

# インタラクティブシステム論 第9回

梶本裕之



## 日程

講義番号	講義日	講義内容	pdf	video
1	4/14	イントロダクション	[ <a href="#">pdf</a> ] 2025年版	<a href="#">video</a>
		Scilab課題	[ <a href="#">pdf</a> ](更新なし)	
		上記資料のPython版	[ <a href="#">pdf</a> ](更新なし)	
2	4/21	フーリエ変換	[ <a href="#">pdf</a> ] 2025年版	<a href="#">video</a>
3	4/28(出張のためオンライン・オンデマンド)	フーリエ変換と線形システム	[ <a href="#">pdf</a> ] 2025年版	<a href="#">video</a>
-	5/5	祝日		
4	5/12	信号処理の基礎	[ <a href="#">pdf</a> ] 2025年版	<a href="#">video</a>
5	5/19	信号処理の応用1(相関)	[ <a href="#">pdf</a> ] 2025年版	<a href="#">video</a>
6	5/26	信号処理の応用2(画像処理)	[ <a href="#">pdf</a> ] 2025年版	<a href="#">video</a>
-	6/2	中間テスト準備 (自習)	[ <a href="#">pdf</a> ]2022年版	
-	6/9	中間確認テストとその解説		
7	6/16(出張のためオンライン・オンデマンド)	ラプラス変換	[ <a href="#">pdf</a> ] 2024年版	<a href="#">video</a>
8	6/23	古典制御の基礎	[ <a href="#">pdf</a> ] 2024年版	<a href="#">video</a>
9	6/30	行列	[ <a href="#">pdf</a> ] 2024年版	<a href="#">video</a>
10	7/7(出張のためオンライン・オンデマンド)	行列と最小二乗法	[ <a href="#">pdf</a> ] 2024年版	<a href="#">video</a>
11	7/14	ロボティクス	[ <a href="#">pdf</a> ] 2024年版	<a href="#">video</a>
-	7/21	海の日：期末テスト準備 (自習)	[ <a href="#">pdf</a> ]2022年版	
-	7/28	期末確認テストとその解説		

変更がある場合は随時アナウンスします。Google Classroomに注意

# 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: **なにをしたことになるか, なぜうれしいのか**
- 情報理論・制御における行列の応用事例

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)

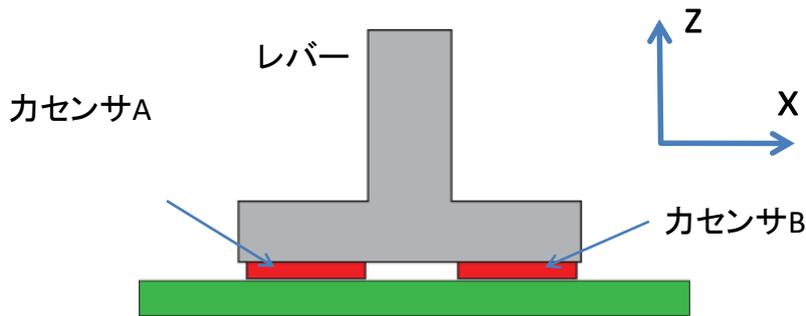
y:フーリエ空間での周波数成分, A:フーリエ変換行列,  
x:実空間でのデータ系列



## (例) 2軸力センサ



# (例) 2軸力センサ



$$F_Z = x_A + x_B$$

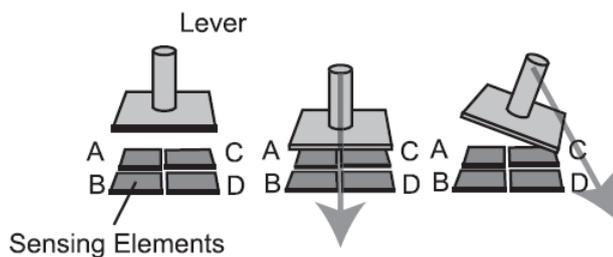
$$F_X = k(x_A - x_B)$$

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

2x2行列



# (例) 多軸力センサ



$$F_Z = k_1(x_A + x_B + x_C + x_D)$$

$$F_X = k_2((x_A + x_B) - (x_C + x_D))$$

$$F_Y = k_3((x_A + x_C) - (x_B + x_D))$$

3  $\left\{ \begin{bmatrix} F_z \\ F_x \\ F_y \end{bmatrix} = \begin{bmatrix} \phantom{0} & \phantom{0} & \phantom{0} & \phantom{0} \\ \phantom{0} & \phantom{0} & \phantom{0} & \phantom{0} \\ \phantom{0} & \phantom{0} & \phantom{0} & \phantom{0} \end{bmatrix} \begin{bmatrix} x_A \\ x_B \\ x_C \\ x_D \end{bmatrix} \right\}$  4

