

# 連立1次方程式の解法

山本昌志\*

2004年11月9日

## 1 連立方程式

### 1.1 表現方法

言うまでも無く連立1次方程式 (Linear Equations) は、次のような形をしている。

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1N}x_N &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2N}x_N &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3N}x_N &= b_3 \\ &\vdots \\ a_{M1}x_1 + a_{M2}x_2 + a_{M3}x_3 + \cdots + a_{MN}x_N &= b_M \end{aligned} \quad (1)$$

ここでは、 $M = N$  の場合を考える。 $M \neq N$  のようなものは、ここでの講義のレベルを超えるので、興味がある人は自分で学習すること。このような連立1次方程式を計算することは、実際に工学の問題でしばしば現れる。例えば、偏微分方程式を離散化して解く場合などである。その場合、方程式の次元数がかなり大きく、100万円くらいは普通である。100万といっても、3次元問題だと、 $100 \times 100 \times 100$  程度であるので、まだまだ精度は荒い。

式 (1) は行列とベクトルで書くと、式がすっきりして考えやすくなる。書き直すと、

$$Ax = b \quad (2)$$

となるのは、以前学習したとおりである。それぞれの行列とベクトルは、

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1N} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2N} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3N} \\ \vdots & & & \ddots & \vdots \\ a_{N1} & a_{N2} & a_{N3} & \cdots & a_{NN} \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_N \end{pmatrix} \quad (3)$$

---

\*国立秋田工業高等専門学校 電気工学科

を表す。行列は大文字の太文字 (bold) スタイル、ベクトルは小文字の太文字スタイル、それぞれの成分は標準スタイルで表されることが多い。

ここで、解く問題は行列  $A$  が  $N \times N$  の正方行列で、その行列式がゼロでないものとする。要するに、普通に解ける連立方程式である。ここで、解くべき問題は、既知の  $A$  と  $b$  から、行列方程式 (2) を満たす  $x$  を求めることになる。この行列方程式解く過程で、 $A$  の逆行列や行列式の値を求めることができる。逆行列や行列式は連立方程式と密接にかかわっているのである。

通常、連立 1 次方程式 (1) は

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1N} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2N} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3N} \\ \vdots & & & \ddots & \vdots \\ a_{N1} & a_{N2} & a_{N3} & \cdots & a_{NN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_N \end{pmatrix} \quad (4)$$

と書き表せる。このようにすると、見通しがかなり良くなる。皆さんも、今後連立方程式を書くときは、行列とベクトルで書く方が良いであろう。ちょっとばかり格好良い。

行列やベクトルを使うと、格好良いばかりでなくコンピューターで扱いやすくなる。例えば、行列  $A$  の要素  $a_{ij}$  はプログラム中では 2 次元配列 `a[i][j]` として扱える。同様にベクトル  $b_k$  は 1 次元配列 `b[k]` として扱える。

## 1.2 計算方法

連立 1 次方程式は、クラメールの公式により、解のベクトル  $x$  は四則演算で計算できる。行列  $A$  が正則 ( $\det A \neq 0$ ) ならば、解は

$$x_j = \frac{1}{\det A} \begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1j-1} & b_{1j} & a_{1j+1} & \cdots & a_{1N} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2j-1} & b_{2j} & a_{2j+1} & \cdots & a_{2N} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3j-1} & b_{3j} & a_{3j+1} & \cdots & a_{3N} \\ \vdots & & & \ddots & & & & \ddots & \vdots \\ a_{N1} & a_{N2} & a_{N3} & \cdots & a_{Nj-1} & b_{Nj} & a_{Nj+1} & \cdots & a_{NN} \end{vmatrix} \quad (5)$$

である。これは、2 つの行列式を計算する必要があり、計算量が大変多くなる。したがって、 $N \geq 4$  の場合は推奨できない。

次に考えられるのは、 $A$  の逆行列  $A^{-1}$  を用いて、 $x = A^{-1}b$  から計算する方法である。この方法も、計算量と精度の面で問題がある。

