Cプログラミング演習

第１３回　行列計算と線形方程式の求解

LU分解

- 連立一次方程式を解く手段。
- 通常、連立方程式を解く場合は、変数を減らしていくという方針で解いていくが、LU分解もその1つ。
- Gaussの消去法とLU分解の比較。
  - 計算機で実装した場合、Gaussの消去法はLU分解に比べて計算速度がかなり「遅い」。
  - 逆行列を求める方法も「遅い」。

連立一次方程式の行列表現

- 一般に連立一次方程式は、

$$A_{0,0}x_0+A_{0,1}x_1\cdots A_{0,n-1}x_{n-1} = b_0$$
$$A_{1,0}x_0+A_{1,1}x_1\cdots A_{1,n-1}x_{n-1} = b_1$$

$$A_{n-1,0}x_0+A_{n-1,1}x_1\cdots A_{n-1,n-1}x_{n-1} = b_{n-1} \qquad (1)$$

- これを行列表現すると、

$$\begin{pmatrix} A_{0,0} & A_{0,1} & \cdot\cdot & A_{0,n-1} \\ A_{1,0} & A_{1,1} & \cdot\cdot & A_{1,n-1} \\ : & : & : & : \\ A_{n-1,0} & A_{n-1,1} & \cdot\cdot & A_{n-1,n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ : \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ : \\ b_{n-1} \end{pmatrix}$$

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

上三角行列，下三角行列

- 三角行列であれば解を求めることが容易。
- LU分解では，行列Aを下三角行列Lと上三角行列Uの二つに分解する。

$$
A = \begin{pmatrix} A_{0,0} & A_{0,1} & \cdot\cdot & A_{0,n-1} \\ A_{1,0} & A_{1,1} & \cdot\cdot & A_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots \\ A_{n-1,0} & A_{n-1,1} & \cdot\cdot & A_{n-1,n-1} \end{pmatrix}
$$

$$
= \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ l_{1,0} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ l_{n-2,0} & l_{n-2,1} & \cdots & 1 & 0 \\ l_{n-1,0} & l_{n-1,1} & \cdots & l_{n-1,n-2} & 1 \end{pmatrix} \begin{pmatrix} u_{0,0} & u_{0,1} & \cdots & u_{0,n-2} & u_{0,n-1} \\ 0 & u_{1,1} & \cdots & u_{1,n-2} & u_{1,n-1} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & u_{n-2,n-2} & u_{n-2,n-1} \\ 0 & 0 & \cdots & 0 & u_{n-1,n-1} \end{pmatrix}
$$

$$
= LU \qquad (3)
$$

LU分解で連立一次方程式を解く(1)

- 一次方程式 $Ax = b$ を解く。

- 行列Aを下三角行列Lと上三角行列Uの二つにLU分解する。

  $$A = LU \tag{4}$$

- 行列AがLU分解されると求める方程式は以下のようになる。

  $$Ax = LUx = b \tag{5}$$

LU分解で連立一次方程式を解く(2)

- Ux=c とおいて、まず、

  $$Lc = b \tag{6}$$

  を解く

- c を求めたら、

  $$Ux = c \tag{7}$$

  を解くことで、xが求まる

LU分解する。

$$
\begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} \\ A_{1,0} & A_{1,1} & A_{1,2} \\ A_{2,0} & A_{2,1} & A_{2,2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{1,0} & 1 & 0 \\ l_{2,0} & l_{2,1} & 1 \end{pmatrix} \begin{pmatrix} u_{0,0} & u_{0,1} & u_{0,2} \\ 0 & u_{1,1} & u_{1,2} \\ 0 & 0 & u_{2,2} \end{pmatrix}
$$

$A_{00}=u_{00}$
$A_{01}=u_{01}$
$A_{02}=u_{02}$
$A_{10}=l_{10}*u_{00}$
$A_{11}=l_{10}*u_{01}+u_{11}$
$A_{12}=l_{10}*u_{02}+u_{12}$
$A_{20}=l_{20}*u_{00}$
$A_{21}=l_{20}*u_{01}+l_{21}*u_{11}$
$A_{22}=l_{20}*u_{02}+l_{21}*u_{12}+u_{22}$

$u_{00}=A_{00}$
$u_{01}=A_{01}$
$u_{02}=A_{02}$
$l_{10}=A_{10}/u_{00}$
$u_{11}=A_{11}-l_{10}*u_{01}$
$u_{12}=A_{12}-l_{10}*u_{02}$
$l_{20}=A_{20}/u_{00}$
$l_{21}=(A_{21}-l_{20}*u_{01})/u_{11}$
$u_{22}=A_{22}-l_{20}*u_{02}-l_{21}*u_{12}$

$u_{00}=A_{00}$
$u_{01}=A_{01}$
$u_{02}=A_{02}$
$l_{10}=A_{10}/u_{00}$
$l_{20}=A_{20}/u_{00}$
$u_{11}=A_{11}-l_{10}*u_{01}$
$u_{12}=A_{12}-l_{10}*u_{02}$
$l_{21}=(A_{21}-l_{20}*u_{01})/u_{11}$
$u_{22}=A_{22}-l_{20}*u_{02}-l_{21}*u_{12}$

