固有値解析

中島 研吾
東京大学情報基盤センター
同　大学院情報理工学系研究科数理情報学専攻
数値解析（科目番号 03-500081）

- 行列の固有値問題
- べき乗法
- 対称行列の固有値計算法:ヤコビ法

行列の固有値問題

標準固有値問題（Standard Eigenvalue Problem

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}, \quad \mathbf{x} \neq \mathbf{0}$$

を満足する λ と $\mathbf{x}$ を求める

- λ : 固有値（eigenvalue）
- $\mathbf{x}$: 固有ベクトル（eigenvector）

一般固有値問題（General Eigenvalue Problem）

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{M}\mathbf{x}, \quad \mathbf{x} \neq \mathbf{0}$$

ここでは標準固有値問題を扱う

固有値

- **固有振動数**
- **行列の性質に影響：スペクトル半径，条件数**

固有値問題の計算(1/3)

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$ の固有値・固有ベクトルを求めよ.

$$\mathbf{Ax} = \lambda\mathbf{x}, \quad \mathbf{x} \neq \mathbf{0} \rightarrow [\mathbf{A} - \lambda\mathbf{I}]\cdot\mathbf{x} = \mathbf{0}, \quad \mathbf{x} \neq \mathbf{0}$$

特性方程式 $\therefore \quad \det(\mathbf{A} - \lambda\mathbf{I}) = 0$

特性方程式=0

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \begin{vmatrix} 1-\lambda & -1 \\ -1 & 2-\lambda \end{vmatrix} = \lambda^2 - 3\lambda + 1 = 0$$

$$\lambda = \frac{3 \pm \sqrt{5}}{2} \longrightarrow \lambda_1 = \frac{3+\sqrt{5}}{2}, \lambda_2 = \frac{3-\sqrt{5}}{2}$$

固有値問題の計算(2/3)

$\mathbf{Ax} = \lambda\mathbf{x}$ より

$$x_1 - x_2 = \lambda x_1$$
$$-x_1 + 2x_2 = \lambda x_2$$

→ この連立方程式は、必ず不定

したがって, x_1, x_2のどちらか一方を定数をおく.

たとえば $x_1 = c_1$とおけば $x_2 = (1-\lambda)c_1$

固有ベクトル:

$$\lambda = \lambda_1 \rightarrow \mathbf{x} = c_1 \begin{pmatrix} 1 \\ \dfrac{-1-\sqrt{5}}{2} \end{pmatrix} = c_1 \begin{pmatrix} 1 \\ -2.7 \end{pmatrix}$$

$$\lambda = \lambda_2 \rightarrow \mathbf{x} = c_1 \begin{pmatrix} 1 \\ \dfrac{-1+\sqrt{5}}{2} \end{pmatrix} = c_1 \begin{pmatrix} 1 \\ 0.62 \end{pmatrix}$$

固有値問題の計算例(3/3)

一般のn元の正方行列Aの固有値，固有ベクトルは，前述したような方法で求めることができる

特性方程式は固有値λについてのn次の代数方程式(非線形)

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0$$

大規模な次元(>10^6)を有する行列の固有値問題も扱える方法が開発されている:実に様々な解法がある

実用上重要なのは(絶対値)最大・最小固有値
重根があると特別な扱い必要
- 本講義では基本的に重根は無しとする

標準固有値問題の解法

- べき乗法，逆べき乗法

- 小規模問題：ヤコビ法
- 中規模問題：ハウスホルダー法（対称），QR法
- 大規模問題：逆反復法，同時反復法（べき乗法の拡張）

- 行列の固有値問題
- べき乗法
- 対称行列の固有値計算法:ヤコビ法

べき乗法（Power Method）

絶対値最大の実固有値と
それに対応する固有ベクトルを求める方法

適当な初期ベクトル $\mathbf{x}^{(0)}$ から始めて

$$\mathbf{x}^{(1)} = \mathbf{A}\mathbf{x}^{(0)}$$
$$\mathbf{x}^{(2)} = \mathbf{A}\mathbf{x}^{(1)}$$
$$\vdots$$
$$\mathbf{x}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)}$$

Aをどんどん乗じていく
但し，単に乗じていくだけでは、
発散したり，原点に収束したり
してしまうので，常に $\mathbf{x}^{(k)}$の大きさを
一定（例えば=1）に保つ必要がある.

$\mathbf{x}^{(k)}$は絶対値最大の固有値に対応する固有ベクトルに収束していく

べき乗法のアルゴリズム

- **<u>Step 0</u>**: $\|\mathbf{x}^{(0)}\|_2=1$ である初期ベクトル $\mathbf{x}^{(0)}$ を選び, $k=0$ とする
- **<u>Step 1</u>**: 以下のように $\mathbf{x}^{(k+1)}$ を更新する:

$$\mathbf{y}^{(k)} = \mathbf{A}\mathbf{x}^{(k)},\ \lambda = \left(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}\right) \Rightarrow \mathbf{x}^{(k+1)} = \frac{\mathbf{y}^{(k)}}{\left\|\mathbf{y}^{(k)}\right\|_2}$$

- **<u>Step 2:</u>** $k=k+1$としてStep 1を繰り返す

λ	: $\mathbf{A}$ の絶対値最大の実固有値に収束
$\mathbf{x}^{(k)}$	: $\mathbf{A}$ の絶対値最大の実固有値に対応する固有ベクトルに収束

べき乗法が最大固有値に収束する理由(1/3)

$$\mathbf{y}^{(0)} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + c_3\mathbf{x}_3 + \cdots + c_n\mathbf{x}_n$$

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$$ 固有値(絶対値の大きさ順)

$$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_n$$ それに対応する固有ベクトル(一次独立と仮定)

$$\mathbf{x}^{(1)} = \mathbf{A}\mathbf{y}^{(0)} = \lambda_1 c_1\mathbf{x}_1 + \lambda_2 c_2\mathbf{x}_2 + \lambda_3 c_3\mathbf{x}_3 + \cdots + \lambda_n c_n\mathbf{x}_n$$

$$\mathbf{x}^{(k)} = \mathbf{A}\mathbf{y}^{(k-1)} = \mathbf{A}^k\mathbf{y}^{(0)} = \lambda_1{}^k c_1\mathbf{x}_1 + \lambda_2{}^k c_2\mathbf{x}_2 + \lambda_3{}^k c_3\mathbf{x}_3 + \cdots + \lambda_n{}^k c_n\mathbf{x}_n$$

べき乗法が最大固有値に収束する理由(2/3)

$$if \quad c_1 \neq 0$$

$$\mathbf{y}^{(k)} = \lambda_1{}^k c_1 \cdot \left\{ \mathbf{x}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k \frac{c_2}{c_1}\mathbf{x}_2 + \left(\frac{\lambda_3}{\lambda_1}\right)^k \frac{c_3}{c_1}\mathbf{x}_3 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k \frac{c_n}{c_1}\mathbf{x}_n \right\}$$

$$\left|\frac{\lambda_i}{\lambda_1}\right| < 1 \left(i = 2,3,\cdots,n\right) \Rightarrow \lim_{k\to\infty}\left(\frac{\lambda_i}{\lambda_1}\right)^k = 0$$

$$\therefore \quad if \ k \to \infty : \mathbf{y}^{(k)} = \lambda_1{}^k c_1 \mathbf{x}_1$$

べき乗法によって求められるベクトル$\mathbf{x}^{(k)}$の「方向」が最大固有値 λ_1に対応する固有ベクトル $\mathbf{x}_1$ のそれに収束していく

