「工学のための数値計算」 6章 章末問題 解答例

□ **1** いま，A は対称行列であり，

$$A\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i, \qquad A\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j, \qquad \lambda_i \neq \lambda_j$$

と仮定する．このとき，

$$(A\boldsymbol{v}_i, A\boldsymbol{v}_j) = (\lambda_i \boldsymbol{v}_i, \lambda_j \boldsymbol{v}_j) = \lambda_i \lambda_j (\boldsymbol{v}_i, \boldsymbol{v}_j)$$

であり，また

$$(A\boldsymbol{v}_i, A\boldsymbol{v}_j) = (\boldsymbol{v}_i, A^T A\boldsymbol{v}_j) = (\boldsymbol{v}_i, AA\boldsymbol{v}_j) = \lambda_j^2 (\boldsymbol{v}_i, \boldsymbol{v}_j)$$

であるから

$$\lambda_i (\boldsymbol{v}_i, \boldsymbol{v}_j) = \lambda_j (\boldsymbol{v}_i, \boldsymbol{v}_j)$$

となり，仮定より

$$(\boldsymbol{v}_i, \boldsymbol{v}_j) = 0$$

が言える．

□ **2** いま，A を対称行列，$\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ をそれぞれ A の $\lambda_1, \lambda_2, \ldots, \lambda_n$ に対応する固有ベクトルとし，かつ，

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|, \qquad (\boldsymbol{v}_i, \boldsymbol{v}_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \qquad (i, j = 1, 2, \ldots, n)$$

とする．このとき，任意のベクトル $\boldsymbol{x}$ は

$$\boldsymbol{x} = c_1 \boldsymbol{v}_1 + c_2 \boldsymbol{v}_2 + \cdots + c_n \boldsymbol{v}_n \qquad (c_i = (\boldsymbol{x}, \boldsymbol{v}_i),\ i = 1, 2, \ldots, n)$$

と展開でき，$(\boldsymbol{x}, A\boldsymbol{x})$ は $\boldsymbol{v}_i$ の正規直交性から

$$\begin{aligned} (\boldsymbol{x}, A\boldsymbol{x}) &= (c_1 \boldsymbol{v}_1 + c_2 \boldsymbol{v}_2 + \cdots + c_n \boldsymbol{v}_n, \lambda_1 c_1 \boldsymbol{v}_1 + \lambda_2 c_2 \boldsymbol{v}_2 + \cdots + \lambda_n c_n \boldsymbol{v}_n) \\ &= \lambda_1 c_1^2 + \lambda_2 c_2^2 + \cdots + \lambda_n c_n^2 \end{aligned}$$

と書ける．そして，固有値の仮定を利用すると

$$\begin{aligned} |(\boldsymbol{x}, A\boldsymbol{x})| &= |\lambda_1 c_1^2 + \lambda_2 c_2^2 + \cdots + \lambda_n c_n^2| \\ &\leq |\lambda_1|\, |c_1^2| + |\lambda_2|\, |c_2^2| + \cdots + |\lambda_n|\, |c_n^2| \\ &\leq |\lambda_1|\, (c_1^2 + c_2^2 + \cdots + c_n^2) \end{aligned}$$

が成り立ち，$\|\boldsymbol{x}\|_2 = 1$ を仮定すれば $c_1^2 + c_2^2 + \cdots + c_n^2 = 1$ だから，

$$|(\boldsymbol{x}, A\boldsymbol{x})| \leq |\lambda_1|$$

となる．$\boldsymbol{x} = \boldsymbol{v}_1$ のときは上式の等号が成り立つことから，$\boldsymbol{v}_1$ が $f(\boldsymbol{x}) = |(\boldsymbol{x}, A\boldsymbol{x})|$ を最大とするベクトルであることがわかる．