連立 1 次方程式の計算方法は大きく分けて 2 通りある。1 つは、ここで学習する消去法で、他方は反復法と言われる方法である。どちらが良いかは、係数行列  $A$  の性質に依存する。一般に、 $A$  が密なとき、即ちほとんどの要素がゼロでないときは、消去法が有利である。一方、殆どの要素がゼロで、 $A$  が疎のとき、反復法が有利である。

ここでは消去法を学習するが、反復法について簡単に紹介しておく。まず、係数行列を  $A = B - C$  と変形します。すると、元の連立 1 次方程式は、 $Bx = Cx + b$  となる。これを解くために、漸化式  $Bx^{(k+1)} = Cx^{(k)} + b$

とする。もし、初期値  $x^{(0)}$  が良ければ、 $x^{(\infty)}$  は真の解  $x$  に収束する。もちろん、 $B$  は容易に計算できる連立 1 次方程式になるように選ぶ。この選び方により、ヤコビの反復法やガウス・ザイデル法、SOR 法などがある。

## 2 ガウス消去法と後退代入

ガウス消去法とガウス・ジョルダン法は単純で、諸君が今まで連立 1 次方程式を計算してきた方法と同じである。

ガウス消去法というのは、連立方程式 (4) を次のように変形させて、解く方法である。

$$\begin{pmatrix} a'_{11} & a'_{12} & a'_{13} & \cdots & a'_{1N} \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2N} \\ 0 & 0 & a'_{33} & \cdots & a'_{3N} \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a'_{NN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ b'_3 \\ \vdots \\ b'_N \end{pmatrix} \quad (6)$$

このように式を変形する方法をガウスの消去法と言う。実際の変形方法については、次のガウス・ジョルダン法とほとんど同じでなので、次節を参考にすること。このように式が変形できると後は簡単で、次にように  $x_N$  から  $x_1$  まで順次計算する。 $x$  の値は、

$$\begin{aligned} x_N &= \frac{1}{a'_{NN}} b'_N \\ x_{N-1} &= \frac{1}{a'_{N-1N-1}} (b'_{N-1} - a'_{N-1N} x_N) \\ x_{N-2} &= \frac{1}{a'_{N-2N-2}} (b'_{N-2} - a'_{N-2N-1} x_{N-1} - a'_{N-2N} x_N) \\ &\vdots \end{aligned} \quad (7)$$

と求めることができる。この式は、

$$x_i = \frac{1}{a'_{ii}} \left[ b'_i - \sum_{j=i+1}^N a'_{ij} x_j \right] \quad (8)$$

とまとめることができる。これを使って、 $N \sim 0$  まで処理することを後退代入と言う。重要なことは、後ろ  $N$  から処理することで、決して、1 から処理することはできない。ガウス消去法と後退代入により連立 1 次方程式は、コンピューターで容易に解くことができる。

## 3 ガウス・ジョルダン法

逆行列が不要であれば、ガウス・ジョルダン法よりも、後で述べる LU 分解の法が計算速度は速い。しかし、教育的効果を考えると、両方の方法を知っておくのは良いことです。

### 3.1 基本的な考え方

ガウス・ジョルダン (Gauss-Jordan) 法というのは、連立方程式 (4) を次のように変形させて、解く方法である。

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ b'_3 \\ \vdots \\ b'_N \end{pmatrix} \quad (9)$$

この式から明らかに、求める解  $x_i = b'_i$  となる。これをどうやって求めるか?。コンピューターで実際に計算する前に、人力でガウス・ジョルダン法で計算してみる。例として、

$$\begin{cases} 3x + 2y + z = 10 \\ x + y + z = 6 \\ x + 2y + 2z = 11 \end{cases} \quad (10)$$

