第 10 章

行列の Jordan 標準形

10.1　対角化可能な行列と Jordan 標準形

一般の n 次正方行列，$A \in \mathbb{C}^{n\times n}$ に対しては大きく二つのケースに分類される。

(A) **対角化可能な場合**　・・・すべての固有値に対応する固有空間の次元数 (=自由度) を合計すると次元数 n と一致する場合。例えば下記のケースが該当する。

(A-1) **すべて異なる固有値 λ_i を持つ場合**　・・・対応する固有空間 $\mathcal{V}_i$ から固有ベクトル $\mathbf{v}_i \in \mathcal{V}_i$ を持ってくると一次独立となり，この固有ベクトルを並べてできる正則行列 $V = [\mathbf{v}_1\ \mathbf{v}_2\ ...\ \mathbf{v}_n] \in \mathbb{C}^{n\times n}$ を用いて

$$V^{-1}AV = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} = \Lambda \tag{10.1}$$

と対角化できる。

(A-2) **重複固有値の重複度と固有空間の次元数 (=自由度) が一致する場合**　・・・例えば n 個の固有値がすべて重複していても ($\lambda = \lambda_1 = \lambda_2 = \cdots = \lambda_n$), $\mathcal{V} = \mathbb{C}^n$ であれば，n 個の一次独立な固有ベクトル $\mathbf{v}_i \in \mathbb{C}^n$ が取れ，(10.1) 同様の対角化が可能である。

(B) **対角化不可能な場合**　・・・重複度が 2 以上の固有値に対応する固有空間が一つでも重複度未満の次元数であれば，その部分が Jordan ブロックとなり，対角化は不可能になる。この場合は例えば $\lambda_1 = \lambda_2 = \lambda$ に該当する固有空間の次元数が 1 だとすると，$\mathbf{u}_2 \in \mathcal{U}_2 = \{\, \mathbf{u}_2 \mid (A - \lambda I)^2\mathbf{u}_2 = 0 \,\}$ から一般化固有ベクトル $\mathbf{u}_2 \in \mathcal{U}_2$ を取り，固有ベク

トル $\mathbf{v}_1 \in \mathcal{V}_1$ と合わせて $V = [\mathbf{v}_1\ \mathbf{u}_2\ \mathbf{v}_3\ \cdots \mathbf{v}_n]$ を作り

$$V^{-1}AV = \left[\begin{array}{cc|ccc} \lambda & 1 & & & \\ 0 & \lambda & & & \\ \hline & & \lambda_3 & & \\ & & & \ddots & \\ & & & & \lambda_n \end{array}\right] = J \tag{10.2}$$

となる。

一般に，重複度と固有空間の次元数 (=自由度) が一致しない固有値を持つ場合は対角化不可能となる。この場合は一般化固有空間の次元数と同じ次数の Jordan ブロックを持つ。

問題 10.1

$A \in \mathbb{C}^{3\times 3}$ の時，A の固有値 $\lambda_1, \lambda_2, \lambda_3$ と固有空間の次元数によってどのような Jordan 標準形 $J = V^{-1}AV$ になるか，整理せよ。

固有値の状態	固有空間の次元数	Jordan 標準形 $J = V^{-1}AV$	対角化可能？
$\lambda_1 = \lambda_2 = \lambda_3$	3	$\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_1 \end{bmatrix}$	○（可能）
	2	$\left[\begin{array}{cc\|c} \lambda_1 & 1 & 0 \\ 0 & \lambda_1 & 0 \\ \hline 0 & 0 & \lambda_1 \end{array}\right]$	×（不可能)
	1		×
$\lambda_1 = \lambda_2 \neq \lambda_3$	2 ($\mathcal{V}_1$ の次元数)	$\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$	○
	1 ($\mathcal{V}_1$ の次元数)		×
$\lambda_1 \neq \lambda_2 \neq \lambda_3$	1 ($\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3$ の次元数)	$\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$	○