□ **3** 行列

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

に対して，$\boldsymbol{x}^{(0)} = [1\ 0\ 0]^T (j_0 = 1)$ を初期値として，p.103 の「べき乗法の基本アルゴリズム」を適用すると，

$$\boldsymbol{x}^{(1)} = A\boldsymbol{x}^{(0)} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad r^{(1)} = \frac{x_{j_0}^{(1)}}{x_{j_0}^{(0)}} = 0, \quad j_1 = 2, \quad \boldsymbol{x}^{(1)} \leftarrow \boldsymbol{x}^{(1)}/x_{j_1}^{(1)} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{x}^{(2)} = A\boldsymbol{x}^{(1)} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \quad r^{(2)} = \frac{x_{j_1}^{(2)}}{x_{j_1}^{(1)}} = 1, \quad j_2 = 1, \quad \boldsymbol{x}^{(2)} \leftarrow \boldsymbol{x}^{(2)}/x_{j_2}^{(2)} = \begin{bmatrix} 1 \\ 1/2 \\ 1/2 \end{bmatrix}, \quad |r^{(2)} - r^{(1)}| = 1$$

$$\boldsymbol{x}^{(3)} = A\boldsymbol{x}^{(2)} = \begin{bmatrix} 1 \\ 3/2 \\ 3/2 \end{bmatrix}, \quad r^{(3)} = \frac{x_{j_2}^{(3)}}{x_{j_2}^{(2)}} = 1, \quad j_3 = 2, \quad \boldsymbol{x}^{(3)} \leftarrow \boldsymbol{x}^{(3)}/x_{j_3}^{(3)} = \begin{bmatrix} 2/3 \\ 1 \\ 1 \end{bmatrix}, \quad |r^{(3)} - r^{(2)}| = 0$$

となり，3 回の反復で終了してしまう．もちろん，$r^{(3)} = 1$ は A の固有値ではない．

□ **4** 式 (6.33) の $Q_{i,j,\theta}$ を

$$Q_{i,j,\theta} = \left[\begin{array}{ccc|ccc|ccc} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ \hline & & & \cos\theta & & & \sin\theta & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & -\sin\theta & & & \cos\theta & & \\ \hline & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{array}\right] = \begin{bmatrix} I & & \\ & \hat{Q} & \\ & & I \end{bmatrix}$$

とおけば

$$Q_{i,j,\theta}^T Q_{i,j,\theta} = \begin{bmatrix} I & & \\ & \hat{Q}^T\hat{Q} & \\ & & I \end{bmatrix}$$

となり，$Q_{i,j,\theta}$ が直交行列であることを示すには，$\hat{Q}$ が直交行列であることを示せばよいことがわかる．ただし，I はそのサイズに応じた単位行列を表すものとする．さらに，$\hat{Q}$ を

$$\hat{Q} = \left[\begin{array}{c|ccc|c} \cos\theta & & & & \sin\theta \\ \hline & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ \hline -\sin\theta & & & & \cos\theta \end{array}\right] = \begin{bmatrix} \cos\theta & & \sin\theta \\ & I & \\ -\sin\theta & & \cos\theta \end{bmatrix}$$

と分割すれば

$$\begin{aligned}
\hat{Q}^T\hat{Q} &= \begin{bmatrix} \cos\theta & & -\sin\theta \\ & I & \\ \sin\theta & & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & & \sin\theta \\ & I & \\ -\sin\theta & & \cos\theta \end{bmatrix} \\
&= \begin{bmatrix} \cos^2\theta+\sin^2\theta & & \cos\theta\sin\theta-\sin\theta\cos\theta \\ & I & \\ \sin\theta\cos\theta-\cos\theta\sin\theta & & \sin^2\theta+\cos^2\theta \end{bmatrix} \\
&= I
\end{aligned}$$

となり，$\hat{Q}$ が直交行列であることが示された．

□ **5** 問題４の $Q_{i,j,\theta}$ の分割を利用し，A も $Q_{i,j,\theta}$ と同様に

$$A = \begin{bmatrix} A_{11} & A_{21}^T & A_{31}^T \\ A_{21} & A_{22} & A_{32}^T \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

と分割すれば，

$$\begin{aligned}
B &= Q_{i,j,\theta}^T A Q_{i,j,\theta} \\
&= \begin{bmatrix} I & & \\ & \hat{Q}^T & \\ & & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{21}^T & A_{31}^T \\ A_{21} & A_{22} & A_{32}^T \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} I & & \\ & \hat{Q} & \\ & & I \end{bmatrix} = \begin{bmatrix} A_{11} & A_{21}^T\hat{Q} & A_{31}^T \\ \hat{Q}^T A_{21} & \hat{Q}^T A_{22}\hat{Q} & \hat{Q}^T A_{32}^T \\ A_{31} & A_{32}\hat{Q} & A_{33} \end{bmatrix}
\end{aligned}$$