をガウス・ジョルダン法で解を求める。

まずは、1行目の  $x$  の係数を 1 に、2 と 3 行目のそれは 0 にします。そのために、1 行目は  $x$  の係数の値で割る。2 行目と 3 行目は、1 行目に適当な係数を掛けて引く。次のようにする。

$$\begin{cases} x + \frac{2}{3}y + \frac{1}{3}z = \frac{10}{3} \\ x + y + z = 6 \\ x + 2y + 2z = 11 \end{cases} \Rightarrow \begin{cases} x + \frac{2}{3}y + \frac{1}{3}z = \frac{10}{3} \\ 0 + \frac{1}{3}y + \frac{2}{3}z = \frac{8}{3} \\ 0 + \frac{4}{3}y + \frac{5}{3}z = \frac{23}{3} \end{cases} \quad (11)$$

つぎに、2 行目の  $y$  の係数を 1 にして、1 と 3 行目のそれを 0 にする。

$$\begin{cases} x + \frac{2}{3}y + \frac{1}{3}z = \frac{10}{3} \\ 0 + y + 2z = 8 \\ 0 + \frac{4}{3}y + \frac{5}{3}z = \frac{23}{3} \end{cases} \Rightarrow \begin{cases} x + 0 - z = -2 \\ 0 + y + 2z = 8 \\ 0 + 0 - z = -3 \end{cases} \quad (12)$$

同じことを  $z$  についても繰り返す。すると、

$$\begin{cases} x + 0 - z = -2 \\ 0 + y + 2z = 8 \\ 0 + 0 + z = 3 \end{cases} \Rightarrow \begin{cases} x + 0 + 0 = 1 \\ 0 + y + 0 = 2 \\ 0 + 0 + z = 3 \end{cases} \quad (13)$$

となる。従って、連立方程式 (10) の解は、

$$\begin{cases} x = 1 \\ y = 2 \\ z = 3 \end{cases} \quad (14)$$

となる。これがガウス・ジョルダン法である。もっともらしい名前が付けられているが、大したことはない。

これで、ガウス・ジョルダン法が理解できたと思う。もう少し数学的にその内容を説明する<sup>1</sup>。そのために、次の線形行列方程式を考える。ここでは、紙面の関係で係数行列が  $4 \times 4$  について述べるが、一般的に  $N$  への拡張は容易である。

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \left[ \begin{array}{c} \left( \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right) \\ \sqcup \\ \left( \begin{array}{cccc} y_{11} & y_{12} & y_{13} & y_{14} \\ y_{21} & y_{22} & y_{23} & y_{24} \\ y_{31} & y_{32} & y_{33} & y_{34} \\ y_{41} & y_{42} & y_{43} & y_{44} \end{array} \right) \end{array} \right] = \left[ \begin{array}{c} \left( \begin{array}{c} b_1 \\ b_2 \\ b_3 \\ b_4 \end{array} \right) \\ \sqcup \\ \left( \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right) \end{array} \right] \quad (15)$$

ここで、 $\sqcup$ は列拡大、つまりこの両側の括弧を取り去って行列をくっつけて幅を広くすること意味する。この式から、容易に

$$Ax = b \quad (16)$$

$$AY = E \quad (17)$$

が分かる。もちろん、 $E$ は単位行列である。要するに行列方程式(15)を解くということは、この2つの連立方程式を解くことに他ならない。 $x$ はもとの方程式(1)の解となり、 $Y$ は $A$ の逆行列である。

式(15)について、以下のことが容易に分かる。

- $A$ の任意の2行を入れ替えて、それに対応する $b$ と $E$ を入れ替えた場合、 $x$ や $Y$ の値と順序は変わらない。ただし、 $E$ はもはや単位行列ではない。これは連立方程式の順序を入れ替えて書いていることに相当している。
- $A$ の任意の行を、その行と別の行との線形結合に置き換え、同時に対応する $b$ と $E$ も同様に置き換える。この場合、 $x$ と $Y$ の値と順序は変わらない。当然、この場合も $E$ はもはや単位行列ではない。
- $A$ の任意の2列を入れ替えて、それに対応する $x$ と $Y$ の行を入れ替えれば、 $b$ と $E$ の順序は入れ替える必要は無い。この場合、解の行の順序が変わるので、最後に元に戻す操作が必要になる。

