Chapter 4

力学系とプログラミング基礎

4.1 はじめに

力学系とは、ある規則に従って時間発展する変数からなる系 (システム) のことである。英語では dynamical system と呼ばれる。最も簡単な力学系は、前回講義した数列および漸化式である。各項 a_k は時間を離散化して、時刻 k における変数 a の値であると考えればよい。

ここでは、差分方程式で記述された離散力学系と微分方程式で記述された連続力学系をこの順に紹介する。なお、離散か連続かは、時間の扱い方によるもである。

4.2 離散力学系

典型的な離散力学系は漸化式で記述された数列である。ただし、場合分けや複雑な関数を用いることがよくある。

4.2.1 簡単な力学系

まず手始めに、次の式で記述された力学系を Octave でプログラムしてみよう。

$$x_{k+1} = ax_k(1 - x_k) \tag{4.1}$$

これは、ロジスティックマップと呼ばれるものである。詳しくは後述する。a はパラメータで、[0,4] の範囲で変化する。繰り返しの回数を n とする。また、$x_0 = 0.01$ とする。数列を作る部分をまとめて関数 logistic() として、これを定義する”logistic.m” というプログラムを入力しよう。

```
#logistic.m
function L=logistic(x0,a,n)
  L=zeros(n,1);
  x=x0;
  for(i=1:n)
    L(i,1)=x;
    x = a*x*(1-x);
  endfor
endfunction

#initialise
x0=0.04;
```

実行例

```
octave:2> source("logistic.m")
octave:3> X=logistic(x0,3.0,100);
```

これで、X に 100 個の数列が代入された。これは数値のままみるよりは、プロットした方がわかりやすい。

```
octave:>plot(X)
```

を実行し、どのような振る舞いになるか見てみよう。パラメータ a の値を変えて試してみよう。

練習問題

a の値を 2.9 3.0, 3.3, 3.5, 3.7, 3.85, 3.9 として、それぞれ 100 個の数列を求め、プロットせよ。

4.2.2 ロジスティックマップ

式 4.1 はロジスティックマップと呼ばれる力学系である。非常に単純な力学系ではあるが、カオスと呼ばれる予測不能で複雑な振る舞いを示す特殊な力学系でもある。マップとは、写像のことである。写像とは、ある関数によって、ある点を別の点に写すことである。英語でいう mapping に相当し、単に「地図を作る」というよりも「あるモノを、別の空間のモノに置き換える」作業である。

もう一度式を見直してみよう。

$$x_{k+1} = ax_k(1 - x_k) \tag{4.2}$$

右辺は 2 次方程式である。重要なことは、この写像は「線形性が成り立たない」ということである。線形性とは、ある写像について

$$f(x+y) = f(x)+f(y) \tag{4.3}$$
$$f(\alpha x) = \alpha f(x) \tag{4.4}$$

という関係が成立することである (α は定数)。ここでは、「線形であれば、2 点間の差は大きく増大しない」ということが重要である。例えば 2 次関数であれば、$f(x+\delta x)$ を計算してみればわかるように、差は 2 倍に増幅される。毎回 2 倍に増幅されると仮定すると、差は指数関数的に増幅する (前回の複利計算を思い出そう)。ただし、x は [0,1] に閉じ込められているので、近い 2 点から出発した軌道 (x_n の列) は近寄ったり離れたり、と複雑になる。

これをプロットしてみよう。まず以下の関数を定義する。(logiShape.m とする)

```
# plotLogi
function LS=logishape(a)
  n=1;
  for(x=0:0.01:1)
    LS(n,:)=[x,a*x*(1-x)];
    n=n+1;
  endfor
endfunction
```

実行例

```
octave:> source("logishape.m");
octave:> LS1=logishape(3.5);
octave:> plot(LS1(:,1),LS1(:,2));
```

ロジスティックマップの振る舞いは、作図によって求めることができる。

```
octave:> LS1=logishape(3.5);
octave:> plot(LS1(:,1),LS1(:,2));
octave:> diagline=[0,0;1,1];
octave:> hold on
octave:> plot(diagline(:,1),diagline(:,2))
```

