第 11 章

線型空間と固有値の応用

本章は，今まで学んできた「基底」「線型空間」「固有空間」の知識を応用し，より広い範囲の「線型問題」に対するアプローチの方法を習得する。無限数列と線型空間とはなかなかイメージとして結び付けづらいものがあるが，コンパニオン行列を具体的事例として，固有値と固有ベクトルを媒介とした線型空間における任意のベクトルの表現方法を掴み取って頂きたい。

11.1　線型変換を固有空間上で表現する

今まで学んできたように，任意の正方行列 $A \in \mathbb{C}^{n\times n}$ が対角化可能な場合は，全ての固有空間の次元数の総計は必ず n になる。即ち，$\lambda_i(A) = \lambda_i\ (i = 1, 2, ...m, m \leq n)$ に対応する固有空間 $\mathcal{V}_i = \{\mathbf{x}_i \mid A\mathbf{x}_i = \lambda_i\mathbf{x}_i\}$ において

$$\sum_{i=1}^{m} \dim(\mathcal{V}_i) = n$$

となる。

従って，それぞれの固有空間から次元数分だけ基底をなすベクトルを取ることができ，それをまとめて $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$ とすると，任意の n 次元ベクトル $\mathbf{y} \in \mathbb{C}^n$ は

$$\mathbf{y} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i\ (\alpha_i \in \mathbb{C}) \tag{11.1}$$

と表現できる。従って，この線型変換 $A\mathbf{y}$ は

$$A\mathbf{y} = \sum_{i=1}^{n} \lambda_i \alpha_i \mathbf{v}_i$$

のように，それぞれの固有ベクトルの定数倍になる。

つまり，対角化可能な正方行列 A による線型変換 (行列・ベクトル積) は，あらかじめ固有値と（一次独立な）固有ベクトルが分かっていれば，それぞれの固有ベクトルの成分の固有値倍の総和（一次結合）で求めることができる。

■対角化可能な 3 次正方行列の場合

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

とする。この時，$\mathbf{y} = [-3\ 5\ 9]^T$ に対して，$A\mathbf{y}$ を，A の固有値・固有ベクトルから求めてみよう。手順としては次のようになる

1. Λ, P を求める。この時

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix},\ P = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}$$

   である。

2. $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \alpha_3]^T$ を連立一次方程式

$$P\boldsymbol{\alpha} = \mathbf{y}$$

   を解いて求める。

3. $\sum_{i=1}^{3} \lambda_i \alpha_i \mathbf{v}_i (= A\mathbf{y})$ を計算する。

上記の手順を MATLAB を使って求めてみよう。

1. eig 関数を使って Λ, P を導出。

```
>> A = [3, 2, 1; 2, 2, 1; 1, 1, 1]

A =

     3     2     1
     2     2     1
     1     1     1

>> [P, Lambda] = eig(A)

P =

   -0.3280   -0.5910    0.7370
    0.7370    0.3280    0.5910
   -0.5910    0.7370    0.3280


Lambda =

    0.3080         0         0
         0    0.6431         0
         0         0    5.0489

>> rank(P)

ans =

     3
```

2. $\mathbf{y}$ を入力し，$P\boldsymbol{\alpha} = \mathbf{y}$ を $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \alpha_3]^T$ について解く。

```
>> y = [-3; 5; 9]

y =

    -3
     5
     9

>> alpha = P \ y

alpha =

   -0.6502
   10.0457
    3.6960

>> P * alpha

ans =

   -3.0000
```

```
    5.0000
    9.0000
```

3. $\sum_{i=1}^{3} \lambda_i \alpha_i \mathbf{v}_i (= A\mathbf{y})$ を計算する。

```
>> Lambda(1,1) * alpha(1) * P(:, 1) + Lambda(2, 2) * alpha(2) * P(:, 2)
 + Lambda(3, 3) * alpha(3) * P(:, 3)

ans =

   10.0000
   13.0000
   11.0000

>> A * y

ans =

    10
    13
    11
```

問題 11.1

3 次の Hilbert 行列 H_3 と $\mathbf{w} = [1\ 2\ 3]^T$ に対して，$H\mathbf{w}$ を上記の手順で求めよ。また，$H\mathbf{w}$ を陽に計算してこれを真値として，上記の手順で求めたベクトルの誤差を評価せよ。

