

数値解析 (塩田)

— LU 分解法 —

状況設定

n 次正則行列 $A = (a_{ij})$ と n 次ベクトル $\mathbf{b} = (b_i)$ に対して、方程式 $A\mathbf{x} = \mathbf{b}$ の解 $\mathbf{x}$ を求める。

LU 分解法

Step 1° n 次正則行列 $A = (a_{ij})$ を次の形に分解する:

$$A = LU,$$

$$L = \begin{pmatrix} \ell_{11} & & & \mathbf{O} \\ \ell_{21} & \ell_{22} & & \\ \vdots & & \ddots & \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{pmatrix}, \quad U = \begin{pmatrix} 1 & u_{12} & \cdots & u_{1n} \\ & 1 & \cdots & u_{2n} \\ & & \ddots & \vdots \\ \mathbf{O} & & & 1 \end{pmatrix}$$

Step 2° $L\mathbf{y} = \mathbf{b}$ の解 $\mathbf{y}$ を求める。

Step 3° $U\mathbf{x} = \mathbf{y}$ の解 $\mathbf{x}$ を求める。(すると $\mathbf{x}$ は $A\mathbf{x} = \mathbf{b}$ の解となる。)

Step 1° の例

$$\bullet n = 2 \text{ のとき } \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a & 0 \\ c & d - \frac{bc}{a} \end{pmatrix} \begin{pmatrix} 1 & \frac{b}{a} \\ 0 & 1 \end{pmatrix}$$

$$\bullet n = 3 \text{ のとき } \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ d & e' & 0 \\ g & h' & i' \end{pmatrix} \begin{pmatrix} 1 & \frac{b}{a} & \frac{c}{a} \\ 0 & 1 & j \\ 0 & 0 & 1 \end{pmatrix}$$

ただし

$$e' = e - \frac{bd}{a}, \quad h' = h - \frac{bg}{a}, \quad j = \frac{1}{e'} \left(f - \frac{cd}{a} \right), \quad i' = i - \frac{cg}{a} - jh'$$

LU 分解法の使いどころ

同じ A に対して $\mathbf{b}$ だけが異なる方程式 $A\mathbf{x} = \mathbf{b}$ を複数個解く場合に有効。
(2 個目以降は計算量の少ない Step 2°-3° のみ実行すれば良い。)

Step 1° の手順

$A = LU$ という式を成分で書いて、1 行目の左から

$$\begin{pmatrix} \longrightarrow \\ \longrightarrow \\ \vdots \\ \longrightarrow \end{pmatrix}$$

の順に見てゆくと、 i 行目では

$$\ell_{i1}, \dots, \ell_{ii}, \quad u_{i,i+1}, \dots, u_{in}$$

の順に未知数が求まってゆく (下線は既に求まっていて、赤字だけが未知数) :

$$\begin{aligned} 1 \text{ 行目: } \underline{a_{11}} &= \ell_{11} \\ \underline{a_{12}} &= \underline{\ell_{11}} \times \mathbf{u_{12}} \\ &\vdots \\ \underline{a_{1n}} &= \underline{\ell_{11}} \times \mathbf{u_{1n}} \end{aligned}$$

$$\begin{aligned} 2 \text{ 行目: } \underline{a_{21}} &= \ell_{21} \\ \underline{a_{22}} &= \underline{\ell_{21}} \times \underline{u_{12}} + \ell_{22} \\ \underline{a_{23}} &= \underline{\ell_{21}} \times \underline{u_{13}} + \underline{\ell_{22}} \times \mathbf{u_{23}} \\ &\vdots \\ \underline{a_{2n}} &= \underline{\ell_{21}} \times \underline{u_{1n}} + \underline{\ell_{22}} \times \mathbf{u_{2n}} \end{aligned}$$