“hold on” というコマンドは、プロットを重ね書きする指令である。(解除して、新しい図を描くには hold off コマンドを使う)。

作図による求め方は以下の通りである。

- 横軸を x_n、縦軸を x_{n+1} とする。

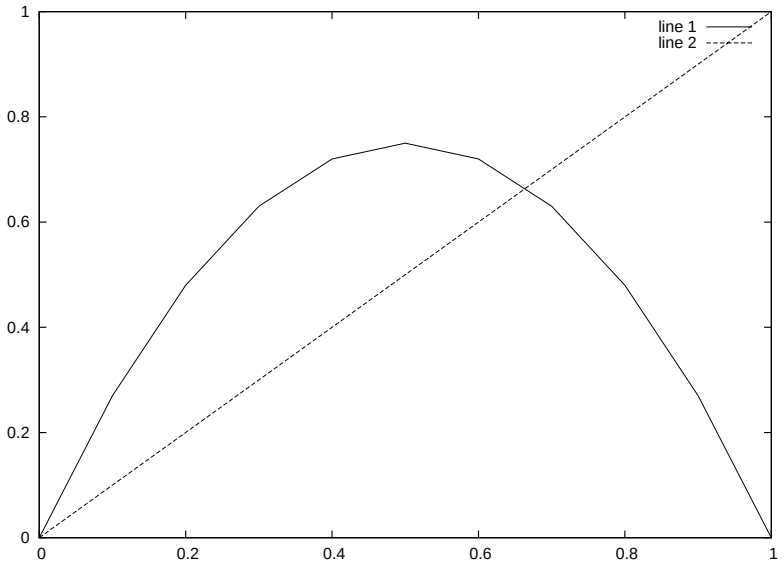

Figure 4.1: ロジスティックマップの形。a=3.5

- x_0(初期値) を横軸上にとる。
- x_0 から垂直に上に線を引き、放物線に当たるまで引く。
- 放物線の交点の「高さ」が x_1 である。
- 今度は水平に、対角線に交わる水平線を引く。
- これで x_1 を横軸に写し終えた。垂直線と放物線の交点が、x_2 となる。
- 以下繰り返し。

このように、 x_n と x_{n+1} を両軸にとる作図方法をリターンマップと呼ぶ。これに対し、上で求めたような x_n を時刻 (順番) を横軸にとって表示する方法を時系列と呼ぶ。

プログラムで書くと、以下のようになる。

```
#logidraw.m
n=100;
x0=0.1;
a=3.5;
X=logistic(x0,a,n);
P=logishape(a);
```

```
diagline=[0,0;1,1];
hold off;
plot(P(:,1),P(:,2));
hold on;
plot(diagline(:,1),diagline(:,2));
XX(1,1)=X(1,1);
XX(1,2)=0;
XX(2,1)=X(1,1);
XX(2,2)=X(2,1);
XX(3,1)=X(2,1);
XX(3,2)=X(2,1);
for(i=2:n)
  XX(i*2,1)=X(i-1,1);
  XX(i*2,2)=X(i,1);
  XX(i*2+1,1)=X(i,1);
  XX(i*2+1,2)=X(i,1);
endfor
plot(XX(:,1),XX(:,2));
hold off;
```

練習問題

上のプログラムを変更し、様々な a の値についてロジスティックマップの振る舞いを調べ、分類せよ。必要ならば n を大きくするとよい。
(お勧めの a 値:2.9 3.0, 3.3, 3.5, 3.7, 3.85, 3.9)

なお、プログラム終了後も a の値は保存されるので、x の時系列を表示するには、

```
octave-3.0.0:102> plot(logistic(x0,a,100))
```

とすればよい。

4.2.3 レポート問題 7

上の練習問題を実行し、プログラムと実行結果をレポートにせよ。
(上級) プログラムを改造し、初期値を僅かに変えた 2 点の振る舞いを調べ、上の分類と比較せよ。例えば、x0=0.01, y0=0.011 とし、XX, YY という 2 つの変数にロジスティックマップの結果を格納し、比較する。

4.2.4 付録:ロジスティックマップの分岐図

図 4.2 に横軸に a、縦軸に X の値 (ただし、初期のステップを捨てたもの) をプロットした。このように表示することで、収束するか、乱雑になるかの振る舞いのパターンを把握することが出来る。詳しく知りたい向きは、