べき乗法が最大固有値に収束する理由(3/3)

$$\mathbf{y}^{(k-1)} = \lambda_1^{\,k-1} c_1 \mathbf{x}_1 \Rightarrow \mathbf{x}^{(k)} = \frac{\mathbf{y}^{(k-1)}}{\left\|\mathbf{y}^{(k-1)}\right\|_2} = \lambda_1^{\,k-1} c_1 \mathbf{x}_1 \cdot \frac{1}{\left\|\mathbf{y}^{(k-1)}\right\|_2}$$

$$\mathbf{y}^{(k)} = \mathbf{A} \cdot \mathbf{x}^{(k)} = \lambda_1^{\,k} c_1 \mathbf{x}_1 \cdot \frac{1}{\left\|\mathbf{y}^{(k-1)}\right\|_2}$$

$$\frac{\left(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}\right)}{\left(\mathbf{x}^{(k)}, \mathbf{x}^{(k)}\right)} = \frac{\left(\left(\lambda_1^{\,k-1} c_1 \mathbf{x}_1 \cdot \frac{1}{\left\|\mathbf{y}^{(k-1)}\right\|_2}\right), \left(\lambda_1^{\,k} c_1 \mathbf{x}_1 \cdot \frac{1}{\left\|\mathbf{y}^{(k-1)}\right\|_2}\right)\right)}{\left(\left(\lambda_1^{\,k-1} c_1 \mathbf{x}_1 \cdot \frac{1}{\left\|\mathbf{y}^{(k-1)}\right\|_2}\right), \left(\lambda_1^{\,k-1} c_1 \mathbf{x}_1 \cdot \frac{1}{\left\|\mathbf{y}^{(k-1)}\right\|_2}\right)\right)} = \lambda_1$$

$$\frac{\left(\mathbf{x}^{(k)}, \mathbf{y}^{(k)},\right)}{\left(\mathbf{x}^{(k)}, \mathbf{x}^{(k)}\right)} = \left(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}\right) = \lambda_1 \quad \because \left(\mathbf{x}^{(k)}, \mathbf{x}^{(k)}\right) = 1$$

べき乗法の収束

$$\left|\frac{\lambda_i}{\lambda_1}\right| < 1 \left(i = 2,3,\cdots,n\right) \Rightarrow \lim_{k\to\infty}\left(\frac{\lambda_i}{\lambda_1}\right)^k = 0$$

$|\lambda_i/\lambda_1|$が1より充分小さいことが収束に影響，特に以下の成立が高速な収束に必要

$$\left|\frac{\lambda_2}{\lambda_1}\right| << 1$$

べき乗法の例(1/3)

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$

の絶対値最大の固有値およびその固有ベクトルをべき乗法により求めよ.

1回目

$$\mathbf{x}^{(0)} = \{1,0\}, \quad \mathbf{y}^{(0)} = \mathbf{A}\mathbf{x}^{(0)} = \{1,-1\}$$

$$\mathbf{x}^{(1)} = \frac{1}{\left\|\mathbf{y}^{(0)}\right\|_2}\{1,-1\} = \frac{1}{\sqrt{2}}\{1,-1\}$$

$$\left(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}\right) = \left(\{1 \quad 0\}, \{1,-1\}\right) = 1$$

べき乗法の例(2/3)

2回目

$$\mathbf{x}^{(1)} = \frac{1}{\sqrt{2}}\{1,-1\}, \quad \mathbf{y}^{(1)} = \mathbf{A}\mathbf{x}^{(1)} = \frac{1}{\sqrt{2}}\{2,-3\}$$

$$\mathbf{x}^{(2)} = \frac{1}{\left\|\mathbf{y}^{(1)}\right\|_2}\frac{1}{\sqrt{2}}\{2,-3\} = \frac{\sqrt{2}}{\sqrt{13}}\frac{1}{\sqrt{2}}\{2,-3\} = \frac{1}{\sqrt{13}}\{2,-3\}$$

$$\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}\right) = \left(\frac{1}{\sqrt{2}}\{1 \quad -1\}, \frac{1}{\sqrt{2}}\{2,-3\}\right) = \frac{5}{2} = 2.500$$

べき乗法の例(3/3)

3回目

$$\mathbf{x}^{(2)} = \frac{1}{\sqrt{13}}\{2,-3\}, \quad \mathbf{y}^{(2)} = \mathbf{A}\mathbf{x}^{(2)} = \frac{1}{\sqrt{13}}\{5,-8\}$$

$$\mathbf{x}^{(3)} = \frac{1}{\left\|\mathbf{y}^{(2)}\right\|_2}\frac{1}{\sqrt{13}}\{5,-8\} = \frac{\sqrt{13}}{\sqrt{89}}\frac{1}{\sqrt{13}}\{5,-8\} = \frac{1}{\sqrt{89}}\{5,-8\}$$

$$\left(\mathbf{x}^{(2)}, \mathbf{y}^{(2)}\right) = \left(\frac{1}{\sqrt{13}}\{2 \quad -3\}, \frac{1}{\sqrt{13}}\{5,-8\}\right) = \frac{34}{13} = 2.6153$$

前述した厳密解

$$\lambda_1 = \frac{3+\sqrt{5}}{2} = 2.618034$$

逆べき乗法（Inverse Power Method）

絶対値「最小」の実固有値と
それに対応する固有ベクトルを求める方法

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \Rightarrow \mathbf{A}^{-1}\mathbf{x} = \lambda^{-1}\mathbf{x}$$

$\mathbf{A}' = \mathbf{A}^{-1}, \lambda' = \lambda^{-1}$ として $\mathbf{A}'\mathbf{x} = \lambda'\mathbf{x}$ にべき乗法を適用する

$\mathbf{LU} = \mathbf{A}$ としてLU分解を求めておくと効率が良い

LU分解による前進後退代入：$\mathbf{A}^{-1}$を乗じるのと同じこと

べき乗法の加速手法：原点移動（Shift）

$|\lambda_2/\lambda_1|$の値を小さくすることにより収束を加速する

$$\mathbf{Ax} = \lambda\mathbf{x}, \quad \mathbf{B} = \mathbf{A} - p\mathbf{I} \ \textit{where}\ p : \text{constant}$$

$$\mathbf{Ax} = (\mathbf{B} + p\mathbf{I})\mathbf{x} = \lambda\mathbf{x}$$

$$\mathbf{Bx} = \lambda\mathbf{x} - p\mathbf{Ix} = (\lambda - p)\mathbf{x}$$

$(\lambda - p)$：行列**B**の固有値（λ：行列**A**の固有値）

$\mathbf{x}$：　行列**B**の固有ベクトル（**A**の固有ベクトルに一致）

適当な定数pを選択することにより行列**B**の絶対値最大／2番目に大きな固有値の比を小さくできれば，行列**B**にべき乗法を適用した方が良い

$$\frac{|\lambda_2 - p|}{|\lambda_1 - p|} < \frac{|\lambda_2|}{|\lambda_1|}$$

行列**B**の固有値　行列**A**

Scilabによるプログラム（1/2）

http://nkl.cc.u-tokyo.ac.jp/16n/

数値解析(科目番号 03-500081)(平成28年度秋学期)(中島担当分)

講義資料(講義に出なくて済むためのものではありません)

- スーパーコンピューティングへの招待，プログラミングについて[資料]
- 線形方程式の解法(直接法)[資料][Scilabプログラム]
- 線形方程式の解法(反復法)[資料][Scilabプログラム]
  - (参考:前処理手法)[資料]
- 固有値解析[資料] Scilabプログラム]
- 偏微分方程式の数値解法[資料][Scilabプログラム]
  - (「修正方程式(Modified Equation)の説明」)[資料]

クリックして Eigen.zip をダウンロード

期末試験:1月30日(月)XX:XX-XX:XX A4両面の自作メモ持ち込み可

Scilabによるプログラム（2/2）

http://nkl.cc.u-tokyo.ac.jp/15n/

- Eigen.zipを解凍⇒直下に Eigen というディレクトリ
  - Scilabプログラムのソースファイル（*.sce, *.sci）
  - README.txt（下記：必ず読むこと）

eigen-p0.sce: Power Method 2x2 with/without shift
eigen-p2.sce: Power Method 6x6 with/without shift
eigen-p3.sce: Inverse Power Method 6x6 with/without shift

lu.sci : LU factorization LU分解
fbs.sci: Forward/Backward Substitution 前進後退代入

プログラム名 内容

原点移動の効果：eigen-p0.sce

下記の条件においてAの絶対値最大の固有値およびその固有ベクトルをべき乗法，原点移動付きべき乗法により求めよ.

