

認識行動システム論 第10回

梶本裕之

Twitter ID kajimoto

ハッシュタグ #ninshiki

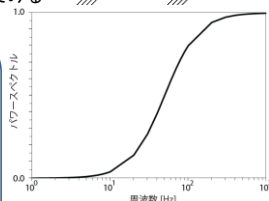
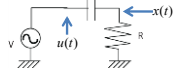
日程

10/13 イントロダクション
10/20 Scilabの紹介(3階PCルーム)
10/27 フーリエ変換
11/03 文化の日
11/10 出張
11/17 調布祭準備
11/24 出張
12/01 フーリエ変換と線形システム
12/08 創立記念日(配属説明会)
12/15 信号処理の基礎
12/22 信号処理応用1(相関)
～中間レポート(冬休み中)～
01/05 信号処理応用2(画像処理)
01/12 ラプラス変換
01/19 古典制御の基礎
01/26 行列
02/02 行列と最小二乗法
02/09 ロボティクス
～期末テスト～

レポート課題: ハイパスフィルタの伝達関数

$$H(\omega) = \frac{sCR}{sCR+1} = \frac{s}{s + \frac{1}{CR}}$$

$s=j\omega$ を代入して、周波数応答をしてみる



```
Scilabソース
R=32; //抵抗 32Ω
C=10^(-4); //コンデンサ100μF
f=[0:1000]; //周波数
w=2 * %pi * f; //角周波数
//応答
H=%i*w ./ (%i*w + 1/(R*C));

//パワースペクトル
power_spec = H .* conj(H);

//片対数グラフで表示
plot2d1("oln",f,power_spec);
```

32Ωの抵抗と100μFのコンデンサで、
10Hz程度以下の周波数を阻止する
ハイパスフィルタが出来た。
(コンデンサの容量を変更しました。)

行列

行列... 1, 2年でやったはず

今日の内容

- 固有値とは、固有ベクトルとは
- 行列の対角化: どうやるか、なぜうれしいのか
- 制御における行列: さわりだけ

キーワードは、

固有値, 固有ベクトル, 対角化

行列: データ列を変換するもの

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

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

(例1)

y: 観測データ, A: システムの性質, x: 媒介変数

(例2)

y: フーリエ空間での周波数成分, A: フーリエ変換行列,
x: 実空間でのデータ系列

(例) 2軸力センサ

カセンサA レバー カセンサB

$F_Z = x_A + x_B$
 $F_X = k(x_A - x_B)$

2x1ベクトル $\begin{bmatrix} F_Z \\ F_X \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ k & -k \end{bmatrix} \begin{bmatrix} x_A \\ x_B \end{bmatrix}$ 2x1ベクトル

2x2行列

(例) 多軸力センサ

Lever

Sensing Elements

$F_Z = k_1(x_A + x_B + x_C + x_D)$
 $F_X = k_2((x_A + x_B) - (x_C + x_D))$
 $F_Y = k_3((x_A + x_C) - (x_B + x_D))$

3 $\begin{bmatrix} F_Z \\ F_X \\ F_Y \end{bmatrix} = \begin{bmatrix} \dots \\ \dots \\ \dots \end{bmatrix} \begin{bmatrix} x_A \\ x_B \\ x_C \\ x_D \end{bmatrix}$ 4

3x4行列

一般には正方形行列ではない！！
 (例)6軸力センサには8つ以上のセンサエレメント内蔵

力センサのキャリブレーション(較正)

カセンサA レバー カセンサB

$F_Z = k_1x_A + k_2x_B$
 $F_X = k_3x_A + k_4x_B$

$\begin{bmatrix} F_Z \\ F_X \end{bmatrix} = \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix} \begin{bmatrix} x_A \\ x_B \end{bmatrix}$

k1~k4のパラメータは元々未知。
 これを求めなければ使えない！！

逆行列

カセンサA レバー カセンサB

$\begin{bmatrix} F_Z \\ F_X \end{bmatrix} = \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix} \begin{bmatrix} x_A \\ x_B \end{bmatrix}$

これを $\mathbf{f} = \mathbf{A}\mathbf{x}$ と書く。

ここで、
 「ある力を加えたときの各センサ出力は？」
 という逆の関係を考える。

$\mathbf{x} = \mathbf{A}^{-1}\mathbf{f} = \mathbf{G}\mathbf{f}$

$\begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} F_Z \\ F_X \end{bmatrix}$