となり，b_{ij} を求めるのであれば，$\hat{Q}^T A_{22}\hat{Q}$ のみに注目すればよい．さらに，$\hat{Q}$ も問題４と同様の分割を行い，同じく A_{22} も

$$A_{22} = \begin{bmatrix} a_{ii} & \hat{A}_{21}^T & a_{ij} \\ \hat{A}_{21} & \hat{A}_{22} & \hat{A}_{32}^T \\ a_{ij} & \hat{A}_{32} & a_{jj} \end{bmatrix}$$

と分割し，簡単のために $c = \cos\theta, s = \sin\theta$ とおけば，

$$\begin{aligned}
\hat{Q}^T A_{22}\hat{Q} &= \begin{bmatrix} c & & -s \\ & I & \\ s & & c \end{bmatrix} \begin{bmatrix} a_{ii} & \hat{A}_{21}^T & a_{ij} \\ \hat{A}_{21} & \hat{A}_{22} & \hat{A}_{32}^T \\ a_{ij} & \hat{A}_{32} & a_{jj} \end{bmatrix} \begin{bmatrix} c & & s \\ & I & \\ -s & & c \end{bmatrix} \\
&= \begin{bmatrix} ca_{ii}-sa_{ij} & c\hat{A}_{21}^T - s\hat{A}_{32} & ca_{ij}-sa_{jj} \\ \hat{A}_{21} & \hat{A}_{22} & \hat{A}_{32}^T \\ sa_{ii}+ca_{ij} & s\hat{A}_{21}^T + c\hat{A}_{32} & sa_{ij}+ca_{jj} \end{bmatrix} \begin{bmatrix} c & & s \\ & I & \\ -s & & c \end{bmatrix} \\
&= \begin{bmatrix} (ca_{ii}-sa_{ij})c-(ca_{ij}-sa_{jj})s & c\hat{A}_{21}^T - s\hat{A}_{32} & (ca_{ii}-sa_{ij})s+(ca_{ij}-sa_{jj})c \\ \hat{A}_{21}c-\hat{A}_{32}^T s & \hat{A}_{22} & \hat{A}_{21}s+\hat{A}_{32}^T c \\ (sa_{ii}+ca_{ij})c-(sa_{ij}+ca_{jj})s & s\hat{A}_{21}^T + c\hat{A}_{32} & (sa_{ii}+ca_{ij})s+(sa_{ij}+ca_{jj})c \end{bmatrix}
\end{aligned}$$

となる．すなわち，

$$\begin{aligned}
b_{ij} &= a_{ii}\sin\theta\cos\theta + a_{ij}\cos^2\theta - a_{ij}\sin^2\theta - a_{jj}\sin\theta\cos\theta \\
&= (a_{ii}-a_{jj})\sin\theta\cos\theta + a_{ij}(\cos^2\theta-\sin^2\theta) \\
&= \frac{1}{2}(a_{ii}-a_{jj})\sin 2\theta + a_{ij}\cos 2\theta
\end{aligned}$$

となる．

□ **6** C 言語でのヤコビ法のプログラム例を以下に示す．なお，このプログラムでは行列の格納に，２次元配列ではなく，１次元配列を用い，その１次元配列も動的に領域を確保するようになっている．すなわち，$n \times n$ 行列 A は

```
double *a;
a = (double *) malloc((size_t) (n*n * sizeof(double)));
```

のように領域を確保し，ij 要素へのアクセスは

```
*(a + i + n*j)
```

のように行う．また，配列はすべて 0 番目から始まることに注意されたい．

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define N 3
#define EPS 1.0e-5

int main(void)
{
  /* 関数のプロトタイプ宣言 */
  int jacobi(double *, int, double *, double);

  double *a, *eig;
  int it, i, j;

  /* 行列を２次元配列ではなく，１次元配列として扱う
   * ij 成分へのアクセスは *(a + i + N*j) とする */
  a = (double *) malloc((size_t) (N*N * sizeof(double)));

