1　連立一次方程式の解法

連立一次方程式,

$$\begin{aligned} b_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ b_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ &\vdots \\ b_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{aligned}$$

を解くことを考えよう．この方程式は行列の表現を使えば

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

と書ける．$\mathbf{A} := (a_{ij})$, $\mathbf{b} := (b_i)$, $\mathbf{x} := (x_i)$ として

$$\mathbf{b} = \mathbf{A}\mathbf{x}$$

であるから, 逆行列 $\mathbf{A}^{-1}$ を求める事ができれば

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

で与えられる．特に次の定理が良く知られている．

定理 1. $|\mathbf{A}| \neq 0$ の時逆行列 $\mathbf{A}^{-1}$ が存在する．

よく知られたこの定理の証明はクラメルの公式

$$\mathbf{A}^{-1} = (\frac{|\mathbf{B}_{ij}|}{|\mathbf{A}|})_{ij} \quad (\mathbf{B}_{ij} \text{ は } \mathbf{A} \text{ の } (i,j)\text{-余因子})$$

により逆行列を構成するものである．

この公式を用いれば”原理的には”逆行列を求める事ができ, 方程式 $\mathbf{b} = \mathbf{A}\mathbf{x}$ の解を求める事ができる．しかし, この公式は (特に高次元の場合) 解法としては適さない．行列式は

$$|\mathbf{A}| = \sum_{\sigma \in \mathfrak{S}_n} \operatorname{sign} \sigma \prod_{i=1}^{n} a_{\sigma(i)i}$$

で与えられるが n 次置換 $\mathfrak{S}_n$ の濃度が $n!$ である事に注意すれば上式は $nn!$ の乗算を含む.[1]

[1]これではいくら計算機が早くても追いつかない．　実際 $10! = 3628800$, $20! = 2432902008176640000$, $30! \fallingdotseq 265000000000000000000000000000000$ といった具合である．

1.1 ガウスの消去法

いわゆる掃き出し法の事である．

まずは具体的に解くところを見てみる．

例 1.

$$\begin{cases} 10 = x_1 + 3x_2 + 4x_3 & ① \\ 7 = 2x_1 + x_2 + 5x_3 & ② \\ 11 = 6x_1 + 5x_2 + x_3 & ③ \end{cases}$$

④ := ①, ⑤ := ② − 2 × ①, ⑥ := ③ − 6 × ① とすれば

$$\begin{cases} 10 = x_1 + 3x_2 + 4x_3 & ④ \\ -13 = -5x_2 - 3x_3 & ⑤ \\ -49 = -13x_2 - 23x_3 & ⑥ \end{cases}$$

⑧ := $\frac{⑤}{-5}$, ⑦ := ④ − 3 × ⑧, ⑨ := ③ + 13 × ⑧ とすれば

$$\begin{cases} \frac{11}{5} = x_1 + \frac{11}{5}x_3 & ⑦ \\ \frac{13}{5} = x_2 + \frac{3}{5}x_3 & ⑧ \\ -\frac{76}{5} = -\frac{76}{5}x_3 & ⑨ \end{cases}$$

⑫ := $-\frac{5}{76}$ × ⑨, ⑩ := ⑦ − $\frac{11}{5}$ × ⑫, ⑪ := ⑧ − $\frac{3}{5}$ × ⑫ とすれば

$$\begin{cases} 0 = x_1 & ⑩ \\ 2 = x_2 & ⑪ \\ 1 = x_3 & ⑫ \end{cases}$$

一般的にしようと思えば k 列目の掃き出しに用いる行をその時の k 列目の係数が 0 出ない様に選ぶ必要がある．

例 2.

$$\begin{cases} 9 = x_1 + x_2 + 2x_3 & ① \\ 9 = 2x_1 + 2x_2 + x_3 & ② \\ 7 = 2x_1 + x_2 + x_3 & ③ \end{cases}$$

④ := ①, ⑤ := ② − 2 × ①, ⑥ := ③ − 2 × ① とすれば

$$\begin{cases} 9 = x_1 + x_2 + 2x_3 & ④ \\ -9 = -3x_3 & ⑤ \\ -11 = -x_2 - 3x_3 & ⑥ \end{cases}$$

⑤ の x_2 の係数が 0 であるから 0 より”下の列”(未だ掃き出しに用いていない列) で x_2 の係数が非 0 の列⑥と入れ替える．即ち，④′ := ④, ⑤′ := ⑥,

⑥$'$:= ⑤ として

$$\begin{cases} 9 &= x_1 + x_2 + 2x_3 & ④' \\ -11 &= -x_2 - 3x_3 & ⑤' \\ -9 &= -3x_3 & ⑥' \end{cases}$$

⑧ := −⑤$'$, ⑦ := ④$'$ − ⑧, ⑨ := ⑥$'$ とすれば

$$\begin{cases} -2 &= x_1 - x_3 & ⑦ \\ 11 &= x_2 + 3x_3 & ⑧ \\ -9 &= -3x_3 & ⑨ \end{cases}$$

⑫ := $-\frac{1}{3}\times$ ⑨, ⑩ := ⑦ + ⑫, ⑪ := ⑧ − 3 × ⑫ とすれば

$$\begin{cases} 1 &= x_1 & ⑩ \\ 2 &= x_2 & ⑪ \\ 3 &= x_3 & ⑫ \end{cases}$$

ガウスの消去法では乗算の数はおよそ $n^3/2$ である.