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}, \quad \mathbf{x}^{(0)} = \{1,0\}, \quad p = 0.40$$

	原点移動無し	原点移動有り
1	1.000000E+00	1.000000E+00
2	2.500000E+00	2.617647E+00
3	2.615385E+00	2.618034E+00
4	2.617978E+00	
5	2.618033E+00	
6	2.618034E+00	

SHIFT ?
--> 0.40

べき乗法の例(シフト無し) eigen-p0.sce

```
n=3;

A=[1,-1;-1,2];
X=[1;0];

printf("Shift? ?¥n"); Shift= scanf("%f");
printf("¥n");
printf("Shift %e¥n", Shift);
printf("¥n");

printf("#Original ¥n");

for iter= 1:10
  Y(1)= A(1,1)*X(1) + A(1,2)*X(2);
  Y(2)= A(2,1)*X(1) + A(2,2)*X(2);
  Eigen= X(1)*Y(1) + X(2)*Y(2);
  DL= sqrt(Y(1)*Y(1)+Y(2)*Y(2));
  X(1)= Y(1)/DL;
  X(2)= Y(2)/DL;
  printf ("Iters %d %e %e %e ¥n",iter, Eigen, X(1), X(2));
end
```

$$\mathbf{y}^{(k)} = \mathbf{A}\mathbf{x}^{(k)}$$

べき乗法の例(シフト無し) eigen-p0.sce

```
n=3;

A=[1,-1;-1,2];
X=[1;0];

printf("Shift? ?¥n"); Shift= scanf("%f");
printf("¥n");
printf("Shift %e¥n", Shift);
printf("¥n");

printf("#Original ¥n");

for iter= 1:10
  Y(1)= A(1,1)*X(1) + A(1,2)*X(2);
  Y(2)= A(2,1)*X(1) + A(2,2)*X(2);
  Eigen= X(1)*Y(1) + X(2)*Y(2);
  DL= sqrt(Y(1)*Y(1)+Y(2)*Y(2));
  X(1)= Y(1)/DL;
  X(2)= Y(2)/DL;
  printf ("Iters %d %e %e %e ¥n",iter, Eigen, X(1), X(2));
end
```

$$\lambda = \left(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}\right)$$

べき乗法の例(シフト無し) eigen-p0.sce

```
n=3;

A=[1,-1;-1,2];
X=[1;0];

printf("Shift? ?¥n"); Shift= scanf("%f");
printf("¥n");
printf("Shift %e¥n", Shift);
printf("¥n");

printf("#Original ¥n");

for iter= 1:10
  Y(1)= A(1,1)*X(1) + A(1,2)*X(2);
  Y(2)= A(2,1)*X(1) + A(2,2)*X(2);
  Eigen= X(1)*Y(1) + X(2)*Y(2);
  DL= sqrt(Y(1)*Y(1)+Y(2)*Y(2));
  X(1)= Y(1)/DL;
  X(2)= Y(2)/DL;
  printf ("Iters %d %e %e %e ¥n",iter, Eigen, X(1), X(2));
end
```

$$\mathbf{x}^{(k+1)} = \frac{\mathbf{y}^{(k)}}{\left\|\mathbf{y}^{(k)}\right\|_2}$$

べき乗法の例(シフト有り) eigen-p0.sce

```
X(1)= 1.0;
X(2)= 0.0;
A(1,1)= A(1,1) - Shift;
A(2,2)= A(2,2) - Shift;

for iter= 1:10
  Y(1)= A(1,1)*X(1) + A(1,2)*X(2);
  Y(2)= A(2,1)*X(1) + A(2,2)*X(2);
  EIGEN= X(1)*Y(1) + X(2)*Y(2) + Shift;
  DL= sqrt(Y(1)^2+Y(2)^2);
  X(1)= Y(1)/DL;
  X(2)= Y(2)/DL;
  printf ("Iters %d %e %e %e %e ¥n",iter, EIGEN, X(1), X(2), Shift);
end
```

$$\lambda = (\lambda - p) + p$$

$$\mathbf{Ax} = \lambda\mathbf{x}, \quad \mathbf{B} = \mathbf{A} - p\mathbf{I} \ \textit{where}\ p : \text{constant}$$

$$\mathbf{Ax} = (\mathbf{B} + p\mathbf{I})\mathbf{x} = \lambda\mathbf{x}$$

$$\mathbf{Bx} = \lambda\mathbf{x} - p\mathbf{Ix} = (\lambda - p)\mathbf{x}$$

$(\lambda - p)$: 行列**B**の固有値(λ:行列**A**の固有値)

$\mathbf{x}$: 行列**B**の固有ベクトル(**A**の固有ベクトルに一致)

計算例

$$A = \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 5 & 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 4 & 3 & 2 & 1 \\ 3 & 3 & 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

λ_1= 1.721E+01

{5.507E-01 5.187E-01 4.565E-01 3.678E-01 2.578E-01 1.327E-01}

λ_6= 2.652E-01

{1.327E-01 -3.678E-01 5.187E-01 -5.507E-01 4.565E-01 -2.578E-01}

べき乗法の例(シフト無し) eigen-p2.sce

```
n=6;
X=[1;0;0;0;0;0];
Y=[0;0;0;0;0;0];

EIG0= 1.0;
for iter= 1:100
  for i=1:n
    Y(i)=0.0;
  end
  for i=1:n
    for j=1:n
      Y(i)= Y(i) + A(i,j)*X(j);
    end
  end

  Eigen= 0.;
  DL   = 0.;
  for i= 1:n
    Eigen= Eigen + X(i)*Y(i);
    DL   = DL    + Y(i)*Y(i);
  end
```

```
  RDL= 1.0/sqrt(DL);
  for i= 1:n
    X(i)= Y(i)*RDL;
  end

  RESID= sqrt((Eigen-EIG0)*                ..
         (Eigen-EIG0)/(EIG0*EIG0));

  if RESID < EPS then
    break;
  end;
  EIG0= Eigen;
end
```

$$RESID = \frac{\left\| \lambda_1^{(k)} - \lambda_1^{(k-1)} \right\|_2}{\left\| \lambda_1^{(k-1)} \right\|_2}$$

原点移動付きべき乗法の例 eigen-p2.sce

```
n=6;
X=[1;0;0;0;0;0];
Y=[0;0;0;0;0;0];
for i=1:n
  A(i,i)= A(i,i) - Shift;
end

EIG0= 1.0;
for iter= 1:100
  for i=1:n
    Y(i)=0.0;
  end
  for i=1:n
    for j=1:n
      Y(i)= Y(i) + A(i,j)*X(j);
    end
  end

  Eigen= Shift;
  DL   = 0.;
  for i= 1:n
    Eigen= Eigen + X(i)*Y(i);
    DL   = DL    + Y(i)*Y(i);
  end
```

```
  RDL= 1.0/sqrt(DL);
  for i= 1:n
    X(i)= Y(i)*RDL;
  end

  RESID= sqrt((Eigen-EIG0)*                ..
         (Eigen-EIG0)/(EIG0*EIG0));

  if RESID < EPS then
    break;
  end;
  EIG0= Eigen;
end
```