11.2 代数方程式を固有値問題として解く方法

本章の最初に述べたように，行列の固有値を求める問題は，本質的に代数方程式 (固有方程式)

$$|A - \lambda I| = 0$$

の解を求める問題と同じである。では逆に，最高次数の係数が 1 である n 次多項式 $p(\lambda)$ が

$$p(\lambda) = \lambda^n + c_{n-1}\lambda^{n-1} + c_{n-2}\lambda^{n-2} + \cdots + c_1\lambda + c_0 \ (c_i \in \mathbb{C})$$

と与えられる時の代数方程式

$$p(\lambda) = 0 \tag{11.2}$$

が与えられたとき，これを固有方程式と見立てることの行列を作ることができれば，既に述べた行列の固有値を求める MATLAB スクリプトを用いてこの代数方程式の解を求めることもできることになる。結論から言うと，このような固有多項式を持つ行列は下記のようになる。

$$C_n = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ -c_0 & -c_1 & \cdots & \cdots & -c_{n-2} & -c_{n-1} \end{bmatrix} \tag{11.3}$$

これをコンパニオン行列 (companion matrix) と呼ぶ。この時，

$$|C_n - \lambda I| = |(-I)(\lambda I - C_n)| = (-1)^n |\lambda I - C_n| = (-1)^n p(\lambda)$$

となる。よって，$p(\lambda) = 0$ が与えられれば，対応するコンパニオン行列を自動的に得ることができる。

例えば

$$x^3 - 6x^2 + 11x - 6 = 0$$

という代数方程式に対しては

$$C_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 6 & -11 & 6 \end{bmatrix}$$

というコンパニオン行列 C_3 が対応する。もし最高次数の係数 c_n が 1 でなければ，全ての係数を c_n で割ってからコンパニオン行列を作ればよい。

これを利用すると，代数方程式の係数が与えられれば，対応するコンパニオン行列を作り，その固有値を eig 関数から求めて代数方程式の解とすればよい。

これを eig_algebraic_eq.m という MATLAB スクリプトにしてみよう。

```
 1: % 次元数: n
 2: str = ' 代数方程式の次数 n を入力してください：n = ';
 3: n = input(str);
 4: fprintf("次数 (n) = %d", n);3
 5: disp("代数方程式の係数 (c(0), c(1), ..., c(n-1)) を入力してください：");
 6: c = []; % 添え字 1 から開始
 7: for i = 1 : n
 8:     str = sprintf('c(%d) = ', i - 1); c(i) = input(str);
 9: end
10:
11: % コンパニオン行列 C
12: C = zeros(n, n);
13: for i = 1 : n
14:     C(n, i) = -c(i);
15: end;
16:
17: for i = 1 : n - 1
18:     C(i, i + 1) = 1;
19: end
20: % 計算時間 tic -> tac
21: tic;
22: % 固有値計算
23: lambda = eig(C);
24: % 計算時間取得
25: time_spec = toc;
26: fprintf( "固有値計算 (秒) = %f\n", time_spec);
27:
28: % 固有値出力
29: disp('eig(C) = '); disp(lambda);
```

先の例を入力してみると，確かに代数方程式の解が得られていることが分かる。

```
 代数方程式の次数 n を入力してください：
-->3
次数 (n) = 3
 代数方程式の係数 (c(0), c(1), ..., c(n-1)) を入力してください：
c(0) =
-->-6
c(1) =
-->11
c(2) =
-->-6

 固有値計算 (秒) =

    0.001

 eig(C) =

    1.
    2.
    3.
```

問題 11.2

次の代数方程式の解を eig_algebraic_eq.m を用いて求めよ。

1. $x^3 + x^2 + x + 1 = 0$
2. $x^{10} - 1 = 0$
3. $x^5 + 4x^4 + 3x^3 + 2x^2 + 1 = 0$

11.3　コンパニオン行列の応用：線型漸化式によって定義される数列の一般項

前述したように，数列 $\{x_n\}_{n=0}^{\infty}$ が線型漸化式

$$x_{n+m} := -c_m x_{n+m-1} - c_{m-1} x_{n+m-2} - \cdots - c_1 x_n \ (c_i \in \mathbb{R} \text{ は定数})$$

を満足している時，この漸化式によって定義される数列 $x = \{x_n\}_{n=0}^{\infty}$ を要素とする集合 X_m は m 次線型空間となる。実際，任意の $x, y = \{y_n\}_{n=0}^{\infty} \in X_m$ に対して，加算とスカラー倍を

$$x + y = \{x_n + y_n\}_{n=0}^{\infty},\ \alpha x = \{\alpha x_n\}_{n=0}^{\infty}$$