10.2　固有値・固有ベクトルが既知の問題の作り方

実験的に，固有値と固有ベクトル，あるいは Jordan 標準形が既知の問題を使って，本当にプログラムが正確な固有値・固有ベクトルを求めることができているか，確認したいことがある。そのような場合は (10.1) や (10.2) の関係式を使って，あらかじめ対角行列 Λ や Jordan 標準形 J を与えておき，適当な正則行列 V を使って

$$A = V\Lambda V^{-1} \text{ または } VJV^{-1} \tag{10.3}$$

として行列 $A \in \mathbb{C}^{n\times n}$ を生成すればよい。

では，実際に固有値・固有ベクトルが既知の問題を作り，それを eig 関数で解いてみよう。ここで正則行列 V は

$$V = \begin{bmatrix} 1 & 2 & 3 \\ -3 & -2 & -2 \\ 3 & 4 & 5 \end{bmatrix}$$

とする。この時，逆行列 V^{-1} は

$$V^{-1} = \begin{bmatrix} 1 & -1 & -1 \\ -9/2 & 2 & 7/2 \\ 3 & -1 & -2 \end{bmatrix}$$

である。

10.2.1　対角化可能なケース

次の対角行列 Λ_1，Λ_2 のケースを考える。

それを MATLAB の eig 関数で

$$\Lambda_1 = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10.4}$$

```
>> Lambda_1 = diag([3, 2, 1])

Lambda_1 =

     3     0     0
     0     2     0
     0     0     1

>> A = V * Lambda_1 * inv_V

A =
```

```
    -6     2     5
     3     3    -1
   -12     2     9

>> eig(A)

ans =

    1.0000
    2.0000
    3.0000
```

固有値だけではハッキリしないので，固有ベクトルと対角行列も求めてみる。

```
>> [v, lambda] = eig(A)

v =

   -0.4867   -0.4082   -0.2294
    0.3244    0.4082    0.6882
   -0.8111   -0.8165   -0.6882


lambda =

    1.0000         0         0
         0    2.0000         0
         0         0    3.0000

>>
```

となって，正しく求められていることが分かる。

同様に，重複固有値を持つが，対角化可能であるケースを計算してみよう，

$$\Lambda_2 = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10.5}$$

同様に，$A = V\Lambda_2 V^{-1}$ を求めて対角化してみる。

```
>> Lambda_2 = diag([3, 3, 1])

Lambda_2 =

     3     0     0
     0     3     0
     0     0     1

>> A = V * Lambda_2 * inv_V

A =

   -15     6    12
    12    -1    -8
   -30    10    23
```

```
>> [v, lambda] = eig(A)

v =

   -0.4867   -0.4438    0.5816
    0.3244    0.3328    0.1435
   -0.8111   -0.8321    0.8007


lambda =

    1.0000         0         0
         0    3.0000         0
         0         0    3.0000
```

従って，正しく対角化できていることが分かる。

10.2.2 対角化不可能なケース

2 次以上の Jordan ブロックを含む Jordan 標準形のケースを計算してみよう。

$$J_1 = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix} \tag{10.6}$$

```
>> [v, lambda] = eig(A)
>> J = [3, 1, 0; 0, 3, 0; 0, 0, 3]

J =

     3     1     0
     0     3     0
     0     0     3

>> A = V * J * inv_V

A =

   -1.5000    2.0000    3.5000
   13.5000   -3.0000  -10.5000
  -13.5000    6.0000   13.5000
```

この場合，特にエラーもなく計算はできるが・・・

```
>> [v, lambda] = eig(A)

v =

    0.2294    0.2294   -0.4910
   -0.6882   -0.6882    0.3159
    0.6882    0.6882   -0.8118


lambda =
```

```
    3.0000         0         0
         0    3.0000         0
         0         0    3.0000
```

対角化できてしまい，しかも 1 番目と 2 番目の固有ベクトルが同じものになっていることが分かる。

原理的に，有限桁の浮動小数点数演算を使う限り，任意の正方行列の Jordan 標準形を求めることは難しい。そこで，丸め誤差の混入しない記号処理演算パッケージ (Symbolic Math Toolbox) を使った Jordan 標準形を求める jordan 関数が提供されている。この関数を使う際には必ずアドオンとして Symbolic Math Toolbox がインストールされていることを確認すること。