一般化すると、

$$
\begin{pmatrix}
{}^{0}A_{0,0} & A_{0,1} & \cdots & A_{0,n-1} \\
{}^{1}A_{1,0} & {}^{2}A_{1,1} & \cdots & A_{1,n-1} \\
\vdots & {}^{3}\vdots & \vdots & \vdots \\
A_{n-1,0} & A_{n-1,1} & \cdots & A_{n-1,n-1}
\end{pmatrix}
$$

$$
\begin{pmatrix}
1 & 0 & \cdots & 0 & 0 \\
{}^{1}l_{1,0} & 1 & \cdots & 0 & 0 \\
\vdots & {}^{3}\vdots & & \vdots & \vdots \\
l_{n-2,0} & l_{n-2,1} & \cdots & 1 & 0 \\
l_{n-1,0} & l_{n-1,1} & \cdots & l_{n-1,n-2} & 1
\end{pmatrix}
\begin{pmatrix}
{}^{0}u_{0,0} & u_{0,1} & \cdots & u_{0,n-2} & u_{0,n-1} \\
0 & {}^{2}u_{1,1} & \cdots & u_{1,n-2} & u_{1,n-1} \\
\vdots & \vdots & & \vdots & \vdots \\
0 & 0 & \cdots & u_{n-2,n-2} & u_{n-2,n-1} \\
0 & 0 & \cdots & 0 & u_{n-1,n-1}
\end{pmatrix}
$$

$$
\begin{aligned}
&u_{0,k} = A_{0,k} && \text{where}(0 \leqq k \leqq n-1) \\
&l_{i,0} = A_{i,0}/u_{0,0} && \text{where}(1 \leqq i \leqq n-1) \\
&u_{1,k} = A_{1,k} - u_{0,k} \cdot l_{1,0} && \text{where}(1 \leqq k \leqq n-1) \qquad (8) \\
&\quad \vdots
\end{aligned}
$$

(6) の Lc = b

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ l_{1,0} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ l_{n-2,0} & l_{n-2,1} & \cdots & 1 & 0 \\ l_{n-1,0} & l_{n-1,1} & \cdots & l_{n-1,n-2} & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{pmatrix} \tag{9}$$

これを c について解くと、

$$\begin{aligned} c_0 &= b_0 \\ c_1 &= b_1 - l_{1,0} \cdot c_0 \\ c_2 &= b_2 - l_{2,0} \cdot c_0 - l_{2,1} \cdot c_1 \\ &\vdots \\ c_k &= b_k - \sum_{i=0}^{k-1} l_{k,i} \cdot c_i \qquad \text{where}(0 \leqq k \leqq n-1) \end{aligned} \tag{10}$$

最後に (7) の $Ux = c$

$$\begin{pmatrix} u_{0,0} & u_{0,1} & \cdots & u_{0,n-2} & u_{0,n-1} \\ 0 & u_{1,1} & \cdots & u_{1,n-2} & u_{1,n-1} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & u_{n-2,n-2} & u_{n-2,n-1} \\ 0 & 0 & \cdots & 0 & u_{n-1,n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix} \tag{11}$$

これを x について解くと、

$$\begin{aligned} x_{n-1} &= c_{n-1}/u_{n-1,n-1} \\ x_{n-2} &= (c_{n-2}-u_{n-2,n-1}\cdot x_{n-1})/u_{n-2,n-2} \\ x_{n-3} &= (c_{n-3}-u_{n-3,n-2}\cdot x_{n-2}-u_{n-3,n-1}\cdot x_{n-1})/u_{n-3,n-3} \\ &\ \vdots \\ x_{n-k} &= (c_{n-k}-\sum_{i=1}^{k-1} u_{n-k,n-i}\cdot x_{n-i})/u_{n-k,n-k} \quad \text{where}(1 \leqq k \leqq n) \end{aligned} \tag{12}$$