と定義すると，線型空間 (定義 8.1) の性質をすべて満足することが分かる。

では，任意の線型漸化式 (8.3) によって生成される数列 $x = \{x_n\}_{n=0}^{\infty}$ の一般項はどのように表現できるか？　これを固有値・固有ベクトルを使って求める方法を考える。

11.3.1　$m = 1, 2$ の場合

もっとも簡単な X_1, 即ち $m = 1$ の場合を考えよう。この場合は例えば

$$x_{n+1} := -3x_n$$

という漸化式になるので，初期値 x_0 が決まれば任意の x_n (一般項) は

$$x_n = (-3)^n x_0$$

となることはすぐにわかる。

では $m = 2$ の時，例えば

$$x_{n+2} := -3x_{n+1} - 2x_n \tag{11.4}$$

の時はどうなるだろう？　もちろん，初期値としては x_0, x_1 がセットで決まらなければ任意の x_n を決定することはできないことになる。

この場合は次のように考える。まず $(x_0, x_1) = (1, 0)$ のケースの場合は $x_2 = (-2)x_0 = -2$, $x_3 = (-3)x_2 = 6$ となり，以後の数列が決定される。この数列を $e_1 = \{1, 0, -2, 6, ...\}$ と置く。同様に，$(x_0, x_1) = (0, 1)$ のケースに作られる数列を $e_2 = \{0, 1, -3, 7, ...\}$ と置く。

写像 $\varphi : X_2 \to \mathbb{R}^2$ を

$$\varphi(x) = \varphi(\{x_0, x_1, ..., x_n, ...\}) = [x_0\ x_1]^T$$

とすると

$$\varphi(e_1) = [1\ 0]^T,\ \varphi(e_2) = [0\ 1]^T$$

であるから，任意の数列 $x = \{x_0, x_1, ..., x_n, ...\}$ に対しては

$$\varphi(x) = x_0\varphi(e_1) + x_1\varphi(e_2) = x_0\mathbf{e}_1 + x_1\mathbf{e}_2 \in \mathbb{R}^2$$

と見ることができる。

今，写像 T を，数列 $x = \{x_0, x_1, ..., x_n, ...\}$ から漸化式 (11.4) に基づいて一つ後ろにずらした数列 $\{x_1, x_2, ..., x_{n+1}, ...\}$ に対応させるものとする。さすれば，これに対応する $\mathbb{R}^2$ 上における線型変換は

$$\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

になることは自明である。つまり

$$\begin{array}{ccccc} x = \{x_0, x_1, ..., x_n, ...\} & \longrightarrow T \longrightarrow & & & T(x) = \{x_1, x_2, ..., x_{n+1}, ...\} \\ \downarrow & & & & \uparrow \\ \varphi & & & & \varphi^{-1} \\ \downarrow & & & & \uparrow \\ \varphi(x) = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} & \longrightarrow C_2 \longrightarrow & C_2\varphi(x) = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} & = & \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{array} \tag{11.5}$$

という流れになる。もし C_2 の固有値 λ_1, λ_2 が異なれば対角化可能，即ち，それぞれの固有値に対する固有ベクトル $\mathbf{v}_1, \mathbf{v}_2$ が一次独立となるので，行列 $P = [\mathbf{v}_1\ \mathbf{v}_2]$ は正則行列となるから，連立一次方程式

$$P\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \varphi(x)$$

の解 $[\alpha_1\ \alpha_2]^T$ は一意に定まる。よって，任意の数列は

$$\varphi(x) = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2$$

と表現できる。即ち

$$C_2\varphi(x) = \lambda_1(\alpha_1\mathbf{v}_1) + \lambda_2(\alpha_2\mathbf{v}_2)$$

となるので，任意の数列 x は

$$\begin{aligned} T(x) = \varphi^{-1}(C_2\varphi(x)) &= \varphi^{-1}(\lambda_1\alpha_1\mathbf{v}_1) + \varphi^{-1}(\lambda_2\alpha_2\mathbf{v}_2) \\ &= \lambda_1\alpha_1\varphi^{-1}(\mathbf{v}_1) + \lambda_2\alpha_2\varphi^{-1}(\mathbf{v}_2) \\ &= \lambda_1\alpha_1 v_1 + \lambda_2\alpha_2 v_2 \end{aligned}$$