  /* 固有値を格納するための１次元配列 */
  eig = (double *) malloc((size_t) (N * sizeof(double)));

  /* 行列の要素の格納
   * 行列は対称行列であるため，本来ならば対角成分より下か，
   * もしくは上のみを格納すれば十分であるが，
   * ここでは総ての要素を格納する */
  *(a + 0 + N * 0) = 1.0;  /* A の 0,0 要素 */
  *(a + 0 + N * 1) = 2.0;  /* A の 0,1 要素 */
  *(a + 0 + N * 2) = 3.0;  /* A の 0,2 要素 */
  *(a + 1 + N * 0) = 2.0;  /* A の 1,0 要素 */
  *(a + 1 + N * 1) = 4.0;  /* A の 1,1 要素 */
  *(a + 1 + N * 2) = 5.0;  /* A の 1,2 要素 */
  *(a + 2 + N * 0) = 3.0;  /* A の 2,0 要素 */
  *(a + 2 + N * 1) = 5.0;  /* A の 2,1 要素 */
  *(a + 2 + N * 2) = 6.0;  /* A の 2,2 要素 */
```

```
  /* 固有値，固有ベクトルを求めるべき行列の表示 */
  for(i = 0; i < N; i++) {
    for(j = 0; j < N; j++) {
      printf("%f   ", *(a + i + N*j));
    }
    printf("\n");
  }

  /* ヤコビ法による固有値問題の解法 */
  it = jacobi(a, N, eig, EPS);

  /* 収束に要した反復回数の表示 */
  printf("\nIteration = %d\n", it);

  /* 固有値の表示 */
  for(i = 0; i < N; i++) {
    printf("eig(%d) = %f\n", i, *(eig+i));
  }

  /* 固有ベクトルの表示 */
  printf("\nEigenvectors = \n");
  for(i = 0; i < N; i++) {
    for(j = 0; j < N; j++) {
      printf("%f   ", *(a + i + N*j));
    }
    printf("\n");
  }

  /* 配列の領域の解放 */
  free(a);
  free(eig);

  return 0;
}

int jacobi(double *a, int n, double *eig, double eps)
/* ヤコビ法による行列の固有値問題の解法
 * ------
 * double *a   .. 入出力  サイズ   n × n
 *                行列を入力，固有ベクトルが出力される
 * int n       .. 入力
 *                行列の次元
 * double *eig .. 出力    サイズ   n
 *                行列の固有値を出力
```

```
 * double eps  .. 入力
 *                収束判定のための定数
 * 戻り値は，収束に要した反復回数である．
 * なお，このプログラムは，行列の対角を含む右上部分のみに
 * アクセスし，左下部分は用いない
 * ------  */
{
  /* 補助関数のプロトタイプ宣言 */
  int ijtheta(double *, int, int *, int *,
      double *, double *, double);
  /* ローカル変数の宣言 */
  double *v;
  double c, s, x, y, ss, cc, cs;
  int i, j, imax, jmax, itnum = 0;

  /* 固有ベクトルを一時的に保管するための配列の領域を確保する */
  v = (double *) malloc((size_t) (n*n * sizeof(double)));

  /* v を単位行列に初期化 */
  for(i = 0; i < n; i++) {
    for(j = 0; j < n; j++) {
      if(i == j) {
        *(v + i + n*j) = 1.0;
      } else {
        *(v + i + n*j) = 0.0;
      }
    }
  }

  while( /* 収束判定と絶対値最大要素の行，列番号を探す
  * 補助関数を呼び出す．もし，絶対値最大要素の値が
  * eps 以下ならば関数 ijtheta は -1 を，そうでなければ
  * 0 を返す．*/
         ijtheta(a, n, &imax, &jmax, &c, &s, eps) == 0
       ) {

    /* 対角成分の計算 */
    x = *(a + imax + n*imax);
    y = *(a + jmax + n*jmax);
    ss = s*s;
    cc = c*c;
    cs = 2.0*c*s;
    *(a + imax + n*imax) = cc*x - cs*(*(a + imax + n*jmax)) + ss*y;
    *(a + jmax + n*jmax) = ss*x + cs*(*(a + imax + n*jmax)) + cc*y;
```

