

スライド1 LU分解 p.49

1.  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ をLU分解せよ。

ヒント

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix} \begin{bmatrix} b & c \\ 0 & d \end{bmatrix}$$

2.  $\begin{bmatrix} 2 & 6 & 4 \\ 5 & 7 & 9 \end{bmatrix}$ をLU分解せよ。

スライド2

部分ピボット選択付きガウス消去法

- (1) k番方程式と ip番方程式を入れ換える  
 (2) k番方程式を  $\alpha_{ik}$ 倍して i番方程式( $i=k+1, k+2, \dots, n-1$ )に加える

$$P_k = \begin{bmatrix} 1 & & & & & & & & & \\ & \dots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 0 & \dots & 1 & & & & \\ & & & \vdots & & \vdots & & & & \\ & & & 1 & \dots & 0 & & & & \\ & & & & & & 1 & & & \\ & & & & & & & \dots & & \\ & & & & & & & & 1 & \\ & & & & & & & & & 1 \end{bmatrix}, G_k = \begin{bmatrix} 1 & & & & & & & & & \\ & \dots & & & & & & & & \\ & & 1 & & & & & & & \\ & & \alpha_{k+1,k} & 1 & & & & & & \\ & & \vdots & & & & \dots & & & \\ & & \alpha_{n-1,k} & & & & & & & 1 \end{bmatrix}$$

スライド3 LU分解 p.50

$$G_{n-2}P_{n-2} \cdots G_0P_0Ax = G_{n-2}P_{n-2} \cdots G_0P_0b$$

$$PA = LU$$

U: 上三角行列, L: 下三角行列, P: 置換行列

$$U = \begin{bmatrix} u_{00} & u_{01} & \dots & u_{0,n-1} \\ & u_{11} & \dots & u_{1,n-1} \\ & & \dots & \vdots \\ & & & u_{n-1,n-1} \end{bmatrix}, L = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ l_{10} & 1 & 0 & 0 & \dots & 0 \\ l_{20} & l_{21} & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & & \dots & \dots & 0 \\ l_{n-1,0} & l_{n-1,1} & \dots & \dots & l_{n-1,n-2} & 1 \end{bmatrix}$$

スライド4 LU分解

フロベニウス(Frobenius)行列

$$f_k = (0, \dots, 0, f_{k+1,k}, \dots, f_{n-1,k})^T$$

$$F_k = I - f_k e_k^T = \begin{bmatrix} 1 & 0 & 0 \\ & 1 & \\ 0 & -f_{k+1,k} & \\ & \vdots & \\ & -f_{n-1,k} & \\ & & 1 \end{bmatrix}$$

$$F_k^{-1} = I + f_k e_k^T = \begin{bmatrix} 1 & 0 & 0 \\ & 1 & \\ 0 & f_{k+1,k} & \\ & \vdots & \\ & f_{n-1,k} & \\ & & 1 \end{bmatrix}$$

スライド5 LU分解

フロベニウス(Frobenius)行列

$$F_0^{-1} \dots F_{n-2}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ f_{1,0} & 1 & 0 \\ & f_{2,1} & \\ \vdots & \vdots & 0 \\ f_{n-1,0} & f_{n-1,1} & 1 \end{bmatrix}$$

ガウス消去法 前進消去の第k段の操作(行の交換が不要の場合)

$$A^{(n-1)} = F_{n-2} A^{(n-2)}$$

ただし,  $f_k = (0, \dots, 0, f_{k+1,k}, \dots, f_{n-1,k})^T, f_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}$

$$A^{(n-1)} = F_{n-2} \dots F_0 A \Leftrightarrow A = F_0^{-1} \dots F_{n-1}^{-1} A^{(n-1)}$$

スライド6 LU分解

以下の行列を計算せよ.

$$\begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & 0 & 1 \end{bmatrix}^{-1}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & c & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ ac + b & c & 1 \end{bmatrix}^{-1}$$

$$A^{(k)} = F_{k-1}P_{k-1}A^{(k-1)}$$

$P_{k-1}$ :第  $k$  行と第  $p_k$  行を交換する置換行列

$F_{k-1}$  の成分  $f_{p_{k-1},k-1} = \frac{a_{k-1,k-1}^{(k-1)}}{a_{p_{k-1},k-1}^{(k-1)}}$ ,  $f_{i-1,k-1} = \frac{a_{i-1,k-1}^{(k-1)}}{a_{p_{k-1},k-1}^{(k-1)}}$  ( $i = k, \dots, n-1, i \neq p_{k-1}$ )

$$A^{(1)} = F_0P_0A$$

$$A^{(2)} = F_1P_1A^{(1)} = F_1(P_1F_0P_1)(P_1P_0)A$$

$$A^{(3)} = F_2P_2A^{(2)} = F_2(P_2P_1F_0P_1P_2)(P_2P_1P_0)A$$

$$A^{(4)} = \dots$$

$$G_{k-1} = P_{n-2} \cdots P_k F_{k-1} P_k \cdots P_{n-2}$$

$$A^{n-1} = (G_{n-1} \cdots G_0)(P_{n-2} \cdots P_0A)$$

置換行列

$$\sigma(i) = r, \sigma(r) = i, \sigma(j) = j \quad (j \in N_n, j \neq i, r)$$

命題

$i, r \geq k+1$  ならば,

$$PF_{k-1}P = I - (Pf_{k-1}e_{k-1}^T)$$

$$A' = \begin{bmatrix} u_{00} & u_{01} & u_{02} & u_{03} & \cdots & u_{0,n-1} \\ l_{10} & u_{11} & u_{12} & u_{13} & \cdots & u_{1,n-1} \\ l_{20} & l_{21} & u_{22} & u_{23} & \cdots & u_{2,n-1} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & u_{n-2,n-1} \\ l_{n-1,0} & l_{n-1,1} & \cdots & \cdots & l_{n-1,n-2} & u_{n-1,n-1} \end{bmatrix}$$