と表現できる。ここで v_1, v_2 は C_2 の固有ベクトルと対応付けされる数列で，$v_1 = \{v_n^{(1)}\}_{n=0}^{\infty} = \varphi^{-1}(\mathbf{v}_1)$, $v_2 = \{v_n^{(2)}\}_{n=0}^{\infty} = \varphi^{-1}(\mathbf{v}_2) \in X_2$ である。

さすれば

$$T(x) = \{x_1, x_2, ..., x_{n+1}, ...\} = \{\lambda_1\alpha_1 v_0^{(1)} + \lambda_2\alpha_2 v_0^{(2)}, \lambda_1\alpha_1 v_1^{(1)} + \lambda_2\alpha_2 v_1^{(2)}, ..., \lambda_1\alpha_1 v_n^{(1)} + \lambda_2\alpha_2 v_n^{(2)}, ...\}$$

と表現できる。これを繰り返すことで

$$\begin{aligned} T^n(x) = T(T(\cdots T(x)\cdots)) &= \varphi^{-1}(C_2^n\varphi(x)) \\ \Rightarrow \{x_n, x_{n+1}, ...\} &= \{\lambda_1^n\alpha_1 v_0^{(1)} + \lambda_2^n\alpha_2 v_0^{(2)}, \lambda_1^n\alpha_1 v_1^{(1)} + \lambda_2^n\alpha_2 v_1^{(2)}, ...\} \end{aligned} \tag{11.6}$$

となるので，一般項 x_n は

$$x_n = \lambda_1^n\alpha_1 v_0^{(1)} + \lambda_2^n\alpha_2 v_0^{(2)} \tag{11.7}$$

と表現できる。実際 (11.7) の場合，

$$\lambda_1 = -1, \mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix},\ \lambda_2 = -2, \mathbf{v}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix} \Rightarrow P = [\mathbf{v}_1\ \mathbf{v}_2] = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \Rightarrow P^{-1} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

より，

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = P^{-1}\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 2x_0 + x_1 \\ x_0 + x_1 \end{bmatrix}$$

となり，かつ $v_1 = \varphi^{-1}(\mathbf{v}_1) = \{1, -1, ...\}$, $v_2 = \varphi^{-1}(\mathbf{v}_2) = \{-1, 2, ...\}$ となるから，一般項 (11.7) は

$$x_n = (-1)^n(2x_0 + x_1) - (-2)^n(x_0 + x_1) \tag{11.8}$$

と表わすことができる。

■linear_diff.m スクリプト　2 項線型漸化式の係数 c_0, c_1，初期値 x_0, x_1 が与えられた時，一般項を計算する MATLAB スクリプト (linear_diff.m) を下記に示す。

```
 1: clear;
 2:
 3: %  線型漸化式
 4: %  x_{n+2} := c1 * x_{n+1} + c0 * x_n
 5: %  c = [c0; c1]
 6: c0 = -2;
 7: c1 = -3;
 8:
 9: %  初期値
10: %  x = [x0; x1]
11: x0 = 1; x1 = 0;
12: % x0 = 0; x1 = 1;
13:
14: %  数列
15: c = [c0; c1];
16: x = [x0; x1];
17: fprintf("漸化式: x_{n+2} := (%g) * x_{n+1} + (%g) * x_n\n", c(2), c(
1));
18: fprintf("初期値: x0 = %g, x1 = %g\n", x(1), x(2));
19:
20: %  Companion Matrix
21: C = [0, 1; c(1), c(2)];
22: disp("C = "); disp(C);
23:
24: [P, lambda] = eig(C);
25: disp("P = "); disp(P);
26: disp("lambda = "); disp(lambda);
27:
28: %  alpha
29: alpha = inv(P) * x;
30:
31: %  一般項
32: fprintf("\n一般項：\n");
33: fprintf("xn = ((%g) + (%g) * i)^n * ((%g) + (%g) * i) * ((%g) + (%g)
 * i) + ((%g) + (%g) * i)^n * ((%g) + (%g) * i) * ((%g) + (%g) * i)\n",
real(lambda(1, 1)), imag(lambda(1, 1)) , real(alpha(1)), imag(alpha(1)),
 real(P(1, 1)), imag(P(1, 1)), real(lambda(2, 2)), imag(lambda(2, 2)), r
eal(alpha(2)), imag(alpha(2)), real(P(1, 2)), imag(P(1, 2)));
34: fprintf("\n");
35:
36: %  検算
37: fprintf(" n        漸化式計算        一般項計算\n");
38: for n = 0:10
39:     fprintf("%2d %15.7g %15.7g\n", n, seq_n(n, c, x), lambda(1, 1)^n
* alpha(1) * P(1, 1) + lambda(2,2)^n * alpha(2) * P(1, 2));
40: end
41:
42: %  漸化式に基づく数列の第 n 項計算
43: %  x_{n+2} := c1 * x_{n+1} + c0 * x_n
44: function xn = seq_n(n, c, x)
45:     temp = [];
46:     temp(1) = x(1);
47:     temp(2) = x(2);
48:
49:     if n == 0
50:         xn = temp(1);
51:     elseif n == 1
52:         xn = temp(2);
53:     else
54:         for i = 2:n
```

```
55:             xn = c(2) * temp(2) + c(1) * temp(1);
56:             temp(1) = temp(2);
57:             temp(2) = xn;
58:         end
59:     end
60: end
```