一意解の存在する一次連立方程式がガウスの消去法によって必ず解ける事を証明しよう.

定理 2. $|\mathbf{A}| \neq 0$, $\mathbf{A}^{(0)} := \mathbf{A}$ とする. $\sigma_k \geq k$ を適当に取れば次の様にして $\mathbf{A}^{(k)}$ が帰納的に well defined である. (即ち, $\hat{a}_{kk}^{(k-1)} = a_{\sigma_k k}^{(k-1)} \neq 0$ なる $\sigma_k \geq k$ が取れる)

$$\mathbf{P}^{(k)} := \begin{pmatrix} \mathbf{I}_{k-1} & & 0 & & 0 \\ & 0 & \dots\ 0 & 1 & \\ & \vdots & & 0 & \\ 0 & & \mathbf{I}_{\sigma_k-k-1} & & 0 \\ & 0 & & \vdots & \\ & 1 & 0\ \dots & 0 & \\ 0 & & 0 & & \mathbf{I}_{n-\sigma_k-k} \end{pmatrix}, \qquad (\hat{a}_{ij}^{(k-1)}) := \mathbf{P}^{(k)}\mathbf{A}^{(k-1)},$$

$$\mathbf{M}^{(k)} := \begin{pmatrix} & -\hat{a}_{1k}^{(k-1)}/\hat{a}_{kk}^{(k-1)} & \\ \mathbf{I}_{k-1} & \vdots & 0 \\ & -\hat{a}_{k-1k}^{(k-1)}/\hat{a}_{kk}^{(k-1)} & \\ 0 & 1/\hat{a}_{kk}^{(k-1)} & 0 \\ & -\hat{a}_{k+1k}^{(k-1)}/\hat{a}_{kk}^{(k-1)} & \\ 0 & \vdots & \mathbf{I}_{n-k} \\ & -\hat{a}_{nk}^{(k-1)}/\hat{a}_{kk}^{(k-1)} & \end{pmatrix}, \qquad \mathbf{A}^{(k)} := \mathbf{M}^{(k)}\mathbf{P}^{(k)}\mathbf{A}^{(k-1)}$$

また $A^{(n)} = \mathbf{I}$ となる.

note

σ_k は掃き出しに用いる行の選択であり, $\mathbf{P}^{(k)}$ はその行の入れ替えに相当し, $\mathbf{M}^{(k)}$ が実際の掃き出し操作に相当する.

特に $\mathbf{I}_n = \mathbf{A}^{(n)} = \mathbf{M}^{(n)}\mathbf{P}^{(n)}\mathbf{M}^{(n-1)}\mathbf{P}^{(n-1)}\cdots\mathbf{M}^{(1)}\mathbf{P}^{(1)}\mathbf{A}$, 即ち $\mathbf{M}^{(n)}\mathbf{P}^{(n)}\cdots\mathbf{M}^{(1)}\mathbf{P}^{(1)} = \mathbf{A}^{-1}$ であり. 我々は $\mathbf{A}$ の逆行列を求めた事になる.

例 1 では掃き出しに用いる行が掃き出される列と常に等しく, $\sigma_k = k$, 即ち $\mathbf{P}^{(n)} = \mathbf{I}$ である. この時 $\mathbf{M}^{(n)}\cdots\mathbf{M}^{(1)} = \mathbf{A}^{-1}$ となり,

$$\mathbf{b} = \mathbf{A}\mathbf{x}$$

に次々に両辺に $\mathbf{M}^{(k)}$ を作用させる事によって.

$$\mathbf{b}^{(n)} = \mathbf{A}^{(n)}\mathbf{x} = \mathbf{I}\mathbf{x} = \mathbf{x}$$

の形を導いた.

Proof. 特に $\mathbf{A}^{(k)}$ まで well def で $\mathbf{A}^{(k)}$ が次の形をしている

$$\mathbf{A}^{(k)} = \begin{pmatrix} \mathbf{I}_k & * \\ 0 & \end{pmatrix} \tag{1}$$

事を帰納的に示す.