逆べき乗法の例(シフト無し) eigen-p3.sce

```
exec( 'C:¥eigen¥lu.sci' );     外部関数
exec('C:¥eigen¥fbs.sci');

n=6;
X=[1;0;0;0;0;0];
Y=[0;0;0;0;0;0];

[A]=lu(A,n);     LU分解（関数呼び出し）

EIG0= 1.0e-10;
for iter= 1:100
  for i=1:n
    Y(i)=X(i);
  end
  [Y]=fbs (A,Y,n);      前進後退代入

  REIG= 0.;
  DL  = 0.;
  for i= 1:n
    REIG= REIG + X(i)*Y(i);
    DL  = DL   + Y(i)*Y(i);
  end

  Eigen= 1.0/REIG;
```

```
  RDL= 1.0/sqrt(DL);
  for i=1:n
    X(i)= Y(i)*RDL;
  end

  RESID= sqrt((Eigen-EIG0)*  ..
          (Eigen-EIG0)/(EIG0*EIG0));

  if RESID < EPS then
    break;
  end;
  EIG0= Eigen;
end
```

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \Rightarrow \mathbf{A}^{-1}\mathbf{x} = \lambda^{-1}\mathbf{x}$$

$$\mathbf{A}' = \mathbf{A}^{-1}, \lambda' = \lambda^{-1}$$

$\mathbf{A}'\mathbf{x} = \lambda'\mathbf{x}$ にべき乗法適用

逆べき乗法の例(シフト無し) eigen-p3.sce

```
exec( 'C:¥eigen¥lu.sci' );     外部関数
exec('C:¥eigen¥fbs.sci');

n=6;
X=[1;0;0;0;0;0];
Y=[0;0;0;0;0;0];

[A]=lu(A,n);     LU分解（関数呼び出し）

EIG0= 1.0e-10;
for iter= 1:100
  for i=1:n
    Y(i)=X(i);
  end
  [Y]=fbs (A,Y,n);      前進後退代入

  REIG= 0.;
  DL  = 0.;
  for i= 1:n
    REIG= REIG + X(i)*Y(i);
    DL  = DL   + Y(i)*Y(i);
  end

  Eigen= 1.0/REIG;
  RDL= 1.0/sqrt(DL);
  for i=1:n
    X(i)= Y(i)*RDL;
  end

  RESID= sqrt((Eigen-EIG0)*  ..
          (Eigen-EIG0)/(EIG0*EIG0));

  if RESID < EPS then
    break;
  end;
  EIG0= Eigen;
end
```

$$\mathbf{y} = \mathbf{A}'\mathbf{x} = \mathbf{A}^{-1}\mathbf{x}$$

逆行列をかけるかわりに
LU分解の前進後退代入
を実施する。

逆べき乗法の例(シフト無し) eigen-p3.sce

```
exec( 'C:¥eigen¥lu.sci' );     外部関数
exec('C:¥eigen¥fbs.sci');

n=6;
X=[1;0;0;0;0;0];
Y=[0;0;0;0;0;0];

[A]=lu(A,n);     LU分解（関数呼び出し）

EIG0= 1.0e-10;
for iter= 1:100
  for i=1:n
    Y(i)=X(i);
  end
  [Y]=fbs (A,Y,n);      前進後退代入

  REIG= 0.;
  DL  = 0.;
  for i= 1:n
    REIG= REIG + X(i)*Y(i);
    DL  = DL   + Y(i)*Y(i);
  end

  Eigen= 1.0/REIG;
```

```
  RDL= 1.0/sqrt(DL);
  for i=1:n
    X(i)= Y(i)*RDL;
  end

  RESID= sqrt((Eigen-EIG0)*  ..
          (Eigen-EIG0)/(EIG0*EIG0));

  if RESID < EPS then
    break;
  end;
  EIG0= Eigen;
end
```

[A]= lu(A,n)

出力変数　関数　入力変数

[y1,...,yn]= foo(x1,...,xm)

- 行列の固有値問題
- べき乗法
- 対称行列の固有値計算法:ヤコビ法

対称行列の固有値計算法

- 実対称行列の固有値⇒実数
  - 弾性振動問題等
- 相似変換
  - N×Nの正方行列**A**, **B**に対して以下を満たすような正則行列Pが存在するとする:

$$\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\,\mathbf{P}$$

  - このとき**A**と**B**は相似（similar）であると呼び，Bは**A**を相似変換した行列であると言う。

  - **A**と**B**が相似であればそれらの固有値は一致する
  - 任意の固有値に対する**B**の固有ベクトルを **x** とすると，**A**の固有ベクトルは **Px** となる（証明略）

ヤコビ法（Jacobi）（1/5）

- 実対称行列Aに対して二次元の回転に相当する相似変換を繰り返しながら対角行列へ近づけて行く方法
- 結果は合成されたn次元の回転となる
- 繰り返しの段階で得られる行列において指定した行列成分が0になるような相似変換

$$\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\,\mathbf{P}$$

- $P_{pp}=P_{qq}=\cos\phi,\ P_{pq}=\sin\phi,\ P_{qp}=-\sin\phi$
- それ以外は単位行列と同じ

$$\mathbf{P} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{c} \begin{array}{ccccccc} & & \text{p列目} & & \text{q列目} & & \\ \end{array} \\ \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & -\sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin\phi & \cdots & \cos\phi & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \\ \text{p行目} \\ \\ \text{q行目} \\ \\ \end{array}$$

$\mathbf{P}^{-1} = \mathbf{P}^{T}$ 直交行列

$\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\,\mathbf{P}$ 直交変換

$$\mathbf{B}^{T} = (\mathbf{P}^{-1}\mathbf{A}\,\mathbf{P})^{T} = \mathbf{P}^{T}\mathbf{A}\,(\mathbf{P}^{-1})^{T}$$

$$= \mathbf{P}^{-1}\mathbf{A}\,\mathbf{P} = \mathbf{B}$$ 対称

ヤコビ法（Jacobi）（2/5）

$$b_{ij} = a_{ij}, \quad i, j \neq p, q$$

$$b_{pk} = b_{kp} = a_{pk}\cos\phi + a_{qk}\sin\phi, \quad k \neq p, q$$

$$b_{qk} = b_{kq} = -a_{pk}\sin\phi + a_{qk}\cos\phi, \quad k \neq p, q$$

$$b_{pp} = \frac{a_{pp} + a_{qq}}{2} - \frac{a_{pp} - a_{qq}}{2}\cos 2\phi + a_{pq}\sin 2\phi$$

$$b_{qq} = \frac{a_{pp} + a_{qq}}{2} + \frac{a_{pp} - a_{qq}}{2}\cos 2\phi - a_{pq}\sin 2\phi$$

$$b_{pq} = b_{qp} = -\frac{a_{pp} - a_{qq}}{2}\sin 2\phi + a_{pq}\cos 2\phi$$

$$b_{pq} = b_{qp} = 0 \Rightarrow \tan 2\phi = \frac{2a_{pq}}{a_{pp} - a_{qq}} \Rightarrow \phi = \frac{1}{2}\tan^{-1}\frac{2a_{pq}}{a_{pp} - a_{qq}}$$

$$if\left(a_{pp} = a_{qq}\right) \Rightarrow \phi = \frac{\pi}{4}$$

繰り返すことによって行列全体の非対角成分が最終的に全て0に収束することが期待される

$$\begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & \sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & -\sin\phi & \cdots & \cos\phi & & \\ 0 & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} & & & & & & \\ & \ddots & & & & & \\ & & a_{pp} & a_{pk} & a_{pq} & & \\ & & \vdots & \ddots & \vdots & & \\ & & a_{qp} & a_{qk} & a_{qq} & & \\ & & & & & \ddots & \\ & & & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & -\sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin\phi & \cdots & \cos\phi & & \\ 0 & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