プログラム例

入力行列は既にファイル(c:¥lu_test.txt)にあるものとする

c:¥lu_data.txtの中身

3	2	6	1	17
2	4	1	6	23
5	4	1	3	23
3	2	5	6	26

これは、以下の方程式を表しているものとする。

$$
\begin{aligned}
3x_0 + 2x_1 + 6x_2 + x_3 &= 17 \\
2x_0 + 4x_1 + x_2 + 6x_3 &= 23 \\
5x_0 + 4x_1 + x_2 + 3x_2 &= 23 \\
3x_0 + 2x_1 + 5x_2 + 6x_3 &= 26
\end{aligned}
$$

```c
#include "stdafx.h"
int _tmain(int argc, _TCHAR* argv[])
{
  const int M = 4;
  double a[M][M];
  double b[M];
  double c[M];
  double l[M][M];
  double u[M][M];
  double x[M];
  int i,j,k;
  FILE *fp;
  char line[1000];
  int ch;

  fp = fopen("c:¥¥lu_data.txt", "r"); /* データの読み込み  */
  for(i = 0; i < M; i++){
    for(j = 0; j < M; j++){
      fscanf(fp,"%lf",&a[i][j]);
    }
    fscanf(fp,"%lf",&b[i]);
  }
  fclose(fp);

  for(i = 0; i < M; i++){                    /* L行列、U行列を1と0で初期化 */
    for(j = 0; j < M; j++){
      u[i][j] = 0;
      if(i == j)
        l[i][j] = 1;
      else
        l[i][j] = 0;
    }
  }

  for(i = 0; i < M; i++){
    for(j = i; j < M; j++){       /* U行列の生成 */
      u[i][j] = a[i][j];
      for(k = 0; k < i; k++){
        u[i][j] -= u[k][j] * l[i][k];
      }
    }
    for(j = i+1; j < M; j++){      /* L行列の生成 */
      l[j][i] = a[j][i];
      for(k = 0; k < i; k++){
        l[j][i] -= u[k][i] * l[j][k];
      }
      l[j][i] /= u[i][i];
    }
  }

  for(i = 0; i < M; i++){          /* c行列の生成 */
      c[i] = b[i];
      for(j = 0; j < i; j++){
        c[i] -= l[i][j] * c[j];
      }
  }
  for(i = M-1; i >= 0; i--){       /* x行列の生成 */
    x[i] = c[i];
    for(j = M-1; j > i; j--){
      x[i] -= u[i][j] * x[j];
    }
    x[i] /= u[i][i];
  }

  printf("入力行列¥n");             /* 入力行列の出力 */
  for(i = 0; i < M; i++){
    for(j = 0; j < M; j++){
      printf("%10.5lf ",a[i][j]);
    }
    printf("%10.5lf¥n",b[i]);
  }
  printf("¥nL行列¥n");             /* L行列の出力 */
  for(i = 0; i < M; i++){
    for(j = 0; j < M; j++){
      printf("%10.5lf ",l[i][j]);
    }
    printf("¥n");
  }

  printf("¥nU行列¥n");            /* U行列の出力 */
  for(i = 0; i < M; i++){
      for(j = 0; j < M; j++){
        printf("%10.5lf ",u[i][j]);
      }
      printf("¥n");
  }
  printf("¥n");
  for(i = 0; i < M; i++){       /* 解の出力 */
    printf("x%d = %10.5lf¥n",i,x[i]);
  }
  printf( "Enter キーを1,2回押してください. プログラムを終了します¥n");
  ch = getchar();
  ch = getchar();
  return 0;
}
```

実行結果

```
入力行列
    3.00000     2.00000     6.00000     1.00000    17.00000
    2.00000     4.00000     1.00000     6.00000    23.00000
    5.00000     4.00000     1.00000     3.00000    23.00000
    3.00000     2.00000     5.00000     6.00000    26.00000

L行列
    1.00000     0.00000     0.00000     0.00000
    0.66667     1.00000     0.00000     0.00000
    1.66667     0.25000     1.00000     0.00000
    1.00000     0.00000     0.12121     1.00000

U行列
    3.00000     2.00000     6.00000     1.00000
    0.00000     2.66667    -3.00000     5.33333
    0.00000     0.00000    -8.25000     0.00000
    0.00000     0.00000     0.00000     5.00000

x0 =     2.00000
x1 =     1.50000
x2 =     1.00000
x3 =     2.00000
Enter キーを1,2回押してください. プログラムを終了します
```

LU分解できない場合

- $A_{i,i}$や$u_{k,k}$等の対角上の要素が0であれば、0除算が発生しLU分解できない。

例

0	3	4	1	20
3	2	2	2	19
5	6	2	3	27
4	8	5	2	14

ピボッティング(1)

- これを避けるために、0である要素を含む行と0でない要素を含む行を入れ替える（ピボッティング）。

例

$$
\begin{matrix}
0 & 3 & 4 & 1 & 20 \\
3 & 2 & 2 & 2 & 19 \\
5 & 6 & 2 & 3 & 27 \\
4 & 8 & 5 & 2 & 14
\end{matrix}
\quad \Rightarrow \quad
\begin{matrix}
3 & 2 & 2 & 2 & 19 \\
5 & 6 & 2 & 3 & 27 \\
4 & 8 & 5 & 2 & 14 \\
0 & 3 & 4 & 1 & 20
\end{matrix}
$$

ピボッティング(2)

- 0でなくても、0に近い小さな値で割ったときは丸め誤差が大きくなる。
- この誤差を小さくするために、0でなくても、最も絶対値の大きな値を含む行と最も絶対値の小さな値を含む行 を入れ替える方がよい。

解が求まらない場合

- 以下の例のような場合は、連立一次方程式は解くことができない。

$$
\begin{cases}
x_0 + 2x_1 + 2x_2 + 2x_3 = 19 \\
x_0 + 2x_1 + 2x_2 + 2x_3 = 17 \\
x_0 + x_1 + x_2 + 2x_2 = 14 \\
x_1 + x_2 + x_3 \ = 10
\end{cases}
$$

$$
\begin{cases}
x_0 + 2x_1 + 2x_2 + 2x_3 = 17 \\
x_0 + x_1 + x_2 + 2x_2 = 14 \\
x_1 + x_2 + x_3 \ = 10
\end{cases}
$$