5. 行列の固有値問題

n×n 正方行列 A に対する n 個の固有値 λ_i (i=1, 2, …, n) と対応する固有ベクトル $\mathbf{u}_i$ は次式を満たす。

$$A\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad \boldsymbol{u}_i = \begin{bmatrix} u_{1i} \\ u_{2i} \\ \vdots \\ u_{ni} \end{bmatrix}$$

これらはまとめて，つぎのように書ける。

$$AU = U\Lambda, \qquad U = [\boldsymbol{u}_1 | \boldsymbol{u}_2 | \cdots | \boldsymbol{u}_n], \qquad \Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

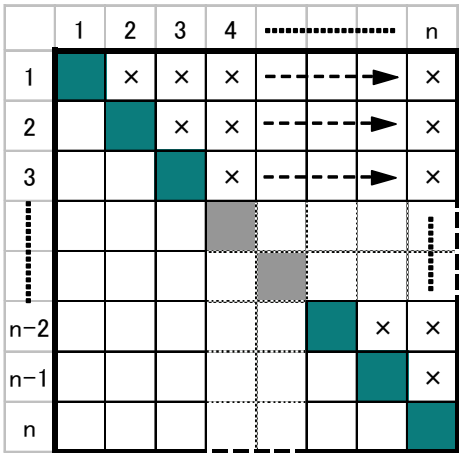
5.1 ヤコビ法

実数対称行列 A の固有値と対応する固有ベクトルをすべて求める方法であり，古典的ではあるが，行列サイズが 10 以下ならば現在でも使用されている。
直交行列 P を用いて A の相似変換（すなわち直交変換）$\tilde{A}=P^{T}AP$ を行い，対角行列に収束させていくと，対角要素が固有値となる（相似変換に関して固有値は不変）。

【アルゴリズム 5.1.1】

```
A:=与えられた行列
U:=I（単位行列）

for m=1,2,…          {収束条件を満たす（すべての非対角要素がεより小さくなる）まで繰り返す}
    for i=1 to n-1                          {Aの対角要素を除いた上三角部分を走査}
        for j=i+1 to n
            if |a_ij|≧ε then          {非対角要素で絶対値がεより大きい要素 a_ij を零にする変換}
                θ:=(1/2)cot^-1((a_ii-a_jj)/(2a_ij))
                A:=P^TAP                    {Aの直交変換（非対角要素 a_ij を零にする）}
                U:=UP                       {Uの変換}
            end
        end
    end
end
```



Aの直交変換 $\tilde{A}=P^TAP$ はつぎのように書き表せ，

$\cot 2\theta = (a_{ii}-a_{jj})/(2a_{ij})$

$\tilde{a}_{ii} = (1/2)(a_{ii}+a_{jj}) + (1/2)(a_{ii}-a_{jj})\cos 2\theta + a_{ij}\sin 2\theta$
$\tilde{a}_{jj} = (1/2)(a_{ii}+a_{jj}) - (1/2)(a_{ii}-a_{jj})\cos 2\theta - a_{ij}\sin 2\theta$
$\tilde{a}_{ij} = \tilde{a}_{ji} = 0$
$\tilde{a}_{ik} = \tilde{a}_{ki} = a_{ik}\cos\theta + a_{jk}\sin\theta$　　(k=1,2,…,n, ただし k≠i, j)
$\tilde{a}_{jk} = \tilde{a}_{kj} = -a_{ik}\sin\theta + a_{jk}\cos\theta$　　(k=1,2,…,n, ただし k≠i, j)
$\tilde{a}_{kl} = a_{kl}$　　(k, l=1,2,…,n, ただし k, l≠i, j)

Uの変換 $\tilde{U}=UP$ はつぎのようになる。
$\tilde{u}_{ki} = u_{ki}\cos\theta + u_{kj}\sin\theta$　(k=1,2,…,n)
$\tilde{u}_{kj} = -u_{ki}\sin\theta + u_{kj}\cos\theta$　(k=1,2,…,n)
$\tilde{u}_{kl} = u_{kl}$　　(k, l=1,2,…,n, ただし l≠i, j)

計算誤差を少なくするための注意：

実際の数値計算においては，θ を求めることなく，代数式から $\cos 2\theta$，$\sin 2\theta$，$\cos\theta$，$\sin\theta$ を求めるほうが，計算時間，計算誤差とも少なくできる。
一般に，$\cot\alpha$（あるいは $\tan\alpha=\cot^{-1}\alpha$）から $\cos\alpha$，$\sin\alpha$ を求める場合，