■実行結果　上記の linear_diff.m を実行すると下記のような結果を得る。漸化式に基づく x_2〜x_{10} の値と，一般項の定義式に基づく値が一致していることが分かる。

```
漸化式: x_{n+2} := (-3) * x_{n+1} + (-2) * x_n
初期値: x0 = 1, x1 = 0
C =
     0     1
    -2    -3

P =
    0.7071   -0.4472
   -0.7071    0.8944

lambda =
    -1     0
     0    -2
```

一般項：

```
xn = ((-1) + (0) * i)^n * ((2.82843) + (0) * i) * ((0.707107) + (0) * i)
+ ((-2) + (0) * i)^n * ((2.23607) + (0) * i) * ((-0.447214) + (0) * i)
```

n	漸化式計算	一般項計算
0	1	1
1	0	0
2	-2	-2
3	6	6
4	-14	-14
5	30	30
6	-62	-62
7	126	126
8	-254	-254
9	510	510
10	-1022	-1022

問題 11.3

1. 線型漸化式 (11.4) によって規定される数列 $x \in X_2$ に対して次の問いに答えよ。
   （a）$T(v_1) = \lambda_1 v_1, T(v_2) = \lambda_2 v_2$ であることを確認せよ。
   （b）一般項 x_n が (11.8) として表現できることを数学的帰納法を用いて答えよ。
2. $x_0 = x_1 = 1$ という初期値を取る時，線型漸化式

$$x_{n+2} = x_{n+1} + x_n$$

   で規定される数列をフィボナッチ数列と呼ぶ。この一般項 x_n を求めよ。

中間レポート 2: 一般の斉次線型漸化式の一般解

X_m の場合

以上見てきたように，線型漸化式 (8.3) の一般項 x_n は，写像 T と $\varphi : X_m \to \mathbb{R}^m$,

$$\varphi(x) = \varphi(\{x_0, x_1, ..., x_{m-1}, ...\}) = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{m-1} \end{bmatrix}$$

そしてコンパニオン行列 C_m

$$C_m = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ -c_1 & -c_2 & \cdots & \cdots & -c_{m-1} & -c_m \end{bmatrix}$$

が対角化可能である場合，(11.5) 同様，次のような写像の構造ができる。

$$\begin{array}{ccccc} x = \{x_0, x_1, ..., x_n, ...\} & \longrightarrow T \longrightarrow & T(x) = \{x_1, x_2, ..., x_{n+1}, ...\} \\ \downarrow & & \uparrow \\ \varphi & & \varphi^{-1} \\ \downarrow & & \uparrow \\ \varphi(x) = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{m-1} \end{bmatrix} & \longrightarrow C_m \longrightarrow & C_m\varphi(x) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \end{array} \tag{11.9}$$

従って，次のようにして一般項 x_n を求めることができる。

1. 初期値 $\mathbf{x}_0 = [x_0\ x_1\ ...\ x_{m-1}]^T$ を定める。
2. C_m の固有値 $\lambda_1, \lambda_2, ..., \lambda_m$ と対応する固有ベクトル $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m$ を求め，$v_1 = \varphi(\mathbf{v}_1) = \{v_0^{(1)}, ...\}$, $v_2 = \varphi(\mathbf{v}_2) = \{v_0^{(2)}, ...\}$, ..., $v_m = \varphi(\mathbf{v}_m) = \{v_0^{(m)}, ...\}$ を得る。
3. $P = [\mathbf{v}_1\ \mathbf{v}_2\ ...\mathbf{v}_m]$ を作り，連立一次方程式 $P\alpha = \mathbf{x}_0$ を解いて $\alpha = [\alpha_1\ \alpha_2\ ...\ \alpha_m]^T$ を求める。
4. $x_n = \lambda_1^n\alpha_1 v_0^{(1)} + \lambda_2^n\alpha_2 v_0^{(2)} + \cdots + \lambda_m^n\alpha_m v_0^{(m)}$ を得る。