定義より $|\mathbf{M}^{(l)}| = 1/\hat{a}_{ll}^{(l-1)}$, $|\mathbf{P}^{(l)}| = 1$, $|\mathbf{A}^{(k-1)}| = |\mathbf{M}^{(k-1)}||\mathbf{P}^{(k-1)}|\cdots|\mathbf{M}^{(1)}||\mathbf{P}^{(1)}||\mathbf{A}^{(0)}|$ であり, 正規性の仮定 $|\mathbf{A}^{(0)}| \neq 0$ より $|\mathbf{A}^{(k-1)}| \neq 0$. 特に

$$|\mathbf{A}^{(k-1)}| = \sum_{\sigma\in\mathfrak{S}_n} \operatorname{sign}\sigma \prod_{i=1}^{n} a_{\sigma(i)i}^{(k-1)}$$

であるので, ある $\sigma^{(k)} \in \mathfrak{S}_n$ があって $\prod_{i=1}^{n} a_{\sigma^{(k)}(i)i}^{(k-1)} \neq 0$ である. 更に (1) より $(\sigma^{(k)}(1), \sigma^{(k)}(2), \ldots, \sigma^{(k)}(k-1)) = (1, 2, \ldots, k-1)$ で無ければならない. この時 $\sigma^{(k)}(k) \geq k$ かつ $a_{\sigma^{(k)}(k)k}^{(k-1)} \neq 0$ であるから $\sigma_k := \sigma^{(k)}(k)$ と置けば $\mathbf{P}^{(k)}, \mathbf{M}^{(k)}$ が well def.

次に $\mathbf{P}^{(k+1)}$ の初めの k 列が単位行列と同じである事に注意すれば, $\mathbf{P}^{(k+1)}\mathbf{A}^{(k)}$ の初めの k 列は $\mathbf{A}^{(k)}$ のものと同じ, i.e., $\mathbf{P}^{(k+1)}\mathbf{A}^{(k)}$ が

$$\mathbf{P}^{(k+1)}\mathbf{A}^{(k)} = \begin{pmatrix} & \hat{a}_{1k+1}^{(k)} & \\ \mathbf{I}_k & \vdots & \\ & \hat{a}_{k-1k+1}^{(k)} & * \\ & \hat{a}_{kk+1}^{(k)} & \\ 0 & \vdots & \\ & \hat{a}_{nk+1}^{(k)} & \end{pmatrix}$$

の形をしている事がわかる．更に $\mathbf{M}^{(k+1)}$ の定義から

$$\mathbf{A}^{(k+1)} = \mathbf{M}^{(k+1)}\mathbf{P}^{(k+1)}\mathbf{A}^{(k)} = \begin{pmatrix} & 0 & \\ \mathbf{I}_k & \vdots & \\ & 0 & \\ & 1 & * \\ & 0 & \\ 0 & \vdots & \\ & 0 & \end{pmatrix} = \begin{pmatrix} \mathbf{I}_{k+1} & * \\ 0 & \end{pmatrix}$$

も容易に導かれる． $\square$

サンプル

```c
#include <stdio.h>
#include <math.h>
#define DIM 4
#define M_EPSILON 0.001

void print_equation(double A[][DIM], double b[]){ //表示関数
    int i,j;
    for(i=0;i<DIM;i++){
        for(j=0;j<DIM;j++) printf("%7.3f, ",A[i][j]);
        printf("|  %7.3f\n", b[i]);
    }
        printf("\n");
}

void swap_rows(double A[][DIM], int i, int j){ //行入れ替え関数
    int k;
    double temp;

    for(k=0;k<DIM;k++){
        temp = A[i][k]; A[i][k] = A[j][k]; A[j][k] = temp;
    }
}

int main(void){
    int i,j,k,l, sigma_k;
    double temp, max, alpha_jk, alpha_kk;
    double A[DIM][DIM] = {
        {1,  3, 4, 7},
        {2,  4, 5, 2},
        {6, 18, 1, 1},
        {3,  8, 3, 2}
    };
    double b[DIM] = {10, 7, 11, 0};

    print_equation(A,b);

    for(k=0;k<DIM;k++){ // k:掃き出しをする列

        //掃き出しに用いる行の探索
        max = M_EPSILON;
        sigma_k = -1;
        for(j=k;j<DIM;j++){
            if( max < fabs( A[j][k] ) ){
                max = fabs( A[j][k] );
                sigma_k = j;
            }
        }
        if( sigma_k == -1 ){
            printf("掃き出し行未発見:k=%d\n",k); return 1;
        }else{
            swap_rows(A,k,sigma_k);
            temp = b[k]; b[k]=b[sigma_k]; b[sigma_k]=temp;
        }

        //掃き出し行の正規化
        alpha_kk = A[k][k];
        for(l=0;l<DIM;l++) A[k][l] /= alpha_kk;
        b[k] = b[k]/alpha_kk;

        //掃き出し
        for(j=0;j<DIM;j++){
            if( j == k ) continue;
            alpha_jk = A[j][k];
            for(l=0;l<DIM;l++) A[j][l] = A[j][l] - alpha_jk*A[k][l];
            b[j] = b[j] - alpha_jk*b[k];
        }

        printf("sigma_%d = %d\n", k, sigma_k);
        print_equation(A,b);
    }

    printf("答え\n"); for(i=0;i<DIM;i++) printf("x[%d] = %f, ",i, b[i]);
    return 0;
}
```