$$\begin{aligned} i \text{ 行目: } i \geq j \text{ のとき} \\ \underline{a_{ij}} &= \sum_{1 \leq k < j} \underline{\ell_{ik}} \times \underline{u_{kj}} + \ell_{ij} \\ &\quad \left(\begin{array}{l} \ell_{ik} \text{ たちは } j-1 \text{ 列目までに求まっていて、} \\ u_{kj} \text{ たちは } i-1 \text{ 行目までに求まっている。} \end{array} \right) \end{aligned}$$

$$\begin{aligned} i < j \text{ のとき} \\ \underline{a_{ij}} &= \sum_{1 \leq k < i} \underline{\ell_{ik}} \times \underline{u_{kj}} + \underline{\ell_{ii}} \times \mathbf{u_{ij}} \\ &\quad \left(\begin{array}{l} \ell_{ik} \text{ たちは } i \text{ 列目までに全て求まっていて、} \\ u_{kj} \text{ たちは } i-1 \text{ 行目までに求まっている。} \end{array} \right) \end{aligned}$$

Step 1° のアルゴリズム ver.1

$A = (a_{ij})$, $L = (\ell_{ij})$, $U = (u_{ij})$ として、

```
for  $i = 1$  to  $n$ 
  for  $j = 1$  to  $i$ 
     $\ell_{ij} = a_{ij} - \sum_{1 \leq k < j} \ell_{ik} u_{kj}$ 
  for  $j = i + 1$  to  $n$ 
     $u_{ij} = \left( a_{ij} - \sum_{1 \leq k < i} \ell_{ik} u_{kj} \right) / \ell_{ii}$ 
```

Step 1° のアルゴリズム ver.2

メモリ節約の為に L , U の為の変数を使わない方法。 $A = (a_{ij})$ について次の手続きを行うと、最終的に

$$\begin{cases} \ell_{ij} = a_{ij} & (i \geq j \text{ のとき}) \\ u_{ij} = a_{ij} & (i < j \text{ のとき}) \end{cases}$$

となっている。

```
for  $i = 1$  to  $n - 1$ 
   $c = a_{ii}$ 
  for  $j = i + 1$  to  $n$ 
     $a_{ij} = a_{ij} / c$ 
  for  $k = i + 1$  to  $n$ 
    for  $j = i + 1$  to  $n$ 
       $a_{kj} = a_{kj} - a_{ki} a_{ij}$ 
```

Step 2° のアルゴリズム

```
for  $i = 1$  to  $n$ 
   $y_i = \left( b_i - \sum_{j=1}^{i-1} \ell_{ij} y_j \right) / \ell_{ii}$ 
```

Step 3° のアルゴリズム

```
for  $i = n$  downto  $1$ 
   $x_i = y_i - \sum_{j=i+1}^n u_{ij} x_j$ 
```

部分ピボット選択を用いる場合の注意

- (a) 行の入れ換えを Step 2° に反映させなければならない。
- (1) 配列 $s[]$ を用意し、初期値は $s[i] = i$.
 - (2) i 行目と p 行目を入れ換えたら $\text{swap}(s[i], s[p])$
 - (3) Step 2° では $b[i]$ の代わりに $b[s[i]]$ を入れて計算する
- (b) ガウスの消去法は 0 になっている成分は入れ換えなくてよいのに対し、LU 分解法では行を全て入れ換える。

配列番号を $0 \sim n-1$ にする場合

Step 1° ver.1

```
for  $i = 0$  to  $n - 1$ 
  for  $j = 0$  to  $i$ 
     $\ell_{ij} = a_{ij} - \sum_{0 \leq k < j} \ell_{ik} u_{kj}$ 
  for  $j = i + 1$  to  $n - 1$ 
     $u_{ij} = \left( a_{ij} - \sum_{0 \leq k < i} \ell_{ik} u_{kj} \right) / \ell_{ii}$ 
```