```
  /* 0 行から imax-1 行までの計算 */
  for(i = 0; i < imax; i++) {
    x = *(a + i + n*imax);
    y = *(a + i + n*jmax);
    *(a + i + n*imax) = c*x - s*y;
    *(a + i + n*jmax) = s*x + c*y;
  }
  /* jmax+1 列から n-1 列までの計算 */
  for(j = jmax+1; j < n; j++) {
    x = *(a + imax + n*j);
    y = *(a + jmax + n*j);
    *(a + imax + n*j) = c*x - s*y;
    *(a + jmax + n*j) = s*x + c*y;
  }
  /* imax 行の imax+1 ～ jmax-1 番目の要素と
   * jmax 列の imax+1 ～ jmax-1 番目の要素の計算 */
  for(i = imax+1; i < jmax; i++) {
    x = *(a + imax + n*i);
    y = *(a + i + n*jmax);
    *(a + imax + n*i) = c*x - s*y;
    *(a + i + n*jmax) = s*x + c*y;
  }
  /* a の imax,jmax 要素をゼロにする */
  *(a + imax + n*jmax) = 0.0;

  /* 固有ベクトルの計算 */
  for(i = 0; i < n; i++) {
    x = *(v + i + n*imax);
    y = *(v + i + n*jmax);
    *(v + i + n*imax) = c*x - s*y;
    *(v + i + n*jmax) = s*x + c*y;
  }

  /* 反復回数のカウンタのインクリメント */
  itnum++;
}

/* a の対角成分を eig に移す */
for(i = 0; i < n; i++) {
  *(eig + i) = *(a + i + n*i);
}
```

```
  /* 固有ベクトルを a に上書きする */
  for(i = 0; i < n; i++) {
    for(j = 0; j < n; j++) {
      *(a + i + n*j) = *(v + i + n*j);
    }
  }

  /* v の領域の解放 */
  free(v);

  return itnum;
}

int ijtheta(double *a, int n,
    int *imax, int *jmax,
    double *c, double *s, double eps)
/* ヤコビ法のための補助関数
 * 行列 A の非対角成分のうちで，絶対値が最大となる要素の
 * 行番号と列番号をそれぞれ imax と jmax に格納して返す.
 * そして，ヤコビ法の直交行列の構成に必要な cos theta と
 * sin theta の値も計算し，c と s に格納して返す.
 * もし，絶対値最大の要素の値が eps 未満ならば戻り値として -1 を
 * そうでなければ 0 を返す. */
{
  double amax, t2, c2, s2, z;
  int i, j;

  /* 非対角成分の中で，絶対値が最大の要素を探し，
   * その行番号を imax に，列番号を jmax に格納する */
  for(amax = 0.0, j = 0; j < n; j++) {
    for(i = 0; i < j; i++) {
      if(fabs(*(a + i + n*j)) > amax) {
        *imax = i;
        *jmax = j;
        amax = fabs(*(a + i + n*j));
      }
    }
  }

  /* もし，絶対値最大要素の値が eps 未満であるならば
   * -1 を戻り値として終了 */
  if(amax < eps) {
    return -1;
  } else {
```

```
    /* 絶対値最大要素の値が eps 以上であれば,
     * 直交行列に必要な cos theta と sin theta の値を計算する */
    t2 = - 2.0*(*(a + (*imax) + n*(*jmax))) /
      ((*(a + (*imax) + n*(*imax)))
       - (*(a + (*jmax) + n*(*jmax))));  /* p.115 式 (6.37) */
    z = sqrt(1.0 + t2*t2);
    c2 = 1.0 / z;
    s2 = t2  / z;
    *c = sqrt((1.0 + c2) * 0.5);
    *s = s2 / (2.0 * (*c));                /* p.115 式 (6.38) */
    /* 戻り値は 0 とする */
    return 0;
  }
}
```

プログラム実行例

```
1.000000   2.000000   3.000000
2.000000   4.000000   5.000000
3.000000   5.000000   6.000000

Iteration = 6
eig(0) = -0.515729
eig(1) = 0.170915
eig(2) = 11.344814

Eigenvectors =
0.736976   -0.591009   0.327985
0.327985   0.736976   0.591009
-0.591009   -0.327985   0.736976
```