山口昌哉「カオスとフラクタル–非線形の不思議」講談社ブルーバックス 1986

等を読むとよい。

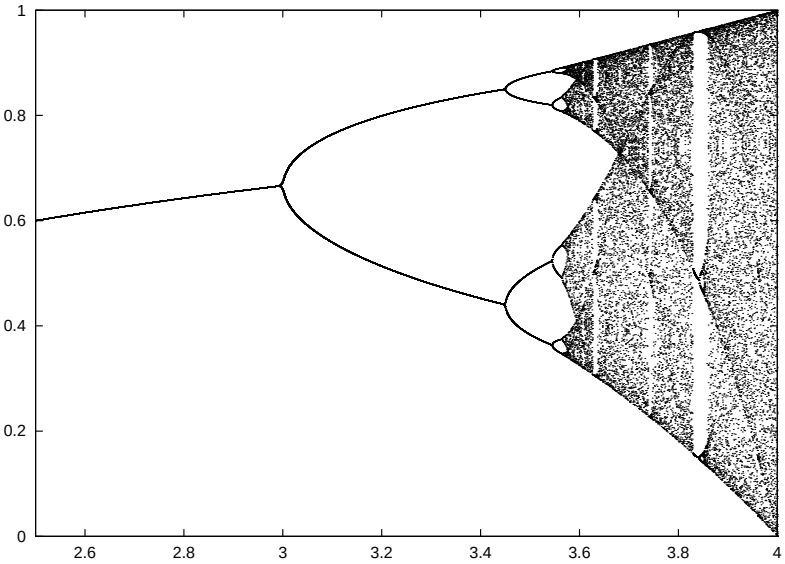

Figure 4.2: ロジスティックマップの分岐図。

4.2.5 力学系の安定性

ロジスティックマップの特徴として、a が小さいところ ($a \leq 3$) では 1 点に収束し、大きくなるにつれ 2,4,,,, と 2^n 周期が現れ、3.6 付近では複雑な軌道になる、ということがわかった。なぜ、このようなことが起こるのか考えてみよう。

まず、a が小さいとき最初にあらわれる収束点を考えよう。作図で求めた振る舞いでは、放物線と対角線の交点に落ち込んでいくことがわかる。これを「固定点 (fixed point)」と呼ぶ。この点は $x = f(x)$ の解として求めることができる。

練習問題

この点に軌道が引き込まれていくのはなぜか？まず作図で考えよ。

放物線と対角線の交点は $0, \dfrac{a-1}{a}$ の 2 点である。そのうち、固定点は後者である。前者では、初期値を 0 にとらない軌道はすべてそこから離れていってしまう。(a が 1 以下なら 0 にも引き込まれるが、ここではあまり小さい a の値は考慮しないことにする)

作図では、交点でのグラフの「傾き」が重要であることがみてとれた筈である。つまり、交点での放物線の傾きの絶対値が「　」より大きければ軌道は離れ、それより小さければ軌道はその点に吸い込まれる。ロジスティックマップの傾きは導関数で与えられ、

$$f'(x) = -2ax + a \tag{4.5}$$

である。$a > 1$ であれば、$f'(0) > 1$ である。では、もう一つの交点 $\dfrac{a-1}{a}$ ではどうか？傾きが負であることに注意すると、

$$-2ax + a < -1 \tag{4.6}$$

の不等式を解けばいいことになる。

練習問題

x に $\dfrac{a-1}{a}$ を代入し、上の不等式を解け。

$a > 3$ 付近では、軌道は周期 2 の軌道に落ち込む。これは、合成関数 $f \circ f($ f(f(x)) のことである) と対角線の交点を調べればよい。詳細を省くが、$f \circ f$ の作り方は以下のプログラム logishape2.m を参照のこと。

```
# logistic f(f(x))
function LS2=logishape2(a)
  loop=2;
  n=1;
  for(x=0:0.01:1)
```

```
    x2=x;
    for(l=1:loop)
      x2 = a*x2*(1-x2);
    endfor
    LS2(n,:)=[x,x2];
    n=n+1;
  endfor
endfunction
```

このプログラムで重要なのは、内側の for ループである。x に f(x) の値を代入し、さらにもう一度 x=f(x) の代入を行っている。すなわち f(f(x)) の作業をしていることになる。これで、loop の値を変えれば、2^n 周期の解析も可能になる。

実行してみよう。

```
octave:> source("logishape2.m");
octave:> LS2=logishape(3.3);
octave:> diagline=[0,0;1,1];
octave:> hold on
octave:> plot(LS2(:,1),LS2(:,2));
octave:> plot(diagline(:,1),diagline(:,2))
```