逆行列の「測定」

$\mathbf{x} = \mathbf{G}\mathbf{f}$

$\begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} F_Z \\ F_X \end{bmatrix}$

(1) $F_Z=1, F_X=0$ の力を加え、各センサの出力を記録

$\begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \end{bmatrix}$

各センサの出力に、逆行列の成分g1, g3が現れる！

(2) $F_Z=0, F_X=1$ の力を加え、各センサの出力を記録

$\begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \end{bmatrix}$

各センサの出力に、逆行列の成分g2, g4が現れる！

逆行列の「測定」

$\mathbf{x} = \mathbf{G}\mathbf{f} = \mathbf{A}^{-1}\mathbf{f}$

$\begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} F_Z \\ F_X \end{bmatrix}$

$\mathbf{G} = \mathbf{A}^{-1}$ の成分, g1~g4が得られたので、その逆行列を計算すればAが得られる。

$\mathbf{f} = \mathbf{A}\mathbf{x}$

まとめると、

$\begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix}$

行列に単位行列をかけたことに相当

単位力でなくて良い

$$\begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} f_{z1} \\ f_{x1} \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$$

2回目の入力 2回目の出力



$$GF = M$$

$$G = MF^{-1}$$

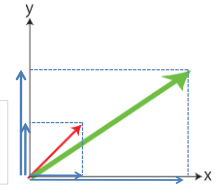
- 2回**既知**のカベクトルを加えて、各センサの出力を得る
- カベクトルを並べたものをカ行列F、センサ出力を並べたものをカ行列Mとする
- カ行列の逆行列F⁻¹をMにかければ、カ行列Gが得られる。
- Gの逆行列が望んだ「校正カ行列」A

ほとんどすべてのカ行列は、ベクトルを「引き延ばす」ものである(1)

$$v = Au \quad \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

$$A = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \text{ の時,}$$

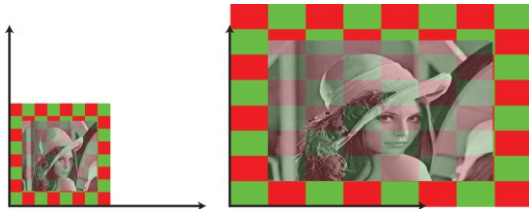
$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} =$$



(カ作用)xカ成分を3倍、yカ成分を2倍に引き延ばす

ほとんどすべてのカ行列は、ベクトルを「引き延ばす」ものである(1)

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

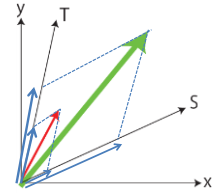


(カ作用)xカ成分を3倍、yカ成分を2倍に引き延ばす

ほとんどすべてのカ行列は、ベクトルを「引き延ばす」ものである(2)

$$A = \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix} \text{ の時は? ... ちょっと分からない.}$$

- カ作用が、
- 謎のsカ成分をs倍、
 - 謎のTカ成分をt倍
- に引き延ばすことだと仮定してみる。



ただしもはや、謎のS,Tカ軸は直交してなくて良い。

固有ベクトルと固有値

固有ベクトル、固有値とは、謎のS、Tカ軸、およびs,t倍のことである。

(求める手続き)

(1) λ倍されるだけでカ方向不変のベクトルがあると仮定

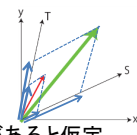
$$Au = \lambda u$$

(2) 式カ変カ

$$Au = \lambda u = \lambda Iu \quad \begin{bmatrix} 8-\lambda & -2 \\ 3 & 1-\lambda \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = 0$$

$$(A - \lambda I)u = 0$$

$$A = \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix}$$



固有ベクトルと固有値

$$\begin{bmatrix} 8-\lambda & -2 \\ 3 & 1-\lambda \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = 0$$

$$\begin{bmatrix} a-\lambda & b \\ c & d-\lambda \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = 0$$

$$(a-\lambda)u_x + bu_y = 0 \rightarrow u_y = -(a-\lambda)u_x / b$$

$$cu_x + (d-\lambda)u_y = 0$$

$$cu_x - (d-\lambda)(a-\lambda)u_x / b = 0$$

$$((a-\lambda)(d-\lambda) - bc)u_x = 0$$

$$\therefore (a-\lambda)(d-\lambda) - bc = 0$$

この解λ₁, λ₂を固有値と呼び、カ対応するベクトルを固有ベクトルと呼ぶ。

固有ベクトルと固有値

$$\begin{bmatrix} 8-\lambda & -2 \\ 3 & 1-\lambda \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = 0$$