この3つの操作を組み合わせて、係数行列 $A$ を単位行列に変換するのがガウス・ジョルダン法である。 $A$ が単位行列に変換されれば、右辺に $x$ と $Y$ が表れる。したがって、解と逆行列が求められたことになる。もし、逆行列が不要であれば $x$ だけ計算し、逆行列のみ必要であれば $Y$ のみ計算する。

### 3.2 ピボット 選択

先に示した、ガウス・ジョルダン法の3つの基本操作のうち、2番目しか使わない方法を「ピボット選択なしのガウス・ジョルダン法」と言う。最初、人力で連立方程式を解いた方法である。この方法の明らかにまずい点は、もし1にしたい対角要素がゼロの場合、計算ができなくなってしまうところにある。この割る要素をピボット (pivot) と言う。ゼロでないにしても、そのピボットが非常に小さい値の場合、丸め誤差が大きくなり問題である<sup>2</sup>。このようなことから、普通はピボット選択なしのガウス・ジョルダン法というものは考えられない。

<sup>1</sup>この辺の議論は 技術評論社の「NUMERICAL RECIPES in C」を参考にしています。これは数値計算の良い教科書です。

<sup>2</sup>数値計算の場合、小さな数で割ることを非常に嫌います。誤差の入り込む要因です。式を変形して、できるだけそれを選ばなくてはなりません。

この問題を避けるためにどうするかというと、ピボット選択という方法を使う。方法は簡単で、先に示した3つの基本操作のうち、1番目と3番目を使って、対角に素性の良い要素をもってくる。1番目の操作のみを用いて行を入れ替える方法を、部分ピボット選択 (partial pivoting) と言う。1番目の操作と3番目の操作を使って、行と列を入れ替えるのを完全ピボット選択 (full pivoting) と言う。すでにある程度出来上がっている単位行列を壊したくないので、ピボットの選択は操作している行の下の行から選ばなくてはならない。

部分ピボット選択の方が明らかに簡単である。解の行列を入れ替える必要が無いからである。その場合、行の入れ替えしかしないので、ピボットはその列から選ばなくてはならない。完全ピボット選択の方が選べる要素が多いが、最終的な解の精度はあまり変わらないようである。したがって、ここではプログラムの簡単な部分ピボット選択で計算する事にする。

次に考えなくてはならないのは、ピボットを選択する基準である。簡単に言えば、大きな要素選択すれば大体よい。しかし、ある行を100万倍して、それに対応する右辺の行も100万倍することもできるので、ただ大きいというだけでは問題がありそうである。どうするかというと、各方程式の最大係数を1に規格化して、最大のものをピボットに選ぶことが行われている。この方法を陰的ピボット選択 (implicit pivoting) と呼ぶ。

これで、ピボットの問題も片付いたので、フローチャートを書いてみる。

### 3.3 フローチャート

ガウス・ジョルダン法のフローチャートを図1に示す。

## 4 LU分解

### 4.1 LU分解による解の計算方法

係数行列  $A$  が下三角行列  $L$  と上三角行列  $U$  の積に展開できたとする。

$$A = LU \tag{18}$$

下三角行列と上三角行列の要素を書き出すと

$$\begin{pmatrix} \alpha_{11} & 0 & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 0 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{pmatrix} \cdot \begin{pmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \\ 0 & \beta_{22} & \beta_{23} & \beta_{24} \\ 0 & 0 & \beta_{33} & \beta_{34} \\ 0 & 0 & 0 & \beta_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \tag{19}$$

