = A^{-1}$$

を得る。

このアルゴリズムを用いて逆行列を求めるには

$$\begin{cases} \text{乗除算} & : n^3 \text{回} \\ \text{加減算} & : n(n-1)^2 \text{回} \end{cases}$$

の計算を必要とする。よって $\mathbf{x} := A^{-1}\mathbf{b}$ として連立一次方程式を解けば、合計

$$\begin{cases} \text{乗除算} & : n^2(n+1) \text{回} \\ \text{加減算} & : n^2(n-1) \text{回} \end{cases}$$

となる。これは消去法の約 3 倍、Gauss-Jordan 法の倍近い計算量を要する。

7.4 Crout 法, 修正 Cholesky 分解

Crout 法は LU 分解そのものといって良く、計算の順番が異なるだけである。アルゴリズムは以下の通りである。

アルゴリズム 12 (Crout 法)

1. $i = 1, 2, \dots, n$ に対して以下を計算する。

$$(a) \ t_{i1} := a_{i1}^{(0)}$$

(b) $j = 2, 3, \dots, n$ に対して,

$$t_{ij} := a_{ij}^{(0)} - \sum_{l=1}^{i-1} t_{il} \cdot u_{lj}$$

(c) $j = i + 1, i + 2, \dots, n$ に対して,

$$u_{ij} := \frac{a_{ij}^{(0)} - \sum_{l=1}^{i-1} t_{il} \cdot u_{lj}}{t_{ii}}$$

(d) $k_i := (b^{(0)} - \sum_{l=1}^{i-1} t_{il} \cdot b_l^{(0)})/t_{ii}$

2. $x_n := k_n$

3. $i = n - 1, \dots, 1$ に対して

$$x_i := k_i - \sum_{j=i+1}^n u_{ij} \cdot x_j$$

LU 分解との相違点は、計算していく順序だけである。LU 分解は

$$\boxed{\text{LU 分解}} \rightarrow a_{ij}^{(k)} := a_{ij}^{(k-1)} - \sum_{l=k+1}^n a_{il}^{(k-1)} \cdot a_{lj}^{(k-1)} / a_{kk}^{(k-1)} \quad (j = k + 1, \dots, n)$$

と、行方向に計算していたのに対し、Crout 法は

$$\boxed{\text{Crout 法}} \rightarrow \begin{cases} t_{ij} := a_{ij}^{(0)} - \sum_{l=1}^{i-1} t_{il} \cdot u_{lj} & (j = 2, \dots, n) \\ u_{ij} := (a_{ij}^{(0)} - \sum_{l=1}^{i-1} t_{il} \cdot u_{lj}) / t_{ii} & (j = i + 1, \dots, n) \end{cases}$$

とループを二段構えにする。

以上見てきた Gauss の消去法、LU 分解、LDU 分解、Crout 法は正則な係数行列を持つ全ての連立一次方程式を解くことのできる汎用解法である。しかし、係数行列の性質を利用することで、更に計算量を減らすことができる。

特に実用上重要なのは、 A が実対称行列の場合である。これは偏微分方程式の離散化解法で頻出する。このとき次の補題が成立する。

補題 7.4.1

$A \in M_n(\mathbb{R})$ が $A^T = A$ であるとき、 A を枢軸選択なしで LDU 分解すると

$$U = L^T$$

となる。

(証)

LDU 分解は

$$a_{jk}^{(i)} := a_{jk}^{(i-1)} - \frac{a_{ik}^{(i-1)} \cdot a_{kj}^{(i-1)}}{a_{kk}^{(i-1)}}$$

と計算されるから、もし $i - 1$ 段目の消去が終わった段階で $a_{pq}^{(i-1)} = a_{qp}^{(i-1)}$ であれば当然 $a_{pq}^{(i)} = a_{qp}^{(i)}$ となる。よって $a_{pq}^{(0)} = a_{qp}^{(0)}$ から

$$U = L^T$$

が成立する。

(証明終)

この補題から, A の上三角 (もしくは下三角) 部分は計算しなくとも済むことがわかる。この実対称行列に対する LDU 分解を修正 Cholesky 分解と呼ぶ。

例題 7.4.2

実対称行列 A を

$$A = \begin{bmatrix} 2 & 1 & -1 \\ 1 & 3 & 1 \\ -1 & 1 & 4 \end{bmatrix}$$

とする。この A に対する修正 Cholesky 分解は

$$A = \begin{bmatrix} 1 & & \\ 1/2 & 1 & \\ -1/2 & 3/5 & 1 \end{bmatrix} \begin{bmatrix} 2 & & \\ & 5/2 & \\ & & 13/5 \end{bmatrix} \begin{bmatrix} 1 & 1/2 & -1/2 \\ & 1 & 3/5 \\ & & 1 \end{bmatrix}$$

となる。

問題 7.4.1

1. 例題 7.4.2 の A を係数行列として持つ連立一次方程式

$$A\mathbf{x} = [2 \ 5 \ 4]^T$$

の解を求めよ。また係数行列の修正 Cholesky 分解が与えられた時, 解を求める手順を考えよ。

2. 修正 Cholesky 分解のアルゴリズムを書け。

演習問題

1. 次の連立一次方程式 $A\mathbf{x} = \mathbf{b}$ について, 以下の問いに答えよ。

$$\begin{bmatrix} 3 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ -2 \\ 19 \end{bmatrix}$$

- (a) 係数行列 A を LU 分解すると,

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \boxed{(1)} & 1 & 0 & \boxed{(2)} \\ 0 & -\frac{3}{8} & \boxed{(3)} & 0 \\ \boxed{(4)} & 0 & \boxed{(5)} & 1 \end{bmatrix}$$