Step 1° ver.2

```
for  $i = 0$  to  $n - 2$ 
   $c = a_{ii}$ 
  for  $j = i + 1$  to  $n - 1$ 
     $a_{ij} = a_{ij} / c$ 
  for  $k = i + 1$  to  $n - 1$ 
    for  $j = i + 1$  to  $n - 1$ 
       $a_{kj} = a_{kj} - a_{ki} a_{ij}$ 
```

Step 2°

```
for  $i = 0$  to  $n - 1$ 
   $y_i = \left( b_i - \sum_{j=0}^{i-1} \ell_{ij} y_j \right) / \ell_{ii}$ 
```

Step 3°

```
for  $i = n - 1$  downto  $0$ 
   $x_i = y_i - \sum_{j=i+1}^{n-1} u_{ij} x_j$ 
```

実行例

A :

-8.884114715682	9.348650425369	-7.018980419738	-5.939152918728	0.163633353153
9.154459163153	8.433842425437	3.066621624430	2.075486291235	4.915200003849
-4.115226051824	8.171619464723	-7.783609413441	0.536247822706	-2.617435591583
-4.237237821444	3.066381929939	5.711188882455	-2.650166448508	-7.270667905580
-8.225384176814	-9.105048831136	-4.059792568004	2.970680311774	3.658045108271

b :

1.000000000000
1.000000000000
1.000000000000
1.000000000000
1.000000000000

----- Step 1 LU 分解 : $A = L U$ を満たす下三角行列 L と上三角行列 U を求める -----

L :

-8.884114715682	0.000000000000	0.000000000000	0.000000000000	0.000000000000
9.154459163153	18.066973163670	0.000000000000	0.000000000000	0.000000000000
-4.115226051824	3.841215028144	-3.646614381130	0.000000000000	0.000000000000
-4.237237821444	-1.392414070512	8.737791647928	9.808075928379	0.000000000000
-8.225384176814	-17.760524783923	-1.656533628103	2.609729635138	14.475893526647

U :

1.000000000000	-1.052288351125	0.790059633893	0.668513758410	-0.018418644782
0.000000000000	1.000000000000	-0.230583562214	-0.223855737953	0.281387075151
0.000000000000	0.000000000000	1.000000000000	-1.137277114778	1.034960197534
0.000000000000	0.000000000000	0.000000000000	1.000000000000	-1.631326209947
0.000000000000	0.000000000000	0.000000000000	0.000000000000	1.000000000000

----- 検算 : $A = L U$ を確認 -----

L U :

-8.884114715682	9.348650425369	-7.018980419738	-5.939152918728	0.163633353153
9.154459163153	8.433842425437	3.066621624430	2.075486291235	4.915200003849
-4.115226051824	8.171619464723	-7.783609413441	0.536247822706	-2.617435591583
-4.237237821444	3.066381929939	5.711188882455	-2.650166448508	-7.270667905580
-8.225384176814	-9.105048831136	-4.059792568004	2.970680311774	3.658045108271

A :

-8.884114715682	9.348650425369	-7.018980419738	-5.939152918728	0.163633353153
9.154459163153	8.433842425437	3.066621624430	2.075486291235	4.915200003849
-4.115226051824	8.171619464723	-7.783609413441	0.536247822706	-2.617435591583
-4.237237821444	3.066381929939	5.711188882455	-2.650166448508	-7.270667905580
-8.225384176814	-9.105048831136	-4.059792568004	2.970680311774	3.658045108271

----- Step 2 前進代入 : $L y = b$ を解く -----

y :

-0.112560455600
0.112383523005
-0.028820980844
0.094959573271
0.122588330418

----- Step 3 後退代入 : $U x = y$ を解く -----

x :

-0.254427030690
0.185356899839
0.179734773523
0.294941129716
0.122588330418

----- 検算 : $A x = b$ を確認 -----

A x :

1.000000000000
1.000000000000
1.000000000000
1.000000000000
1.000000000000

b :

1.000000000000
1.000000000000
1.000000000000
1.000000000000
1.000000000000

最大誤差 = 0.000000000000