となる。

このようにLU分解できると、連立1次方程式は

$$Ax = (LU)x = L(Ux) = b \tag{20}$$

と書ける。これをさらに書き換えると、

$$Ly = b \tag{21}$$

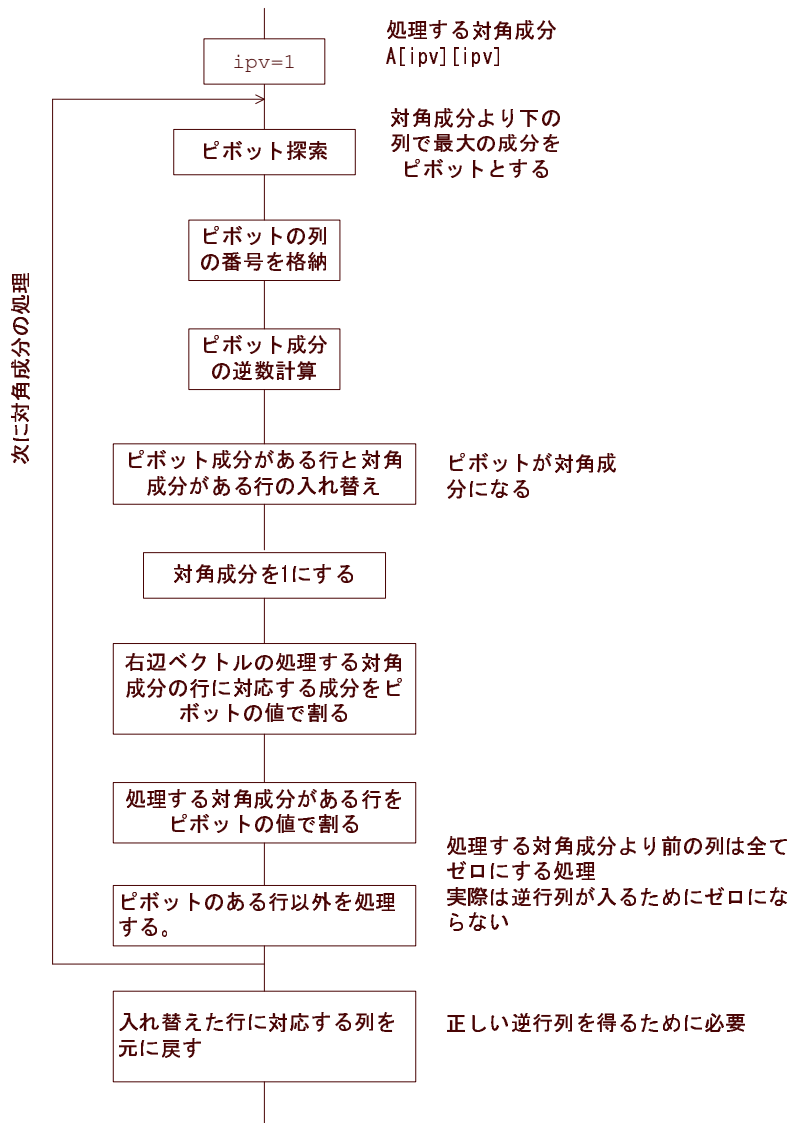


図 1: ガウス・ジョルダン法のフローチャート

$$Ux = y \quad (22)$$

となる。これらの連立方程式の解  $y$  と  $x$  は、それぞれの係数が三角行列なので容易に計算できる (ガウス消去法と後退代入の説明を見よ)。 $y$  の方は、係数が下三角行列なので  $1 \sim N$  まで前進代入により解ける。具体的には、

$$\begin{aligned} y_1 &= \frac{1}{\alpha_{11}} b_1 \\ y_i &= \frac{1}{\alpha_{ii}} \left( b_i - \sum_{j=1}^{i-1} \alpha_{ij} y_j \right) \quad i = 2, 3, \dots, N \end{aligned} \quad (23)$$