$\lambda_1 = 2$ に対応する固有ベクトルは

$$\begin{bmatrix} 8-2 & -2 \\ 3 & 1-2 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = 0$$

大きさを1とすれば, $k = 1/\sqrt{10}$

$\lambda_2 = 7$ に対応する固有ベクトルは

$$\begin{bmatrix} 8-7 & -2 \\ 3 & 1-7 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = 0$$

大きさを1とすれば, $k = 1/\sqrt{5}$

作用: e1軸上のベクトルは2倍,
e2軸上のベクトルは7倍する.

固有ベクトルと固有値

$$A = \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix}$$

作用: e1軸上のベクトルは2倍,
e2軸上のベクトルは7倍する.

- やはり引き延ばす作用である
- ただし引き延ばす軸(固有ベクトル)は非直交

行列と座標変換

- 引き延ばす作用である
- ただし引き延ばす軸(固有ベクトル)は非直交(斜交)

いまいわかりにくい...

行列の作用を,

- (1) 引き延ばし軸での成分表示に変換し,
- (2) 各成分を引き延ばし,
- (3) 合成して元に戻す

ように分解すればわかりやすい...はず??

まとめると

行列Tの作用は次の3段階に分解できる。

- (1) 引き延ばし軸での成分表示に変換し,
- (2) 各成分を引き延ばし,
- (3) 合成して元に戻す

引き延ばし軸での成分表示(誤)

行列の作用を,

- (1) 引き延ばし軸での成分表示に変換し,
- (2) 各成分を引き延ばし,
- (3) 合成して元に戻す

内積で成分を取れる?...誤り

斜交座標では
「成分を取り出す操作」≠「内積」

先に(3)「合成」を考えよう。
(1)はその逆のはず

合成して元に戻す操作

行列の作用を,

- (1) 引き延ばし軸での成分表示に変換し,
- (2) 各成分を引き延ばし,
- (3) 合成して元に戻す

この操作は単なるベクトルの足し算にすぎない
(e1成分の大きさ) $e_1 +$ (e2成分の大きさ) e_2

$$= [e_1 \ e_2] \begin{bmatrix} e_1 \text{軸成分} \\ e_2 \text{軸成分} \end{bmatrix}$$

$P = [e_1 \ e_2]$ とおいて

$$= P \begin{bmatrix} e_1 \text{軸成分} \\ e_2 \text{軸成分} \end{bmatrix}$$

引き延ばし軸での成分表示(改)

行列の作用を、
 (1) 引き延ばし軸での成分表示に変換し、
 (2) 各成分を引き延ばし、
 (3) 合成して元に戻す

(3)「合成」が、

$$P \begin{bmatrix} e_1 \text{軸成分} \\ e_2 \text{軸成分} \end{bmatrix}$$

出来るのだから、(1)はその逆のはず。
すなわち

$$P^{-1} \begin{bmatrix} x \text{軸成分} \\ y \text{軸成分} \end{bmatrix}$$

により引き延ばし軸での成分表示ができる

引き延ばし軸での引き延ばし

行列の作用を、
 (1) 引き延ばし軸での成分表示に変換し、
 (2) 各成分を引き延ばし、
 (3) 合成して元に戻す

各成分を
 固有ベクトル e_1 軸に沿って λ_1 倍、
 固有ベクトル e_2 軸に沿って λ_2 倍する。

この操作は、

$$\begin{bmatrix} e_1 \text{軸成分を} \lambda_1 \text{倍} \\ e_2 \text{軸成分を} \lambda_2 \text{倍} \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} e_1 \text{軸成分} \\ e_2 \text{軸成分} \end{bmatrix}$$