B= P^{-1}A P, A'= P^{-1}A

A':p行 $a'_{pk} = a_{pk}\cos\phi + a_{qk}\sin\phi$

A':q行 $a'_{qk} = -a_{pk}\sin\phi + a_{qk}\cos\phi$

A':p行，**P**:k列（a_{kk}=1，他は0） $b_{pk} = b_{kp} = a'_{pk}$， $k \neq p, q$

A':q行，**P**:k列（a_{kk}=1，他は0） $b_{qk} = b_{kq} = a'_{qk}$， $k \neq p, q$

$$\begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & \sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & -\sin\phi & \cdots & \cos\phi & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} & & & & & & \\ & \ddots & & & & & \\ & & a_{pp} & a_{pk} & a_{pq} & & \\ & & \vdots & \ddots & \vdots & & \\ & & a_{qp} & a_{qk} & a_{qq} & & \\ & & & & & \ddots & \\ & & & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & -\sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin\phi & \cdots & \cos\phi & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

B= P^{-1}A P, A'= P^{-1}A

A':p行 $a'_{pk} = a_{pk}\cos\phi + a_{qk}\sin\phi$

A':q行 $a'_{qk} = -a_{pk}\sin\phi + a_{qk}\cos\phi$

A':p行，**P**:k列（a_{kk}=1，他は0） $b_{pk} = b_{kp} = a'_{pk}, \quad k \neq p, q$

A':q行，**P**:k列（a_{kk}=1，他は0） $b_{qk} = b_{kq} = a'_{qk}, \quad k \neq p, q$

$$\begin{bmatrix} & & & & & \\ & \ddots & & & & \\ & & a'_{pp} & a'_{pk} & a'_{pq} & \\ & & \vdots & \ddots & \vdots & \\ & & a'_{qp} & a'_{qk} & a'_{qq} & \\ & & & & & \ddots \\ & & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & 0 \\ & \ddots & & & & \\ & & \cos\phi & \cdots & -\sin\phi & \\ & & \vdots & \ddots & \vdots & \\ & & \sin\phi & \cdots & \cos\phi & \\ & 0 & & & & \ddots \\ & & & & & 1 \end{bmatrix}$$

B= **P**$^{-1}$**A P, A'**= **P**$^{-1}$**A**

A':p行 $a'_{pk} = a_{pk}\cos\phi + a_{qk}\sin\phi$

A':q行 $a'_{qk} = -a_{pk}\sin\phi + a_{qk}\cos\phi$

A':p行，**P**:k列（a_{kk}=1，他は0） $b_{pk} = b_{kp} = a'_{pk}, \quad k \neq p, q$

A':q行，**P**:k列（a_{kk}=1，他は0） $b_{qk} = b_{kq} = a'_{qk}, \quad k \neq p, q$

$$\begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & \sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & -\sin\phi & \cdots & \cos\phi & & \\ 0 & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} & & & & & & \\ & \ddots & & & & & \\ & & a_{pp} & \cdots & a_{pq} & & \\ & & \vdots & \ddots & \vdots & & \\ & & a_{qp} & \cdots & a_{qq} & & \\ & & & & & \ddots & \\ & & & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & -\sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin\phi & \cdots & \cos\phi & & \\ 0 & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

B= P^{-1}A P, A'= P^{-1}A

$$a'_{pp} = a_{pp}\cos\phi + a_{qp}\sin\phi,\ a'_{pq} = a_{pq}\cos\phi + a_{qq}\sin\phi$$ （**A'**:p行）

$$b_{pp} = a'_{pp}\cos\phi + a'_{pq}\sin\phi$$ （**A'**:p行, **P**:p列）

$$= \left(a_{pp}\cos\phi + a_{qp}\sin\phi\right)\cos\phi + \left(a_{pq}\cos\phi + a_{qq}\sin\phi\right)\sin\phi$$

$$= a_{pp}\cos^2\phi + 2a_{pq}\cos\phi\sin\phi + a_{qq}\sin^2\phi \quad \left(\because a_{pq} = a_{qp}\right)$$

$$= \frac{a_{pp}\left(\cos 2\phi + 1\right)}{2} + a_{pq}\sin 2\phi + \frac{a_{qq}\left(1 - \cos 2\phi\right)}{2}$$

$$= \frac{a_{pp} + a_{qq}}{2} + \frac{a_{pp} - a_{qq}}{2}\cos 2\phi - a_{pq}\sin 2\phi$$

$$\begin{bmatrix} & & & & \\ & \boxed{\begin{matrix} a'_{pp} & \cdots & a'_{pq} \\ \vdots & \ddots & \vdots \\ a'_{qp} & \cdots & a'_{qq} \end{matrix}} & & & \\ & & \ddots & & \\ & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & \boxed{\begin{matrix} \cos\phi & \cdots & -\sin\phi \\ \vdots & \ddots & \vdots \\ \sin\phi & \cdots & \cos\phi \end{matrix}} & & \\ & 0 & & \ddots & \\ & & & & 1 \end{bmatrix}$$

B= P^{-1}A P, A'= P^{-1}A

$a'_{pp} = a_{pp}\cos\phi + a_{qp}\sin\phi,\ a'_{pq} = a_{pq}\cos\phi + a_{qq}\sin\phi$ （**A'**:p行）

$$b_{pp} = a'_{pp}\cos\phi + a'_{pq}\sin\phi \quad \text{（\textbf{A'}:p行, \textbf{P}:p列）}$$

$$= \left(a_{pp}\cos\phi + a_{qp}\sin\phi\right)\cos\phi + \left(a_{pq}\cos\phi + a_{qq}\sin\phi\right)\sin\phi$$

$$= a_{pp}\cos^2\phi + 2a_{pq}\cos\phi\sin\phi + a_{qq}\sin^2\phi \quad \left(\because a_{pq} = a_{qp}\right)$$

$$= \frac{a_{pp}\left(\cos 2\phi + 1\right)}{2} + a_{pq}\sin 2\phi + \frac{a_{qq}\left(1 - \cos 2\phi\right)}{2}$$

$$= \frac{a_{pp} + a_{qq}}{2} + \frac{a_{pp} - a_{qq}}{2}\cos 2\phi - a_{pq}\sin 2\phi$$

$$\begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & \sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & -\sin\phi & \cdots & \cos\phi & & \\ 0 & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} & & & & & & \\ & \ddots & & & & & \\ & & a_{pp} & \cdots & a_{pq} & & \\ & & \vdots & \ddots & \vdots & & \\ & & a_{qp} & \cdots & a_{qq} & & \\ & & & & & \ddots & \\ & & & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & -\sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin\phi & \cdots & \cos\phi & & \\ 0 & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