である。この  $y$  が求まったならば、今度は係数が上三角行列の式 (22) の  $x$  を求める。これは、 $N \sim 1$  の順序で計算する後退代入を使う。

$$\begin{aligned} x_N &= \frac{1}{\beta_{NN}} y_N \\ x_i &= \frac{1}{\beta_{ii}} \left( y_i - \sum_{j=1+1}^N \beta_{ij} x_j \right) \quad i = N-1, N-2, \dots, 1 \end{aligned} \quad (24)$$

これらの前進代入と後退代入は、コンピューターにとって非常に簡単に計算できる。これは、係数行列  $A$  を LU 分解できれば、連立方程式は簡単に解けると言っている。次節で LU 分解の方法を詳しく説明する。

いったん LU 分解が出来てしまえば、式 (1) の右辺  $b$  が変わっても、その LU 分解の形を変える必要がない。右辺が変わっても、LU 分解は 1 回で済む。これが、ガウスの消去法と後退代入を組み合わせた方法やガウス・ジョルダン法に比べて、際立って優れている点である。

## 4.2 LU 分解 (クラウトのアルゴリズム)

LU 分解するということは、式 (19) の  $\alpha_{ij}$  と  $\beta_{ij}$  を計算することにほかならない。この式の行列方程式は、 $N^2 + N$  の未知数と  $N^2$  の式を含む。未知数の数が方程式の数より多いので、 $N$  個の未知数を勝手に決めて残りを計算することが可能である。従って、LU 分解は一意に決まらない、計算しやすいように  $N$  個の未知数を決めることができる。ここでは、LU 分解のクラウト (Crout) のアルゴリズムに従い、

$$\alpha_{ii} = 1 \quad i = 1, 2, 3, \dots, N \quad (25)$$

とする。

それでは、クラウトのアルゴリズムによる LU 分解の手順を示すことにする。

1.  $\alpha_{ii} = 1$  ( $i = 1, \dots, N$ ) とします。この操作により、解くべき行列方程式 (19) は

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ \alpha_{21} & 1 & 0 & 0 \\ \alpha_{31} & \alpha_{32} & 1 & 0 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \end{pmatrix} \cdot \begin{pmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \\ 0 & \beta_{22} & \beta_{23} & \beta_{24} \\ 0 & 0 & \beta_{33} & \beta_{34} \\ 0 & 0 & 0 & \beta_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \quad (26)$$

と変形できる。



2. この式を見ると、 $\beta_{ij}$  と  $\alpha_{ij}$  が次に示す順序で簡単に求められることが分かる。まずは式を見て分かるように、 $\beta_{11}$  が直ちに計算できる。次に  $\beta_{11}$  を利用して、 $\alpha_{21}, \alpha_{31}, \alpha_{41}$  を求めることができる。これで、 $L$  と  $U$  の第 1 列目が求められた。次に第 2 列目である。これも  $\beta_{12}$  は直ちに計算できる。そうして、これまで分かっている  $\beta_{ij}$  と  $\alpha_{ij}$  を使うと、 $\beta_{22}, \alpha_{32}, \alpha_{42}$  を求めることができる。これで第 2 列目は終わり、同じことを繰り返すと、全ての  $\beta_{ij}$  と  $\alpha_{ij}$  が計算できる。これをアルゴリズムにすると次のようになる。

$j = 1, 2, 3, \dots, N$  という順序で計算する。 $L$  と  $U$  の  $j$  列目を計算することになる。具体的には、以下のようにして、 $j$  列目の  $\beta_{ij}$  と  $\alpha_{ij}$  を求める。

- まず、 $i = 1, 2, \dots, j$  について、次式に従い  $\beta_{ij}$  を計算する。

$$\begin{aligned}\beta_{1j} &= a_{1j} \\ \beta_{ij} &= a_{ij} - \sum_{k=1}^{i-1} \alpha_{ik} \beta_{kj}\end{aligned}\tag{27}$$