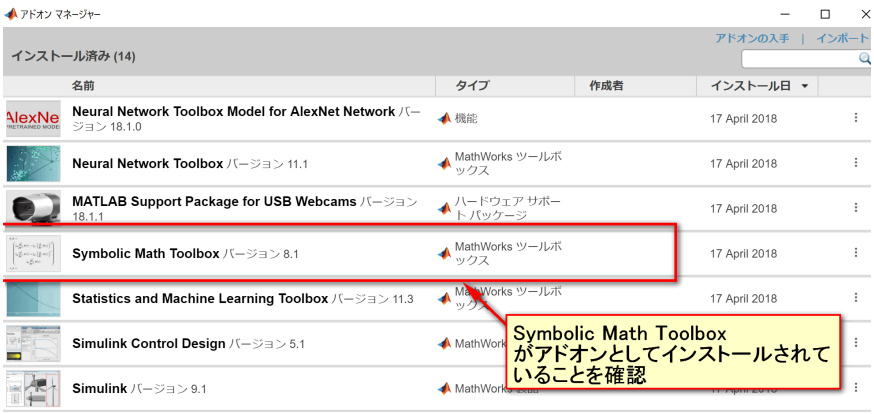

図 10.1　MATLAB アドオン確認画面

使用方法は eig 関数同様に

[変換行列，Jordan 標準形] = jordan(正方行列)

とすればよい。

前述の例と同様の V を用いて $A := VJV^{-1}$ を作り，jordan 関数を用いてみると，きちんと Jordan 標準形が求められていることが分かる。

```
>> A = V * J * inv_V

A =

   -1.5000    2.0000    3.5000
   13.5000   -3.0000  -10.5000
  -13.5000    6.0000   13.5000

>> [v, lambda] = jordan(A)

v =

   -4.5000    1.0000    0.7778
```

```
   13.5000         0         0
  -13.5000         0    1.0000


lambda =

     3     1     0
     0     3     0
     0     0     3
```

但しこれは J, V, V^{-1} に誤差が混入していない場合にのみ有効である。実際 J として

$$J = \begin{bmatrix} \sqrt{3} & 1 & 0 \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix}$$

を与えると，正しい Jordan 標準形を求めることに失敗することが分かる。

```
>> J = [sqrt(3), 1, 0; 0, sqrt(3), 0; 0, 0, sqrt(3)]

J =

    1.7321    1.0000         0
         0    1.7321         0
         0         0    1.7321

>> A = V * J * inv_V

A =

   -2.7679    2.0000    3.5000
   13.5000   -4.2679  -10.5000
  -13.5000    6.0000   12.2321

>> [v, lambda] = jordan(A)

v =

   0.3333 - 0.0000i   0.3333 + 0.0000i   6.2222 + 0.0000i
  -1.0000 - 0.0000i  -1.0000 + 0.0000i  12.2500 + 0.0000i
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i


lambda =

    1.7321         0         0
         0    1.7321         0
         0         0    1.7321
```

問題 10.2

1. 正則な正方行列 V を

$$V = \begin{bmatrix} 1 & 2 & 3 \\ -3 & -2 & -2 \\ 3 & 4 & 5 \end{bmatrix}$$

とすると，逆行列 V^{-1} は

$$V^{-1} = \begin{bmatrix} 1 & -1 & -1 \\ -9/2 & 2 & 7/2 \\ 3 & -1 & -2 \end{bmatrix}$$

となる。これを用いて次の Λ と J の場合に $A = V\Lambda V^{-1}$ と $B = VJV^{-1}$ を作り，それぞれ eig 関数でどのような結果が得られるか調べよ。また，その理由も答えよ。

$$\Lambda = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \; J = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