B= P^{-1}A P, A'= P^{-1}A

$$a'_{qp} = -a_{pp}\sin\phi + a_{qp}\cos\phi,\ a'_{qq} = -a_{pq}\sin\phi + a_{qq}\cos\phi$$ （**A'**:q行）

$$b_{qq} = -a'_{qp}\sin\phi + a'_{qq}\cos\phi$$ （**A'**:q行, **P**:q列）

$$= -\left(-a_{pp}\sin\phi + a_{qp}\cos\phi\right)\sin\phi + \left(-a_{pq}\sin\phi + a_{qq}\cos\phi\right)\cos\phi$$

$$= a_{pp}\sin^2\phi - 2a_{pq}\cos\phi\sin\phi + a_{qq}\cos^2\phi \quad \left(\because a_{pq} = a_{qp}\right)$$

$$= \frac{a_{pp}\left(1-\cos 2\phi\right)}{2} - a_{pq}\sin 2\phi + \frac{a_{qq}\left(1+\cos 2\phi\right)}{2}$$

$$= \frac{a_{pp}+a_{qq}}{2} - \frac{a_{pp}-a_{qq}}{2}\cos 2\phi - a_{pq}\sin 2\phi$$

$$\begin{bmatrix} & & & & \\ & \ddots & & & \\ & & a'_{pp} & \cdots & a'_{pq} \\ & & \vdots & \ddots & \vdots \\ & & a'_{qp} & \cdots & a'_{qq} \\ & & & & \ddots \\ & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & -\sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin\phi & \cdots & \cos\phi & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

B= P^{-1}A P, A'= P^{-1}A

$a'_{qp} = -a_{pp}\sin\phi + a_{qp}\cos\phi,\ a'_{qq} = -a_{pq}\sin\phi + a_{qq}\cos\phi$ （**A'**:q行）

$$b_{qq} = -a'_{qp}\sin\phi + a'_{qq}\cos\phi \quad \text{（A':q行, P:q列）}$$

$$= -\left(-a_{pp}\sin\phi + a_{qp}\cos\phi\right)\sin\phi + \left(-a_{pq}\sin\phi + a_{qq}\cos\phi\right)\cos\phi$$

$$= a_{pp}\sin^2\phi - 2a_{pq}\cos\phi\sin\phi + a_{qq}\cos^2\phi \quad \left(\because a_{pq} = a_{qp}\right)$$

$$= \frac{a_{pp}\left(1-\cos 2\phi\right)}{2} - a_{pq}\sin 2\phi + \frac{a_{qq}\left(1+\cos 2\phi\right)}{2}$$

$$= \frac{a_{pp}+a_{qq}}{2} - \frac{a_{pp}-a_{qq}}{2}\cos 2\phi - a_{pq}\sin 2\phi$$

$$\begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & \sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & -\sin\phi & \cdots & \cos\phi & & \\ & & & & & \ddots & \\ 0 & & & & & & 1 \end{bmatrix} \begin{bmatrix} & & & & & & \\ & \ddots & & & & & \\ & & a_{pp} & \cdots & a_{pq} & & \\ & & \vdots & \ddots & \vdots & & \\ & & a_{qp} & \cdots & a_{qq} & & \\ & & & & & \ddots & \\ & & & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & -\sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin\phi & \cdots & \cos\phi & & \\ & & & & & \ddots & \\ 0 & & & & & & 1 \end{bmatrix}$$

B= P^{-1}A P, A'= P^{-1}A

$$a'_{pp} = a_{pp}\cos\phi + a_{pq}\sin\phi,\ a'_{pq} = a_{pq}\cos\phi + a_{qq}\sin\phi$$ （**A'**:p行）

$$b_{pq} = -a'_{pp}\sin\phi + a'_{pq}\cos\phi$$ （**A'**:p行, **P**:q列）

$$= -\left(a_{pp}\cos\phi + a_{pq}\sin\phi\right)\sin\phi + \left(a_{pq}\cos\phi + a_{qq}\sin\phi\right)\cos\phi$$

$$= -a_{pp}\cos\phi\sin\phi + a_{pq}\cos^2\phi - a_{pq}\sin^2\phi + a_{qq}\cos\phi\sin\phi \left(\because a_{pq} = a_{qp}\right)$$

$$= -\frac{\left(a_{pp} - a_{qq}\right)}{2}\sin 2\phi + a_{pq}\cos 2\phi$$

$$\begin{bmatrix} & & & & \\ & \boxed{\begin{matrix} a'_{pp} & \cdots & a'_{pq} \\ \vdots & \ddots & \vdots \\ a'_{qp} & \cdots & a'_{qq} \end{matrix}} & & & \\ & & \ddots & & \\ & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & \boxed{-\sin\phi} & & \\ & & \vdots & \ddots & \boxed{\vdots} & & \\ & & \sin\phi & \cdots & \boxed{\cos\phi} & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

B= P^{-1}A P, A'= P^{-1}A

$a'_{pp} = a_{pp}\cos\phi + a_{pq}\sin\phi,\ a'_{pq} = a_{pq}\cos\phi + a_{qq}\sin\phi$ （**A'**:p行）

$$\begin{aligned} b_{pq} &= -a'_{pp}\sin\phi + a'_{pq}\cos\phi \quad \text{（\textbf{A'}:p行, \textbf{P}:q列）} \\ &= -\left(a_{pp}\cos\phi + a_{pq}\sin\phi\right)\sin\phi + \left(a_{pq}\cos\phi + a_{qq}\sin\phi\right)\cos\phi \\ &= -a_{pp}\cos\phi\sin\phi + a_{pq}\cos^2\phi - a_{pq}\sin^2\phi + a_{qq}\cos\phi\sin\phi \left(\because a_{pq} = a_{qp}\right) \\ &= -\frac{\left(a_{pp} - a_{qq}\right)}{2}\sin 2\phi + a_{pq}\cos 2\phi \end{aligned}$$

$$\begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & \sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & -\sin\phi & \cdots & \cos\phi & & \\ & & & & & \ddots & \\ 0 & & & & & & 1 \end{bmatrix} \begin{bmatrix} & & & & & & \\ & \ddots & & & & & \\ & & a_{pp} & \cdots & a_{pq} & & \\ & & \vdots & \ddots & \vdots & & \\ & & a_{qp} & \cdots & a_{qq} & & \\ & & & & & \ddots & \\ & & & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & -\sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin\phi & \cdots & \cos\phi & & \\ & & & & & \ddots & \\ 0 & & & & & & 1 \end{bmatrix}$$

B= P^{-1}A P, A'= P^{-1}A

$$a'_{qp} = -a_{pp}\sin\phi + a_{qp}\cos\phi,\ a'_{qq} = -a_{qp}\sin\phi + a_{qq}\cos\phi$$ （**A'**:q行）

$$b_{qp} = a'_{qp}\cos\phi + a'_{qq}\sin\phi$$ （**A'**:q行, **P**:p列）

$$= \left(-a_{pp}\sin\phi + a_{qp}\cos\phi\right)\cos\phi + \left(-a_{qp}\sin\phi + a_{qq}\cos\phi\right)\sin\phi$$

$$= -a_{pp}\cos\phi\sin\phi + a_{pq}\cos^2\phi - a_{pq}\sin^2\phi + a_{qq}\cos\phi\sin\phi \left(\because a_{pq} = a_{qp}\right)$$

$$= -\frac{\left(a_{pp} - a_{qq}\right)}{2}\sin 2\phi + a_{pq}\cos 2\phi = b_{pq}$$

$$\begin{bmatrix} \ddots & & & & \\ & a'_{pp} & \cdots & a'_{pq} & \\ & \vdots & \ddots & \vdots & \\ & a'_{qp} & \cdots & a'_{qq} & \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & -\sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin\phi & \cdots & \cos\phi & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

B= P^{-1}A P, A'= P^{-1}A

$a'_{qp} = -a_{pp}\sin\phi + a_{qp}\cos\phi,\ a'_{qq} = -a_{qp}\sin\phi + a_{qq}\cos\phi$ （A':q行）