グラフの交点が増え、新たに現れた交点の傾きを調べてみよう。また、a の値を変えて 4 周期になるところでは、大きくなっているだろうか。

詳細は配布した資料を参照してほしいが、重要な点は、**固定点の性質を調べることで振る舞いを理解できる**ことである。

4.3 行列の固有値–力学系の安定性を例として

4.3.1 2 次元の力学系

次に、2 次元の力学系を考える。今度は、安定性の解析をするのが主眼なので、線形の力学系を用いる。これは、一次変換と呼ばれる。一次変換の一般形は、

$$x_{n+1} = ax_n + by_n \tag{4.7}$$

$$y_{n+1} = cx_n + dy_n \tag{4.8}$$

つまり、

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \tag{4.9}$$

という 1 次変換を繰り返したものであると理解しておけばよい。行列の中身によっては、もっと複雑な演算になる場合もあるが、形は変わらない。

1 次変換を繰り返し適用してみよう。まず、適当な行列を定義する。繰り返し適用は、指数関数的な性質を持つため、あまり大きな値を入れないようにすることに注意。

$$\begin{pmatrix} 0.5 & 0.2 \\ 0.3 & 0.8 \end{pmatrix} \tag{4.10}$$

この行列に対し、初期値 x0=[0.3; 0.5] から出発して 100 回変換を繰り返すプログラムが下のものである。(tr1.m とする)

```
#tr1.m
X0=[0.3;0.5];
M1=[0.5,0.2;0.3,0.8];
[v1,l1]=eig(M1);
n=100;
X=zeros(n,2);
X(1,:)=X0(1,:)';
XX=X0;
for(i=2:n)
  XX = M1*XX;
  X(i,:)=XX';
endfor
```

練習問題

上のプログラム tr1.m を作成し、実行せよ。X をプロットせよ。プロットの仕方は

```
octave:> plot(X(:,1),X(:,2),'+')
```

とするとプラス記号の点列となる。

4.3.2 行列の固有値

前節の例では、点列が原点に落ち込んでいくのがみてとれたはずである。1 次変換の行列の特徴から力学系として得られる軌跡の性質を予測できないだろうか?

行列には、固有値・固有ベクトルという概念がある。ある行列 M に対し、

$$\lambda \mathbf{x} = M\mathbf{x} \tag{4.11}$$

という関係を満たすスカラー量 (実数または複素数)λ, ベクトル $\mathbf{x}$ が求められたとき、前者を固有値、後者を固有ベクトルと呼ぶ。**この重要さは、1 次変換の線**

形性から、固有値と固有ベクトルにより M の振る舞いが理解できるという点にある。

NxN 型の正方行列には、最大で N 個の固有値がある (固有値が存在しない場合もある)。簡単のために 2x2 型とすると、固有値 λ_1、λ_2 と固有ベクトル $\mathbf{x}_1$、$\mathbf{x}_2$ が存在し、線形性から

$$M(a\mathbf{x}_1 + b\mathbf{x}_2) = aM\mathbf{x}_1 + bM\mathbf{x}_2 = a\lambda_1\mathbf{x}_1 + b\lambda_2\mathbf{x}_2 \tag{4.12}$$

であることがわかる。また、$\mathbf{x}_1 \neq \alpha\mathbf{x}_2$ である (「一次独立」である) から、任意のベクトル $\mathbf{y}$ について、

$$\mathbf{y} = c\mathbf{x}_1 + d\mathbf{x}_2 \tag{4.13}$$

の形に置き換える、つまり**座標変換すること**が可能である。このことから、固有ベクトルによって振る舞いが規定されることがわかった。

Octave では、固有値と固有ベクトルは eig() によって同時に求めることができる。eig() の振る舞いは変わっていて、

```
(octave:> M1=[0.5,0.2;0.3,0.8]; #tr1.m を実行していない場合は定義する)
M1 =

  0.50000  0.20000
  0.30000  0.80000

octave:2> eig(M1)
ans =

  0.36277
  0.93723

octave:3>
```

左辺に何もない、または変数が一つの場合は固有値のみが出力され、

```
octave:3> [v,l]=eig(M1)
v =

  -0.82456  -0.41597
   0.56577  -0.90938

l =
```

```
  0.36277   0.00000
  0.00000   0.93723
```

1x2 型の行列の形で変数を与えると、固有ベクトルと固有値が出力される。l の対角成分が固有値、v の各列が固有ベクトルで [-0.82456;0.56577] が固有値 0.36277,[-0.41597,-0.90938] が固有値 0.93723 に対応する。固有値は通常絶対値の大きい順番に並べられる。固有値の中で最大のものを、最大固有値と呼ぶ。ここでは、後者がそれに当たる。