2. 次の方法で Jordan 標準形が求められるかどうか，上記の行列 A, B を用いて確認せよ。
   （a）行列の固有値を求め，対角化できるかどうかを確認する。
   （b）2 次以上の Jordan ブロックが現れる可能性があるかどうか，$\mathrm{rank}(A - \lambda(A)I)$, $\mathrm{rank}(B - \lambda(B)I)$ を計算して調べる。

中間レポート：べき乗法と逆べき乗法

べき乗法は最も単純な，絶対値最大固有値 $\lambda_1 = \lambda_1(A)$ とそれに属する固有ベクトル $\mathbf{v}_1$ を同時に求める方法である。これを正則行列に適用すると，絶対値最小固有値と固有ベクトルを求めることもできる。以下，全ての固有値が相異なるケースを想定して，本手法を解説する。

べき乗法

もし全ての固有値が相異なる ($i < j$ の時，$\lambda_i \neq \lambda_j$, かつ，$|\lambda_i| > |\lambda_j|$) ならば，各固有値 $\lambda_i = \lambda_i(A)$ に属する固有ベクトル $\mathbf{v}_i$ は n 次元線型空間の基底となるため，任意のベクトル $\mathbf{x}_0$ は

$$\mathbf{x}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n$$

と表現できる。従って $\mathbf{x}_k := A^k\mathbf{x}_0$ とすれば

$$\mathbf{x}_k = (\lambda_1)^k \left\{ c_1\mathbf{v}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k \mathbf{v}_2 + \cdots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^k \mathbf{v}_n \right\}$$

であるから，

$$\mathbf{x}_k = (\lambda_1)^k c_1\mathbf{v}_1 + O\left(\left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k\right)$$

となり，固有ベクトル $\mathbf{v}_1$ へ収束する。さすれば固有値 λ_1 は Reiley 商

$$\lambda_1 \approx \frac{(A\mathbf{x}_{k+1}, \mathbf{x}_k)}{(\mathbf{x}_k, \mathbf{x}_k)}$$

を計算することで得られる。実際には overflow を防ぐため，反復一回ごとに $\|\mathbf{x}_k\| = 1$ となるように正規化する。

1. 初期ベクトル $\mathbf{x}_0$(ここで $\|\mathbf{x}_0\| = 1$) を決める。
2. for $k = 0, 1, 2, \ldots$
   （a）$\mathbf{y}_{k+1} := A\mathbf{x}_k$
   （b）$\gamma_{k+1} := (\mathbf{y}_{k+1}, \mathbf{x}_k)/(\mathbf{x}_k, \mathbf{x}_k) = (\mathbf{y}_{k+1}, \mathbf{x}_k)$
   （c）収束判定
   （d）$\mathbf{x}_{k+1} := \mathbf{y}_{k+1}/\|\mathbf{y}_{k+1}\|$

このアルゴリズムに従うと，γ_k が $\lambda_1(A)$ へ，$\mathbf{x}_k$ はそれに属する固有ベクトルへと収束する。収束判定は固有値の近似値 γ_k，あるいは固有ベクトルの近似値 $\mathbf{x}_k$ を見て判断する

■べき乗法の計算例: eig_power.m　実対称行列 A を

$$A = \begin{bmatrix} 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 3 & 2 & 1 \\ 3 & 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{10.7}$$