合成して元に戻す

行列の作用を、
 (1) 引き延ばし軸での成分表示に変換し、
 (2) 各成分を引き延ばし、
 (3) 合成して元に戻す

この操作は、
 (これまでに求めた e_1 成分の大きさ) e_1
 +
 (これまでに求めた e_2 成分の大きさ) e_2

$$= P \begin{bmatrix} \text{これまでに求めた} e_1 \text{軸成分} \\ \text{これまでに求めた} e_2 \text{軸成分} \end{bmatrix} \quad P = [e_1 \ e_2]$$

まとめると

行列Tの作用は次の3段階に分解できる。
 (1) 引き延ばし軸での成分表示に変換し、
 (2) 各成分を引き延ばし、
 (3) 合成して元に戻す

$$Ax = P \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} P^{-1}x$$

固有値を対角成分に並べた行列をTと置く。 $T = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$

$$Ax =$$

重要な応用: A^n

$$\begin{aligned} A^n x &= (PTP^{-1})^n x \\ &= P \underbrace{T P^{-1} P T P^{-1} \dots P T P^{-1}}_n x \\ &= P T^n P^{-1} x \\ &= P \begin{bmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{bmatrix} P^{-1} x \end{aligned}$$

行列のn乗を簡単に計算することができる

重要な結論: nが非常に大きくなった時の A^n

$$A^n x = P \begin{bmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{bmatrix} P^{-1} x$$

行列の固有値の絶対値が引き延ばしの倍率だから、
 固有値が

- 一つでも1より大きければ、 A^n は発散する
- 全て1より小さければ、 A^n は0に収束する

例: A^n

$$A = \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 58 & -18 \\ 27 & -5 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 58 & -18 \\ 27 & -5 \end{bmatrix} \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 410 & -134 \\ 201 & -59 \end{bmatrix}$$

$$P = \begin{bmatrix} 1/\sqrt{10} & 2/\sqrt{5} \\ 3/\sqrt{10} & 1/\sqrt{5} \end{bmatrix} \quad \lambda_1 = 2 \quad \lambda_2 = 7 \quad \text{を代入して,}$$

$$A^3 = P T^3 P^{-1} = \begin{bmatrix} 1/\sqrt{10} & 2/\sqrt{5} \\ 3/\sqrt{10} & 1/\sqrt{5} \end{bmatrix} \begin{bmatrix} 2^3 & 0 \\ 0 & 7^3 \end{bmatrix} \begin{bmatrix} 1/\sqrt{10} & 2/\sqrt{5} \\ 3/\sqrt{10} & 1/\sqrt{5} \end{bmatrix}^{-1}$$

$$= \dots = \begin{bmatrix} 410 & -134 \\ 201 & -59 \end{bmatrix} \quad \text{固有値が大きいのでどんどん大きくなる}$$

ほとんどすべての行列は、
ベクトルを「引き延ばす」ものである(3)

$$A = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad \text{の時は? ... 回転と習ったはず}$$

以前と同様に固有値と固有ベクトルを求めてみる。

$$\begin{bmatrix} \cos \theta - \lambda & -\sin \theta \\ \sin \theta & \cos \theta - \lambda \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = 0$$

$$(a - \lambda)(d - \lambda) - bc = 0$$

$$(\cos \theta - \lambda)(\cos \theta - \lambda) + \sin^2 \theta = 0$$

回転行列の固有値 = $\exp(j\theta)$

$$(\cos \theta - \lambda)(\cos \theta - \lambda) + \sin^2 \theta = 0$$

$$\cos^2 \theta - 2\lambda \cos \theta + \lambda^2 + \sin^2 \theta = 0$$

$$\lambda^2 - 2\lambda \cos \theta + 1 = 0$$

$$\lambda = \frac{2 \cos \theta \pm \sqrt{4 \cos^2 \theta - 4}}{2}$$

$$= \frac{2 \cos \theta \pm j \sqrt{4 \sin^2 \theta}}{2}$$

$$= \cos \theta \pm j \sin \theta$$

$$= \exp(\pm j\theta)$$

回転行列の固有ベクトル

$\cos \theta + j \sin \theta$ に対応する固有ベクトルは

$$\begin{bmatrix} \cos \theta - \lambda & -\sin \theta \\ \sin \theta & \cos \theta - \lambda \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} \cos \theta - \cos \theta - j \sin \theta & -\sin \theta \\ \sin \theta & \cos \theta - \cos \theta - j \sin \theta \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

$$= \sin \theta \begin{bmatrix} -j & -1 \\ 1 & -j \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = 0$$

$$\therefore u_x = j u_y \quad e_1 = k \begin{bmatrix} j \\ 1 \end{bmatrix} \quad \text{大きさを1とすれば例えば, } k = 1/\sqrt{2}$$