Octave で実行してみよう。

```
octave:> source("tr1.m")
octave:> plot(X(:,1),X(:,2),'+')
octave:> e1=[0,0;1,0];
octave:> e2=[0,0;0,1];
octave:> ev1=[0,0;v1(1,2),v1(2,2)];
octave:> ev2=[0,0;v1(1,1),v1(2,1)];
octave:> hold on
octave:> plot(e1(:,1),e1(:,2))
octave:> plot(e2(:,1),e2(:,2))
octave:> plot(ev1(:,1),ev1(:,2))
octave:> plot(ev2(:,1),ev2(:,2))
```

直交座標系を表示するために、e1, e2 という単位ベクトルも一緒に表示した。横軸と縦軸ではスケールが違う場合があるので注意すること。最大固有値に対応する固有ベクトルの方向に沿って原点に引き込まれていることがわかる。固有値が共に 1 より小さいためである。2 番目の固有値は、0.3 程度と小さいため、この方向には急激に収縮する。1 以上であれば、今度は発散する。この性質については、後述する。

4.3.3 固有値の求め方

では、固有値はどうやって求めるのだろうか？

固有値の定義から考えよう。これは、

$$\lambda \mathbf{x} = M\mathbf{x} \tag{4.14}$$

であるから、

$$(M - \lambda E)\mathbf{x} = 0 \tag{4.15}$$

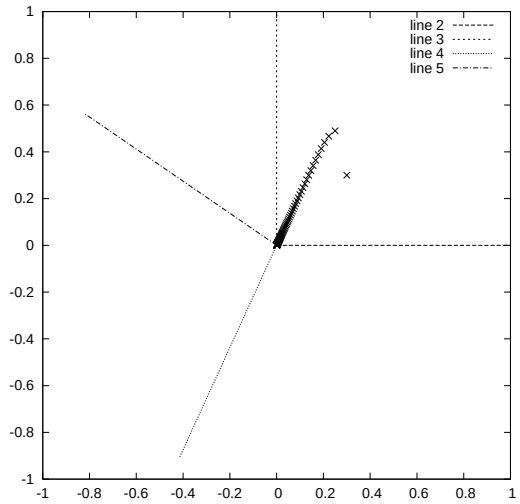

Figure 4.3: 固有値と固有ベクトルと、１次変換の実行例

と変形できる。ただし E は単位行列である。固有ベクトルが存在する、つまり $\mathbf{x} \neq 0$ のときに、この方程式が解をもつための条件は、

$$|M - \lambda E| = 0 \tag{4.16}$$

である。この方程式を λ について解くことが固有値を求めることである。この方程式を展開したものを特性方程式と呼び、解析的に解く場合は非常に重要である。ただし、行列が大きくなると結局計算機に頼らざるを得ない。3x3 以上の行列式は、ガウスの消去法などで 3x3 以下に分解して計算する。

Octave は、行列計算などは高速化したライブラリを使用するので、C 言語で自分でプログラムを組むよりは速く結果を得られる。試しに、巨大な行列の固有値を求めてみよう。

```
octave:6> BigM=randn(300,300);
octave:7> [v,l]=eig(BigM)
```

これくらいならすぐに答えが出てくるはずである。1000x1000 なども充分可能である。Octave 等のプログラムでは、ガウスの消去法で計算するよりも効率の高いアルゴリズムを使用している。興味のある向きは、QR 分解、Schur 分解などを調べてみるといい。

4.4 固有値と力学系の振る舞い

1 次変換の振る舞いは固有値と固有ベクトルをもとに分類できる。ここでは、様々な例により実際に調べてみることにする。

```
octave:49> M2=[1.01, 0.2; 0.01, 0.8]
MB =

  1.010000  0.200000
  0.010000  0.800000

octave:50> [v,l]=eig(M2)
v =

   0.998960  -0.674136
   0.045588   0.738607

l =

  1.01913  0.00000
  0.00000  0.79087
```

行列 M2 の固有値は「　　」と「　　」であり、最大固有値が 1 より大きい。上述の tr1.m をもとにした draweigvec.m を入力し、実行する。

```
#draweigvec.m
function X=draweigvec(M)
  [v1,l1]=eig(M);
  n=100;
  X=zeros(n,2);
  X0=[0.5;0.3];
  X(1,:)=X0(1,:)';
  XX=X0;
  for(i=2:n)
    XX = M*XX;
    X(i,:)=XX';
  endfor
  hold off;
  plot(X(:,1),X(:,2),'+');
  e1=[0,0;1,0];
```

```
  e2=[0,0;0,1];
  ev1=[0,0;v1(1,2),v1(2,2)];
  ev2=[0,0;v1(1,1),v1(2,1)];
  hold on
  plot(e1(:,1),e1(:,2))
  plot(e2(:,1),e2(:,2))
  plot(ev1(:,1),ev1(:,2))
  plot(ev2(:,1),ev2(:,2))
endfunction
```