とする。この時，MATLAB でべき乗法を実行してみよう。

```
 1: % べき乗法 : eig_power.m
 2:
 3: % 行列 A
 4: A = [
 5:     5, 4, 3, 2, 1;
 6:     4, 4, 3, 2, 1;
 7:     3, 3, 3, 2, 1;
 8:     2, 2, 2, 2, 1;
 9:     1, 1, 1, 1, 1
10: ];
11:
12: % 絶対値最大固有値の近似値
13: %absmax_eig = 0.0;
14:
15: % 絶対知最大固有値に対応する固有ベクトルの近似値
16: [row_dim, col_dim] = size(A);
17: absmax_vec = ones(col_dim, 1);
18:
19: % 停止即で使用する絶対相対許容度と相対許容度
20: abs_tol = 1.0e-50;
21: rel_tol = 1.0e-10;
22:
23: % 最大反復回数
24: max_times = max(size(A)) * 10;
25:
26: % メインループ
27: old_eig = 0;
28: absmax_eig = 0;
29: absmax_vec = absmax_vec / norm(absmax_vec);
30: for times = 1:max_times
31:     y = A * absmax_vec;
32:     absmax_eig = absmax_vec.' * y;
33:     if abs(old_eig - absmax_eig) <= rel_tol * abs(old_eig) + abs_tol
34:         break;
35:     end
36:     absmax_vec = y / norm(y);
37:     old_eig = absmax_eig;
38: end
39:
40: % 表示
41: fprintf("Maximum Eigenvalue(%d times): %25.17e\n", times, absmax_eig);
42:
43: fprintf(" i       eigenvector[i]        (A * eivenvector)[i] / eigenve
ctor[i]\n");
```

```
44: y = A * absmax_vec;
45: for i = 1:row_dim
46:     fprintf("%2d %25.17e %25.17e\n", i, absmax_vec(i), y(i) / absmax_v
ec(i));
47: end
```

これを実行すると，以下のような結果を得る。$\lambda_1(A)$ の近似値が

```
  Maximum Eigenvalue:   1.23435375196795842e+01
```

になっている時の固有ベクトルの近似値 $\mathbf{x}$，及び $A\mathbf{x}$ の各要素の $\mathbf{x}$ との比をそれぞれ出力すると

```
 i       eigenvector[i]         (A * eivenvector)[i] / eigenvector[i]
 1   2.23606797749978981e+00   1.23435375196795842e+01
 2   2.05491504837138317e+00   1.23435375196779056e+01
 3   1.70728512307438196e+00   1.23435375196750883e+01
 4   1.22134111072129303e+00   1.23435375196720223e+01
 5   6.36451305172487269e-01   1.23435375196696810e+01
```

となる。

逆べき乗法

逆べき乗法は A の代わりに A^{-1} を用いることで，A の絶対値最小固有値とそれに対応する固有ベクトルを求める方法である。正則行列 A の固有値と対応する固有ベクトルが $\lambda_i, \mathbf{v}_i$ である時,

$$A\mathbf{v}_i = \lambda_i \mathbf{v}_i \Rightarrow A^{-1}\mathbf{v}_i = \frac{1}{\lambda_i}\mathbf{v}_i$$

であることから，A^{-1} の固有値は A の固有値の逆数 $1/\lambda_i$ であることが分かる。また，固有ベクトルは変化しない。

実際に計算する際には A^{-1} を直接求めるのではなく，$\mathbf{z}_{k+1}$ を未知数とする連立一次方程式

$$A\mathbf{z}_{k+1} = \mathbf{z}_k$$

を解くことで得る。

■逆べき乗法の計算例　先の例で使った実対称行列 A ((10.7) 式) の最小固有値を逆べき乗法で求めてみる。前述のべき乗法のスクリプト `eig_power.m` を 2 行 (31 行目と 41 行目) だけ変更して絶対値最小固有値を求めてみると下記のようになる。

```
Minimum Eigenvalue(45 times):   2.71554129365498276e-01
 i       eigenvector[i]         (A * eivenvector)[i] / eigenvector[i]
 1   1.69884404803643779e-01   2.71550898002053009e-01
 2  -4.55721838828032411e-01   2.71551923956332986e-01
 3   5.96881286102514452e-01   2.71553650073526498e-01
 4  -5.48538124495066115e-01   2.71555528275313485e-01
 5   3.26029984657473293e-01   2.71556962263911017e-01
```

べき乗法の時より，反復回数が増え，精度も落ちていることが分かる。

問題 10.3

Hilbert 行列 H_5

$$H_5 = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 & 1/5 \\ 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \end{bmatrix}$$

の絶対値最大固有値 λ_1 と絶対値最小固有値 λ_5 を，べき乗法，逆べき乗法で求めよ。なお，MATLAB では H_5 を `hilb(5)` で得ることができる。