$|\cot\alpha|\le 1$ のとき　　$\sin\alpha = 1/\sqrt{1+\cot^2\alpha}$,　$\cos\alpha = \cot\alpha\cdot\sin\alpha$

$|\cot\alpha|\ge 1$ のとき　$|\tan\alpha| = 1/|\cot\alpha| \le 1$,　$\cos\alpha = 1/\sqrt{1+\tan^2\alpha}$,　$\sin\alpha = \tan\alpha\cdot\cos\alpha$

により計算すると，1よりはるかに大きい数の二乗を行わずにすむのでオーバーフローを避けることができ，かつ0による割算を避けることができる。$\cot 2\theta$ から $\cos 2\theta$，$\sin 2\theta$ を求めるときはこの関係を用いる。
さらに $\cos\theta$，$\sin\theta$ を求めるときには半角公式を用いるが，桁落ちを避けるため倍角公式も併用して，

$\cos 2\theta \ge 0$ のとき　$\cos\theta = \sqrt{\frac{1}{2}(1+\cos 2\theta)}$,　$\sin\theta = \sin 2\theta/(2\cos\theta)$

$\cos 2\theta \le 0$ のとき　$\sin\theta = \sqrt{\frac{1}{2}(1-\cos 2\theta)}$,　$\cos\theta = \sin 2\theta/(2\sin\theta)$

により計算する。
実用に耐える数値計算では，このような細部にまで注意が払われていることに留意しよう。

【プログラム5.1.1】

```
 1 /*  solve eigen-value problem for a symmetric matrix  */
 2 /*                by Jacobi's method                  */
 3
 4 #include <stdio.h>
 5 #include <math.h>
 6
 7 #define NMAX 20
 8
 9 int jacobi(float);
10
11 float a[NMAX][NMAX], u[NMAX][NMAX];
```

```
int   n;

void main(void)
{
    int mcum,i,j;
    float eps;

    printf("Order of a Symmetric matrix: n ? ");
    scanf("%d",&n);
    printf("Input Lower triangle of matrix\n");
    for(i=0;i<n;i++){
            printf("? A(%2d, j), j=1,%2d : ", i+1, i+1);
            for(j=0;j<=i;j++){
                    scanf("%g",&a[i][j]);
                    a[j][i]=a[i][j];
            }
    }
    printf("\nSymmetric matrix: A\n");
    for(i=0;i<n;i++){
            for(j=0;j<n;j++)
                    printf(" %15.6e", a[i][j]);
            printf("\n");
    }
    for(i=0;i<n;i++){
            for(j=0;j<i;j++)
                    u[i][j]=u[j][i]=0;
            u[i][i]=1;
    }

    mcum= 0;    /* initial set for cumulative number of transformations */
    while(printf("\nTolerance : eps ? "), scanf("%g",&eps)!=EOF){
            mcum += jacobi(eps);
            printf("cumulative no. of transformations=%d\n", mcum);
            printf("Eigen-values:\n");
            for(i=0;i<n;i++)
                    printf(" %15.6e", a[i][i]);
            printf("\nEigen-vectors:\n");
            for(i=0;i<n;i++){
                    for(j=0;j<n;j++)
                            printf(" %15.6e", u[i][j]);
                    printf("\n");
            }
    }
}

int jacobi(float eps)
{
    int    i,j,k,m,mcycle;
    float  p,q,t,s2,c2,c,s,r;

    m=0;            /* initial set for transformation counter (total) */
```

```
63      while(1){
64          mcycle=0;         /* initial set for transformation counter (one cycle) */
65          for(i=0;i<n-1;i++)
66              for(j=i+1;j<n;j++)
67                  if( fabs(a[i][j])>=eps ){
68                      m++; mcycle++;      /* increment of transformation counters */
69                      p= (a[i][i]-a[j][j])/2;  q=a[i][j];
70                      if( fabs(p)<fabs(q) ){
71                          t=p/q;                  /* t=cot(2 theta) */
72                          s2= 1/sqrt(1+t*t);
73                       if(q<0)  s2= -s2;
74                         c2= t*s2;
75                      }else{
76                          t=q/p;                  /* t=tan(2 theta) */
77                          c2= 1/sqrt(1+t*t);
78                       if(p<0)  c2= -c2;
79                          s2= t*c2;
80                      }
81                      if(c2>0){
82                          c= sqrt( (1+c2)/2 );  s= s2/c/2;
83                      }else{
84                          s= sqrt( (1-c2)/2 );  c= s2/s/2;
85                      }
86                      r= (a[i][i]+a[j][j])/2;
87                      a[i][i]= r + p*c2 + q*s2;
88                      a[j][j]= r - p*c2 - q*s2;
89                      a[i][j]= a[j][i]=0;
90                      for(k=0;k<n;k++)
91                          if( (k!=i)&&(k!=j) ){
92                              p= a[i][k];  q=a[j][k];
93                              a[i][k]= a[k][i]=  p*c+q*s;
94                              a[j][k]= a[k][j]= -p*s+q*c;
95                          }
96                      for(k=0;k<n;k++){
97                          p= u[k][i];  q= u[k][j];
98                          u[k][i]= p*c+q*s;
99                          u[k][j]=-p*s+q*c;
100                     }
101                 }
102         if(mcycle==0) return(m);      /* return no. of transformations (total) */
103     }
104 }
```