問題 11.4

1. $T(x)$ が線型変換，即ち，任意の定数 α, β および $x, y \in X_m$ に対して，

$$T(\alpha x + \beta y) = \alpha T(x) + \beta T(y)$$

であることを示せ。

2. 次の線型漸化式に対して初期値 $x_0 = x_1 = \cdots = x_{m-1} = 1$ を与えた時に一般項を求める MATLAB プログラムを作れ。

   (a) $x_{n+3} = -6x_{n+2} + 11x_{n+1} - 6x_n$

   (b) $x_{n+4} = -5x_{n+3} - 7x_{n+2} - 9x_{n+1} + x_n$

   (c) $x_{n+5} = 4x_{n+4} + 2x_{n+3} - 5x_{n+2} + x_{n+1} + 3x_n$

参照用として，m 次線型漸化式の一般項表現を求め，検算を行う MATLAB スクリプトを以下に示す。

```
% m 次線型漸化式の一般項表現の導出
clear;

% (a) x_{n+3} := 6x_{n+2} + 11x_{n+1} ? 6x_n
m = 3; c = [-6, 11, -6]; x_init = [1; 1; 1];

% コンパニオン行列生成
C = zeros(m);
for i = 1:m
    C(m, i) = c(i);
end;
for i =1:m-1
    C(i, i + 1) = 1; % 対角成分の上を 1 に
end;

disp("C = "); disp(C);

% 固有値と固有ベクトル
[P, Lambda] = eig(C);
disp("P = "); disp(P);
disp("Lambda = "); disp(Lambda);

% P * alpha = x_init
alpha = inv(P) * x_init;

% 検算 x == x_init?
x = zeros(m,1);
for i = 1:m
    x = x + alpha(i) * P(:, i);
end
disp("x = "); disp(x);
disp("x_init = "); disp(x_init);

% 一般項の表示
fprintf('一般項: x_n :=((%15.7e) + (%15.7e) * i)^n * ((%15.7e) +
(%15.7e) * i) * ((%15.7e) + (%15.7e) * i) ', real(Lambda(1, 1)), imag
(Lambda(1, 1)), real(alpha
(1)), imag(alpha(1)), real(P(1, 1)), imag(P(1, 1)) );
for i = 2:m
    fprintf(' + ((%15.7e) + (%15.7e) * i)^n * ((%15.7e) + (%15.7e)
 * i) * ((%15.7e) + (%15.7e) * i) ', real(Lambda(i, i)), imag(Lambda(i,
 i)), real(alpha(i)), imag(alpha(i)), real(P(1, i)), imag(P(1, i)) );
end
fprintf("\n");

% 検算
for i = 1:50
    xn_true = seq_n(i, c, x, m); % 漸化式で計算
    xn_appr = general_n(i, alpha, Lambda, P, m); % 一般項の式で計算
    fprintf("%3d: %15.7e, %15.7e, %5.1e\n", i, xn_true, xn_appr, re
lerr(xn_app
r, xn_true));
end

% 相対誤差
function ret = relerr(approx, true_val)
    ret = abs(approx - true_val);
    if(abs(true_val) >= 1.0e-308)
```

```
51:         ret = ret / abs(true_val);
52:     end
53: end
54:
55: %  一般項表現に基づく数列の第 n 項計算
56: %  x_n := \sum^{m}_{k = 1} \lambda_k^n * alpha(k) * v^(k)_0
57: function xn = general_n(n, alpha, Lambda, P, m)
58:     xn = 0;
59:     for i = 1:m
60:         xn = xn + Lambda(i, i)^n * alpha(i) * P(1, i);
61:     end
62: end
63:
64: %  漸化式に基づく数列の第 n 項計算
65: function xn = seq_n(n, c, x, m)
66:     temp = [];
67:     for i = 1:m
68:         temp(i) = x(i);
69:     end
70:
71:     if n < m
72:         xn = temp(n);
73:     else
74:         for i = m:n
75:             xn = 0;
76:             for i = 1:m
77:                 xn = xn + c(i) * temp(i);
78:             end
79:             for i = 1:m - 1
80:                 temp(i) = temp(i + 1);
81:             end
82:             temp(m) = xn;
83:         end
84:     end
85: end
```