$b_{qp} = a'_{qp}\cos\phi + a'_{qq}\sin\phi$ （**A'**:q行, **P**:p列）

$$= \left(-a_{pp}\sin\phi + a_{qp}\cos\phi\right)\cos\phi + \left(-a_{qp}\sin\phi + a_{qq}\cos\phi\right)\sin\phi$$

$$= -a_{pp}\cos\phi\sin\phi + a_{pq}\cos^2\phi - a_{pq}\sin^2\phi + a_{qq}\cos\phi\sin\phi \left(\because a_{pq} = a_{qp}\right)$$

$$= -\frac{\left(a_{pp} - a_{qq}\right)}{2}\sin 2\phi + a_{pq}\cos 2\phi = b_{pq}$$

実行列の要素の平方和

$$tr\left(\mathbf{A}^T\mathbf{A}\right)=\sum_{i=1}^{n}\left(\mathbf{A}^T\mathbf{A}\right)_{ij}=\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\mathbf{A}^T\right)_{ij}\left(\mathbf{A}^T\right)_{ji}=\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}a_{ji}=\sum_{i=1}^{n}\sum_{j=1}^{n}\left(a_{ij}\right)^2$$

tr: $trace$　対角和

$$\mathbf{B}=\mathbf{P}^{-1}\mathbf{A}\mathbf{P}=\mathbf{P}^T\mathbf{A}\mathbf{P}$$

$$tr\left(\mathbf{B}^T\mathbf{B}\right)=tr\left(\mathbf{P}^T\mathbf{A}^T\mathbf{P}\cdot\mathbf{P}^T\mathbf{A}\mathbf{P}\right)=tr\left(\mathbf{P}^T\mathbf{A}^T\mathbf{A}\mathbf{P}\right)=tr\left(\mathbf{P}^T\cdot\mathbf{A}^T\mathbf{A}\mathbf{P}\right)$$

$$=tr\left(\mathbf{A}^T\mathbf{A}\mathbf{P}\cdot\mathbf{P}^T\right)$$

$$\left(\because tr\left(\mathbf{A}\mathbf{B}\right)=tr\left(\mathbf{B}\mathbf{A}\right)\right)$$

$$\therefore tr\left(\mathbf{B}^T\mathbf{B}\right)=tr\left(\mathbf{A}^T\mathbf{A}\mathbf{P}\cdot\mathbf{P}^T\right)=tr\left(\mathbf{A}^T\mathbf{A}\right)\Rightarrow\sum_{i=1}^{n}\sum_{j=1}^{n}\left(a_{ij}\right)^2=\sum_{i=1}^{n}\sum_{j=1}^{n}\left(b_{ij}\right)^2$$

直交変換によって行列成分の平方和（二乗和）は一定に保たれる

ヤコビ法（Jacobi）（3/5）

$$(i, j \neq p, q)\, b_{ij} = a_{ij} \Rightarrow {b_{ij}}^2 = {a_{ij}}^2$$

$$(k \neq p, q)\, {b_{pk}}^2 + {b_{qk}}^2 = {a_{pk}}^2 + {a_{qk}}^2$$

$${b_{pp}}^2 + 2{b_{pq}}^2 + {b_{qq}}^2 = {a_{pp}}^2 + 2{a_{pq}}^2 + {a_{qq}}^2$$

$$\Downarrow$$

$${b_{pp}}^2 + {b_{qq}}^2 = {a_{pp}}^2 + 2{a_{pq}}^2 + {a_{qq}}^2 \quad (\because b_{pq} = 0)$$

- 直交変換（相似変換）によって対角成分の二乗和が増加
- 行列全体の成分の二乗和は直交変換（相似変換）によって一定に保たれている
- 非対角成分の二乗和が減少している⇒繰り返すことによって0に近づく

ヤコビ法（Jacobi）（4/5）

以下のように相似変換，行列に番号をつける（$\mathbf{A}=\mathbf{A}_1$）

$$\mathbf{A}_{m+1} = \mathbf{P}_m{}^{-1}\mathbf{A}_m\mathbf{P}_m, \quad m = 1,2,\ldots$$

$$\mathbf{T} = \lim_{m\to\infty} \mathbf{P}_1\mathbf{P}_2\mathbf{P}_3\cdots\mathbf{P}_m$$

これまでの結果から：

$$\mathbf{\Lambda} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$$

ここで$\mathbf{\Lambda}$は下記の対角行列である。行列の固有値は相似変換により不変に保たれるため，$\mathbf{\Lambda}$の対角成分が行列$\mathbf{A}$の固有値に他ならない

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

ヤコビ法（Jacobi）（5/5）

前ページの相似変換は下記のように表される:

$$\mathbf{T}\boldsymbol{\Lambda} = \mathbf{A}\mathbf{T}$$

行列 $\mathbf{T}$ の第k列からなる列ベクトルを $\boldsymbol{t}_k$とすると，下記が成立:

$$\mathbf{A}\mathbf{t}_k = \lambda_k \mathbf{t}_k$$

列ベクトル $\boldsymbol{t}_k$ は固有値 λ_k に対応する固有ベクトルとなる。

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 3 & 2 & 0 & 2 \\ 3 & 0 & 3 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix}$$

```
### Original Matrix
  1.000E+00  3.000E+00  3.000E+00  1.000E+00
  3.000E+00  2.000E+00  0.000E+00  2.000E+00
  3.000E+00  0.000E+00  3.000E+00  1.000E+00
  1.000E+00  2.000E+00  1.000E+00  1.000E+00
  1          2
 -7.028238E-01
```

a_{pq}の絶対値最大成分（p, q）

回転角度

```
### ITERATIONS        1     5.477226E+00
 -1.541E+00 -4.441E-16  2.289E+00 -5.297E-01
 -2.220E-16  4.541E+00  1.939E+00  2.172E+00
  2.289E+00  1.939E+00  3.000E+00  1.000E+00
 -5.297E-01  2.172E+00  1.000E+00  1.000E+00
  1          3
 -3.947134E-01
```

b_{pq}の絶対値最大成分（p, q）

1回目の反復で0になった(1,2)・(2,1)成分の絶対値が再び大きくなっている

```
### ITERATIONS        2     4.418191E+00
 -2.495E+00 -7.457E-01  2.220E-16 -8.735E-01
 -7.457E-01  4.541E+00  1.790E+00  2.172E+00
  0.000E+00  1.790E+00  3.954E+00  7.194E-01
 -8.735E-01  2.172E+00  7.194E-01  1.000E+00
  2          4
  4.434640E-01
```

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 3 & 2 & 0 & 2 \\ 3 & 0 & 3 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix}$$

```
### ITERATIONS        3     3.175159E+00
 -2.495E+00 -1.048E+00  2.220E-16 -4.691E-01
 -1.048E+00  5.573E+00  1.926E+00 -1.197E-16
  0.000E+00  1.926E+00  3.954E+00 -1.182E-01
 -4.691E-01  0.000E+00 -1.182E-01 -3.194E-02
  2         3
  5.863310E-01

### ITERATIONS        4     1.632843E+00
 -2.495E+00 -8.733E-01  5.801E-01 -4.691E-01
 -8.733E-01  6.852E+00 -2.220E-16 -6.542E-02
  5.801E-01 -4.441E-16  2.675E+00 -9.848E-02
 -4.691E-01 -6.542E-02 -9.848E-02 -3.194E-02
  1         2
  9.235850E-02

### ITERATIONS        5     1.068181E+00
 -2.576E+00  0.000E+00  5.776E-01 -4.731E-01
  1.665E-16  6.933E+00 -5.350E-02 -2.188E-02
  5.776E-01 -5.350E-02  2.675E+00 -9.848E-02
 -4.731E-01 -2.188E-02 -9.848E-02 -3.194E-02
  2         4
 -4.763515E-08
```