- 次に、 $i = j + 1, j + 2, \dots, N$  について、 $\alpha_{ij}$  を計算する。

$$\alpha_{ij} = \frac{1}{\beta_{jj}} \left( a_{ij} - \sum_{k=1}^{j-1} \alpha_{ik} \beta_{kj} \right)\tag{28}$$

- これで、 $L$  と  $U$  の  $j$  列目が完成したので、同じ操作を  $j + 1$  列目に行う。同じことを繰り返して、LU 分解の行列を完成させる。

この方法により、LU 分解ができる。次に示すピボット選択をしなければ、アルゴリズムは非常に単純である。

### 4.3 ピボット選択

ここでも、ピボット選択の問題が出てくる。式 (28) の  $\beta_{jj}$  で割る部分である。安定な解を求めるためには、ピボット選択は必要不可欠ということである。完全ピボット選択は複雑なので、部分ピボット選択で十分であろう。

ではどうするかですが、これも途中 ( $j$  列) まで分解した行列は崩したくない。そのためには、行列  $L$  の行を交換し、それに対応した行列  $A$  の行を交換すれば問題がもっとも少なくなる。当然、行列  $U$  は行も列も変化しない。最終的には行を交換した行列  $A'$  の LU 分解が出来る。連立 1 次方程式を解くときには、同様に  $b$  の行も交換しておく。ただし、行の交換であるため、解  $x$  の要素の順序は入れ替わらない。

つぎに、どのようにして交換する行を決めるかである。一般的には、 $\beta_{jj}$  が大きくなるように選択すれば良い結果が得られる。クラウト法のピボット選択は、次のように進める。 $j$  列目のピボットを選択する場合についてである。

1. まずは、 $j - 1$  列目までの行列  $L$  の各行の要素の最大値を 1 に規格化する。同時に、対応する行列  $A$  の行も同じ係数を掛ける。

2. そうして、

$$\gamma_i = a_{ij} - \sum_{k=1}^{i-1} \alpha_{ik} \beta_{kj} \quad (i = j, j+1, \dots, N) \quad (29)$$

を計算する。これは、式 (27) と同じ、式 (28) と  $\beta_{jj}$  で割ること以外は同じであることに注意が必要である。

3. 最大の  $\gamma_i$  となるものをピボットとして選択する。

4. 最大のピボットとなる行が分かったので、後は元 (規格化前) の  $L$  と  $A$  を用いて、式 (28) と  $\beta_{jj}$  を計算する。

これがピボット探索のルーチンである。実際には、ピボット作成用の関数を作成して、計算をすることになる。

## 5 練習問題

### 5.1 ガウス・ジョルダン法

1. 次の連立方程式をピボット選択無しのカウス・ジョルダン法で計算するプログラムを作成しなさい。これは、教科書の P.22 の例題 2 です。プログラムは、 $N = 3$  のみならず、 $N = 100$  程度まで容易に計算できるように汎用的にすること。

$$\begin{cases} 2x - 4y + 6z = 5 \\ -x + 7y - 8z = -3 \\ x + y - 2z = 2 \end{cases}$$

2. プログラムが完成したら、逆行列を計算するルーチンも追加しなさい。そして、逆行列と元の行列をかけ合わせたら単位行列になることを確認しなさい。

3. 逆行列が完成したら、ピボット選択のルーチンを追加しなさい。

4. ピボット選択のルーチンが完成したならば、次の連立方程式を計算しなさい。

$$a_{ij} = \cos \left[ \frac{\pi(i-1)(j-1)}{N} \right] \quad i, j = 1, 2, \dots, N$$
$$b_i = \frac{i-1}{N} \quad i = 1, 2, \dots, N$$

これは、三角波のフーリエ変換になっている。とりあえず、 $N=100$  程度で計算してみて、最後に

$$f(x) = \sum_{j=1}^N x_j \cos[(j-1)x]$$

をプロットして三角波になっていることを確認せよ。