【実行例】

教科書章末問題【２】の質点・ばね系の固有値問題をヤコビ法により解く。

(1) 運動方程式はつぎのように行列・ベクトル表示される。

$$M\ddot{\boldsymbol{s}} = -K\boldsymbol{s}$$

$$\boldsymbol{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}, \quad M = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}, \quad K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix}$$

(2) 系は固有振動数ωの調和運動をしているとすれば，解は 振幅を$\boldsymbol{x} = [x_1, x_2, x_2]^T$として

$$\boldsymbol{s} = \boldsymbol{x}e^{i\omega t}$$

と表され，これを運動方程式に代入すると，一般固有値問題

$$K\boldsymbol{x} = \omega^2 M\boldsymbol{x}$$

となるので，結局，標準固有値問題

$$A\boldsymbol{x} = \lambda\boldsymbol{x} \qquad （ここに，\ A = M^{-1}K,\ \lambda = \omega^2）$$

を得る。$m_1=m_2=m_3=m$, $k_1=k_2=k_3=k$ とすると，行列 A は

$$A = \frac{k}{m}\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

と表される。いま k/m=1 として，標準固有値問題を解く。

```
------------------------実行開始----------------------
Order of a Symmetric matrix: n ? 3
Input Lower triangle of matrix
? A( 1, j), j=1, 1 : 2
? A( 2, j), j=1, 2 : -1 2
? A( 3, j), j=1, 3 : 0 -1 1

Symmetric matrix: A
    2.000000e+00   -1.000000e+00    0.000000e+00
   -1.000000e+00    2.000000e+00   -1.000000e+00
    0.000000e+00   -1.000000e+00    1.000000e+00

Tolerance : eps ? 0.1
cumulative no. of transformations=5
Eigen-values:
    3.246979e+00    1.554825e+00    1.981960e-01
Eigen-vectors:
   -5.907426e-01   -7.404091e-01    3.206514e-01
    7.370955e-01   -3.335710e-01    5.877248e-01
   -3.281969e-01    5.835449e-01    7.428069e-01

Tolerance : eps ? 0.01
cumulative no. of transformations=6
Eigen-values:
    3.246979e+00    1.554958e+00    1.980622e-01
Eigen-vectors:
   -5.907426e-01   -7.371892e-01    3.279864e-01
    7.370955e-01   -3.277196e-01    5.910075e-01
   -3.281969e-01    5.908908e-01    7.369769e-01

Tolerance : eps ? 0.0001
```

```
cumulative no. of transformations=7
Eigen-values:
    3.246980e+00    1.554958e+00    1.980622e-01
Eigen-vectors:
   -5.910084e-01   -7.369762e-01    3.279864e-01
    7.369773e-01   -3.279853e-01    5.910075e-01
   -3.279838e-01    5.910090e-01    7.369769e-01

Tolerance : eps ? 1.0e-6
cumulative no. of transformations=8
Eigen-values:
    3.246980e+00    1.554958e+00    1.980622e-01
Eigen-vectors:
   -5.910090e-01   -7.369762e-01    3.279853e-01
    7.369761e-01   -3.279853e-01    5.910090e-01
   -3.279853e-01    5.910090e-01    7.369762e-01

Tolerance : eps ? ^Z (controlとZを同時に押す) ↲(Enter Keyを押す)
                                                    …… 入力終了コードの入力
------------------------おしまい-----------------------
```

参考文献

高倉葉子：数値計算の基礎―解法と誤差―, コロナ社 (2007)
森口繁一, 伊理正夫, 武市正人 編：Ｃによる算法痛論, 東京大学出版会 (2000)
Heath, Michael T.: Scientific Computing, An Introductory Survey, McGraw-Hill (2002)