Input  $A, \varepsilon$

for  $k = 0, 1, \dots, n-2$

Exchange 2 rows if necessary

$P_k \leftarrow ip$

if  $ip \neq k$  then

for  $j = k, k+1, \dots, n-1$

$a_{kj} \leftrightarrow a_{ip,j}$

end for

end if

for  $i = k+1, k+2, \dots, n-1$

$\alpha \leftarrow -\frac{a_{ik}}{a_{kk}}$

$a_{ik} \leftarrow \alpha$

for  $j = k+1, k+2, \dots, n-1$

$a_{ij} \leftarrow a_{ij} + a_{kj}$

end for

end for

end for

Output  $A, P$

分解アルゴリズム

Input  $A, b, P$

for  $k = 0, 1, \dots, n-2$

$b_k \leftrightarrow b_{P_k}A$

for  $i = k+1, k+2, \dots, n-1$

$b_i \leftarrow b_i + a_{ik}b_k$

end for

end for

for  $k = n-1, n-2, \dots, 0$

$b_k \leftarrow \frac{b_k - \sum_{j=k+1}^{n-1} a_{kj}b_j}{a_{kk}}$

end for

Output  $b$

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define N 4 /* N 次正方行列 */
void input_matrix(double a[N][N],char c,FILE* fin, FILE* fout);
void input_vector(double b[N],char c,FILE* fin,FILE* fout);
void lu_decomp(double a[N][N],int p[N]);
void lu_solve(double a[N][N],double b[N],int p[N]);
int main(void){
    FILE *fin, *fout;
    double a[N][N], b[N];
    int I, p[N]; /* p[0...N-2]を利用, p[N-1]は未使用 */
    if((fin=fopen("input_lu.dat","r"))==NULL) exit(1);
    if((fout=fopen("output_lu.dat","w"))==NULL)
        exit(1);
    input_matrix(a,'A',fin,fout);
    input_vector(b,'b',fin,fout);
    lu_decomp(a,b); lu_solve(a,b,p);
    fprintf(fout,"Ax=b の解は次の通りです\n");
    for(i=0;i<N;i++){ fprintf(fout,"%f\n",b[i]); }
    fclose(fin); fclose(fout);
    return 0;
}
```

```
void lu_decomp(double a[N][N],int p[N]){
    int i,j,k,ip;
    double alpha, tmp;
    double amax, eps=pow(2.0,-50.0);
    for(k=0;k<N-1;k++){
        amax=fabs(a[k][k]); ip=k; /* ピボットの選択 */
        for(i=k+1;i<N;i++){
            if(fabs(a[i][k])>amax){
                amax=fabs(a[i][k]); ip=i;
            }
        }
        if(amax<eps) { /* 正則性の判定 */
            printf("行列は正則ではない!!\n"); exit(1);
        }
        p[k]=ip;
        if(ip!=k){
            for(j=k;j<N;j++){
                tmp=a[k][j]; a[k][j]=a[ip][j]; a[ip][j]=tmp;
            }
        }
        for(i=k+1;i<N;i++){ /* 前進消去 */
            alpha=a[i][k]/a[k][k];
            a[i][k]=alpha;
            for(j=k+1;j<N;j++){
                a[i][j]=a[i][j]+alpha*a[k][j];
            }
        }
    }
}
```

```
void lu_solve(double a[N][N],double b[N],int p[N]){
    int i,j,k;
    double tmp;
    for(k=0;k<N-1;k++){
        tmp=b[k]; b[k]=b[p[k]]; b[p[k]]=tmp;
        /* 右辺の行変換 */
        for(i=k+1;i<N;i++){ /* 前進代入 */
            b[i]=b[i]+a[i][k]*b[k];
        }
    }
    b[N-1]=b[N-1]/a[N-1][N-1];
    for(k=N-2;k>=0;k--){ /* 後退代入 */
        tmp=b[k];
        for(j=k+1;j<N;j++){
            tmp=tmp-a[k][j]*b[j];
        }
        b[k]=tmp/a[k][k];
    }
}
```

スライド6 コレスキー分解 p.57

正定値行列とその性質

コレスキー分解は正定値(positive definite)対称行列の場合に有効

n次元正方行列Aがx≠0である任意のベクトルxに対してn次元の内積

$$(x, Ax) > 0$$

定理 3.2 n次正方行列Aが正定値ならば、Aは正則である。

定理 3.7 n次正方行列Aが正定値対称ならば、ある下三角行列LによってA=LL<sup>t</sup>と分解できる。