1回0になった成分の絶対値が再び大きくなっているが全体としては0に近づいている

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 3 & 2 & 0 & 2 \\ 3 & 0 & 3 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix}$$

```
### ITERATIONS        17     2.292314E-10
 -2.717E+00 -1.612E-10  2.462E-16 -1.707E-11
 -1.612E-10  6.934E+00  2.199E-16 -4.912E-17
 -1.323E-23  4.453E-18  2.745E+00 -7.456E-13
 -1.707E-11 -5.294E-23 -7.458E-13  3.867E-02
  1         2
  1.670097E-11

### ITERATIONS        18     2.415961E-11
 -2.717E+00 -1.406E-16  2.462E-16 -1.707E-11
 -6.462E-27  6.934E+00  2.199E-16 -4.912E-17
 -1.323E-23  4.453E-18  2.745E+00 -7.456E-13
 -1.707E-11  2.321E-22 -7.458E-13  3.867E-02
  1         4
  6.192487E-12

### ITERATIONS        19     1.054565E-12
 -2.717E+00 -1.406E-16  2.462E-16  3.232E-17
 -6.767E-27  6.934E+00  2.199E-16 -4.912E-17
 -1.785E-23  4.453E-18  2.745E+00 -7.456E-13
  0.000E+00  2.321E-22 -7.458E-13  3.867E-02
  3         4
```

絶対値最大成分＜1.e-12のため終了

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 3 & 2 & 0 & 2 \\ 3 & 0 & 3 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix}$$

```
### ITERATIONS        19     1.054565E-12
 -2.717E+00 -1.406E-16  2.462E-16  3.232E-17
 -6.767E-27  6.934E+00  2.199E-16 -4.912E-17
 -1.785E-23  4.453E-18  2.745E+00 -7.456E-13
  0.000E+00  2.321E-22 -7.458E-13  3.867E-02
  3        4

### Eigenvalues/Eigenvectors
 1 -2.717E+00  7.073E-01 -5.384E-01 -4.077E-01  2.091E-01
 2  6.934E+00  5.822E-01  4.984E-01  5.345E-01  3.562E-01
 3  2.745E+00  2.994E-02 -6.222E-01  7.318E-01 -2.766E-01
 4  3.867E-02 -3.999E-01 -2.732E-01  1.121E-01  8.677E-01
```

固有値 $\lambda_1 \sim \lambda_4$	λ_1に対応する固有ベクトル	λ_2に対応する固有ベクトル	λ_3に対応する固有ベクトル	λ_4に対応する固有ベクトル

レポート課題

- 提出期限：2月13日（月）13:00（レポートボックス設置）
- 下記の対称行列の固有値をヤコビ法によって求めよ
  - 算出手順の概要
  - プログラムリスト（プログラムを作成する場合）
  - p.52-55に相当する算出経過を添付すること
  - 更に固有ベクトルを求めた場合は加点する
    - 固有値のみ：70点（合格）
    - ＋固有ベクトル：100点（Tを保存しておく）

$$A = \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 5 & 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 4 & 3 & 2 & 1 \\ 3 & 3 & 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

プログラム例Fortran(1/4)

```
subroutine JACOBI (A,E,V,N,OUT,MAX)
implicit REAL*8 (A-H,O-Z)
dimension A(N,N), E(N), V(N,N)
integer   P, Q

NM1= N - 1

do iter= 1, MAX

  P= 1
  Q= 2
  do i= 1, NM1
  do j= i+1, N
    if (dabs(A(i,j)).gt.dabs(A(P,Q))) then
      P= i
      Q= j
    endif
  enddo
  enddo
```

最大のA(P,Q)
を探索(P≠Q)

プログラム例Fortran(2/4)

```
 do iter= 1, MAX

(…)

   if (dabs(A(P,Q)).lt.1.d-12) exit

   if (dabs(A(P,P)-A(Q,Q)).lt.1.d-24) then
     T= datan(1.d0)
    else
     R= 2.d0*A(P,Q)/(A(P,P)-A(Q,Q))
     T= datan(R)*0.5d0
   endif
```

最大のA(P,Q)
が10^{-12}未満だったら
終了

$$T:\phi$$

$$\phi=\frac{1}{2}\tan^{-1}\frac{2a_{pq}}{a_{pp}-a_{qq}}$$

$$if\left(a_{pp}=a_{qq}\right)\Rightarrow\phi=\frac{\pi}{4}=\tan^{-1}1.00$$

プログラム例Fortran(3/4)

```
 do iter= 1, MAX

(…)

   S= dsin(T)
   C= dcos(T)
   do k= 1, N
     Apk= A(P,k)
     Aqk= A(Q,k)
     A(P,k)=  Apk*C + Aqk*S
     A(Q,k)= -Apk*S + Aqk*C
   enddo
   do k= 1, N
     Akp= A(k,P)
     Akp= A(k,Q)
     A(k,P)=  Akp*C + Akq*S
     A(k,Q)= -Akp*S + Akq*C
   enddo
```

プログラム例Fortran(3/4)

```
 do iter= 1, MAX

(…)

   S= dsin(T)
   C= dcos(T)
   do k= 1, N
      Apk= A(P,k)
      Aqk= A(Q,k)
      A(P,k)=  Apk*C + Aqk*S
      A(Q,k)= -Apk*S + Aqk*C
   enddo
```

$$\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\,\mathbf{P}$$

$$\mathbf{A'} = \mathbf{P}^{-1}\mathbf{A}$$

$$\begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & \sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & -\sin\phi & \cdots & \cos\phi & & \\ 0 & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} & & & & & & \\ & \ddots & & & & & \\ & & a_{pp} & a_{pk} & a_{pq} & & \\ & & \vdots & \ddots & \vdots & & \\ & & a_{qp} & a_{qk} & a_{qq} & & \\ & & & & & \ddots & \\ & & & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & \cos\phi & \cdots & -\sin\phi & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin\phi & \cdots & \cos\phi & & \\ 0 & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

プログラム例Fortran(3/4)

```
 do iter= 1, MAX

(…)

   S= dsin(T)
   C= dcos(T)
```

$$\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\,\mathbf{P}$$

$$\mathbf{A'} = \mathbf{P}^{-1}\mathbf{A}$$

$$\mathbf{B} = \mathbf{A'}\mathbf{P}$$

$$\begin{bmatrix} & & & & & \\ & \ddots & & & & \\ & & a'_{pp} & \cdots & a'_{pq} & \\ & & a'_{kp} & \ddots & a'_{kq} & \\ & & a'_{qp} & \cdots & a'_{qq} & \\ & & & & & \ddots \\ & & & & & \end{bmatrix} \begin{bmatrix} 1 & & & & & 0 \\ & \ddots & & & & \\ & & \cos\phi & \cdots & -\sin\phi & \\ & & \vdots & \ddots & \vdots & \\ & & \sin\phi & \cdots & \cos\phi & \\ & 0 & & & & \ddots \\ & & & & & 1 \end{bmatrix}$$

```
   do k= 1, N
     Akp= A(k,P)
     Akp= A(k,Q)
     A(k,P)=  Akp*C + Akq*S
     A(k,Q)= -Akp*S + Akq*C
   enddo
```

プログラム例Fortran(4/4)

```
 do iter= 1, MAX

(…)

   VAL= 0.d0
   do i= 1, N
   do j= 1, N
      if (i.ne.j) VAL= VAL + A(i,j)**2
   enddo
   enddo

   VAL= dsqrt(VAL)
   write (*,'(i8, 1pe16.6)') iter, VAL

 enddo

 return
 end
```

[A]の非対角成分の二乗和の平方根を計算⇒0