$\cos \theta - j \sin \theta$ に対応する固有ベクトルは

$$e_2 = k \begin{bmatrix} 1 \\ j \end{bmatrix} \quad \text{大きさを1とすれば例えば, } k = 1/\sqrt{2}$$

(参考)

回転行列も(拡張された)引き延ばしである

- 一般の行列は、固有値、固有ベクトル共に複素数。
- x,y軸に加えて、複素軸も含めた4次元空間中でこれまでと同様の引き延ばしを行う演算とみなせる。
- 複素固有値の絶対値が引き延ばし倍率、偏角が回転角度を表す。

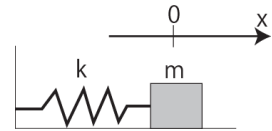
制御における行列

おもりの挙動をシミュレートしたい

```

m=1.0; //重さ
k=1.0; //ばね定数
x=1.0; //初期位置
v=0; //初期速度
dt=0.1; //時間刻み
record=[]; //記録用
for time= 0:dt:10 //時刻
    F=-k*x; //ばねによって生じる力
    a=F/m; //生じる加速度
    v= v+a*dt; //速度
    x= x+v*dt; //位置
    record = [record,x]; //記録(※テクニック:ベクトルが伸びていく)
end
plot([0:dt:10],record);

```

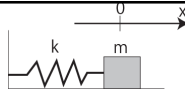


制御における行列

```

for time= 0:dt:10 //時刻
    F=-k*x; //ばねによって生じる力
    a=F/m; //生じる加速度
    v=v+a*dt; //速度
    x=x+v*dt; //位置
end

```



位置, 速度, 加速度を並べた「状態ベクトル」 x を定義 $x = \begin{bmatrix} x \\ v \\ a \end{bmatrix}$

上の関係から, dt時間後の新たな位置, 速度, 加速度は

$$\begin{bmatrix} x_n \\ v_n \\ a_n \end{bmatrix} = \begin{bmatrix} \text{---} \\ \text{---} \\ \text{---} \end{bmatrix} \begin{bmatrix} x_{n-1} \\ v_{n-1} \\ a_{n-1} \end{bmatrix} = Ax$$

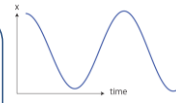
制御における行列

```

Scilabコード
m=1.0; //重さ
k=1.0; //ばね定数
x=1.0; //初期位置
v=0; //初期速度
a=-k/m*x; //初期加速度
dt=0.01; //時間刻み
record=[]; //記録用

state=[x;v;a];

```



$$\begin{bmatrix} x_n \\ v_n \\ a_n \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 \\ 0 & 1 & dt \\ -k/m & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{n-1} \\ v_{n-1} \\ a_{n-1} \end{bmatrix} = Ax_{n-1} = \dots = A^n x_0$$

```

A=[1,dt,0;0,1,dt;-k/m,0,0];

```

```

for time= 0:dt:10 //時刻

```

```

    state= A*state;
    record = [record,state(1)];
end

```

```

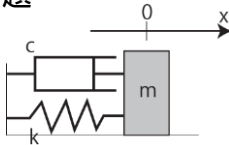
plot([0:dt:10],record);

```

•行列Aのn乗を使えば,
n時刻先の状態をシミュレート可能

•行列Aの固有値を見れば,
システムが将来($n \rightarrow \infty$)収束するか
発散するか予測可能!

レポート課題



●ダンパを加えた際の行列を考え,
同様のシミュレーションプログラムを書け

●行列Aの固有値の絶対値が1よりも小さいこと,
すなわち位置が収束することを確認し, コメントに記せ
(Scilabでは固有値はspec()で求められる)

注意:ここで導入した行列はあくまで導入編用で,
シミュレーションとしては不正確です。