実行例

```
octave:62> source("draweigvec.m")
octave:63> draweigvec(M2);
```

最大固有値の方向に引き延ばされることがみてとれた。1 回変換するごとに約 1.02 倍されるので、ゆっくりだが指数関数的に原点から離れてゆく。

draweigvec は、軌跡を出力するので、下記のようにコマンドを与えれば、その様子を調べることができる。n の値を大きくするともっとわかりやすくなる。

```
octave:> X1=draweigvec(M2);
octave:> hold off
octave:> plot(X1(:,1));
```

固有値が両方とも 1 より大きい場合は、発散的な振る舞いである。各自確かめられたい。

```
octave:> M3=[2.5, 1; 1, 2];
octave:> eig(MBB)
ans =

  1.2192
  3.2808
```

4.4.1 固有値が複素数の場合

固有値は、特性方程式の解として与えられることは既に触れた。つまり、固有値は実数とは限らないということである。

```
octave:24> M4=[0.5, -0.3; 0.25, 0.5]
M4 =
```

```
   0.50000  -0.30000
   0.25000   0.50000

octave:111> [v,l]=eig(M4)
v =

  0.73855 + 0.00000i  0.73855 - 0.00000i
  0.00000 - 0.67420i  0.00000 + 0.67420i

l =

  0.50000 + 0.27386i  0.00000 + 0.00000i
  0.00000 + 0.00000i  0.50000 - 0.27386i

octave:112> norm(l(1,1))
ans = 0.57009
octave:113> norm(l(2,2))
ans = 0.57009
```

行列 M4 の固有値は複素数であることがわかった。また、固有値の大きさはともに 1 より小さい。この場合、行列は回転成分を含む。(鋭い読者は、この章の前半で「回転行列の固有ベクトル」はどう表現されるか疑問に思ったはずである)。

```
octave:110> draweigvec(M4);
```

を実行すると、点列は回転しながら原点に引き込まれていくのがみてとれる。ただし、複素数の固有ベクトルは表示されず、コンソールにエラー出力が現れる。これは無視してよい。

4.4.2 まとめ

以上により求められた、1 次変換による力学系とその固有値による振る舞いをまとめると以下のようになる。

- 固有値が実数で、最大固有値の絶対値 $|\lambda_1| < 1$ のとき
  最大固有値に対応する固有ベクトルに沿って指数関数的に原点に吸引される。
- 固有値が実数で、最大固有値の絶対値 $|\lambda_1| > 1$ のとき
  最大固有値に対応する固有ベクトルに沿って指数関数的に発散する。
- 固有値が複素数で、最大固有値の絶対値 $|\lambda_1| < 1$ のとき
  螺旋状に原点に吸引される。

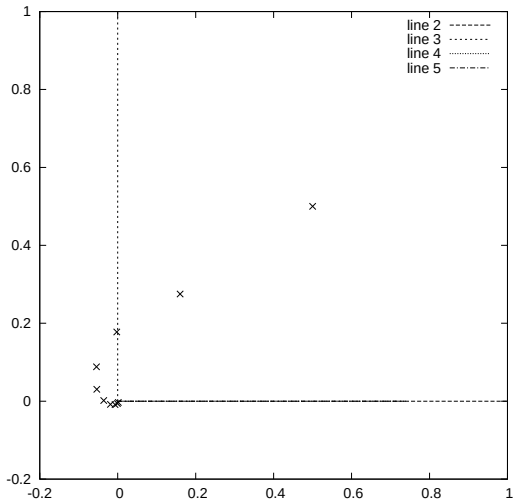

Figure 4.4: 行列 M4 による一次変換により定義された力学系の軌跡

- 固有値が複素数で、最大固有値の絶対値 $|\lambda_1| > 1$ のとき
  螺旋状に発散する。

4.5 レポート問題 8

上記の分類に対応する 1 次変換行列を適当な数を代入することによってつくれ。それぞれについて行列、eig() により求めた固有値と固有ベクトル、draweigvec() により表示されたグラフを示せ。