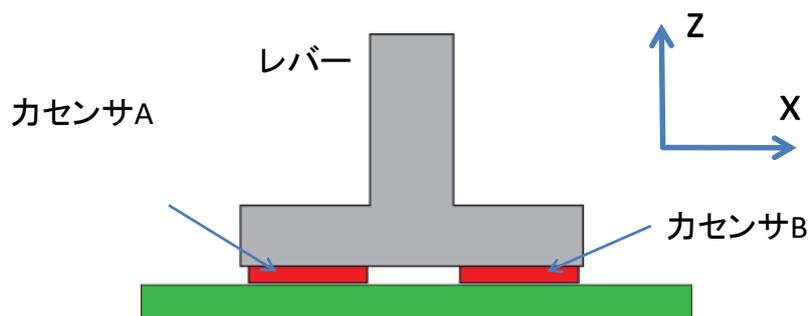
3x4行列

一般には正方行列ではない！！

(例)6軸力センサには8つ以上のセンサエレメント内蔵



# カセンサのキャリブレーション（校正）



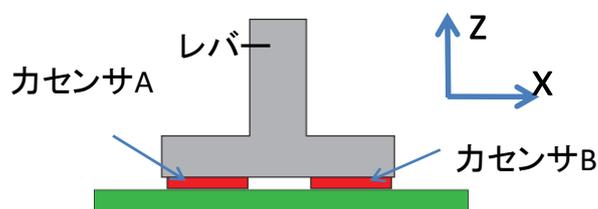
$$\begin{aligned} F_Z &= k_1 x_A + k_2 x_B \\ F_X &= k_3 x_A + k_4 x_B \end{aligned} \quad \begin{bmatrix} F_Z \\ F_X \end{bmatrix} = \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix} \begin{bmatrix} x_A \\ x_B \end{bmatrix}$$

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

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



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

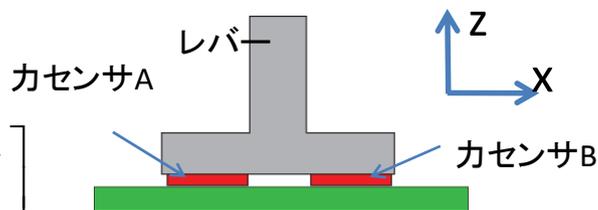
ここで、  
「ある力を加えたときの各センサ出力は？」  
という逆の関係を考える。

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{f} = \mathbf{G}\mathbf{f} \quad \begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} F_Z \\ F_X \end{bmatrix}$$



# 逆行列の「測定」

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



(1)  $F_z=1, F_x=0$ の力を加え、各センサの出力を記録

$$\begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} \phantom{0} \\ \phantom{0} \end{bmatrix} = \begin{bmatrix} \phantom{0} \\ \phantom{0} \end{bmatrix}$$



各センサの出力に、逆行列の成分 $g_1, g_3$ 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

$$\begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} \phantom{0} \\ \phantom{0} \end{bmatrix} = \begin{bmatrix} \phantom{0} \\ \phantom{0} \end{bmatrix}$$



各センサの出力に、逆行列の成分 $g_2, g_4$ が現れる！

# 逆行列の「測定」

$$\mathbf{x} = \mathbf{Gf} = \mathbf{A}^{-1}\mathbf{f} \quad \begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} F_Z \\ F_X \end{bmatrix}$$



$\mathbf{G}=\mathbf{A}^{-1}$ の成分,  $g_1 \sim g_4$ が得られたので、その逆行列を計算すれば $\mathbf{A}$ が得られる。

$$\mathbf{f} = \mathbf{Ax}$$

まとめると,

$$\left. \begin{array}{l} \begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_3 \end{bmatrix} \\ \begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} g_2 \\ g_4 \end{bmatrix} \end{array} \right\} \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix}$$

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

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

1回目の入力      1回目の出力  
2回目の入力      2回目の出力



$$GF = M$$

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



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
2. 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行列Mとする
3. カ行列の逆行列F<sup>-1</sup>をMにかければ、行列Gが得られる。
4. Gの逆行列が望んだ「較正行列」A

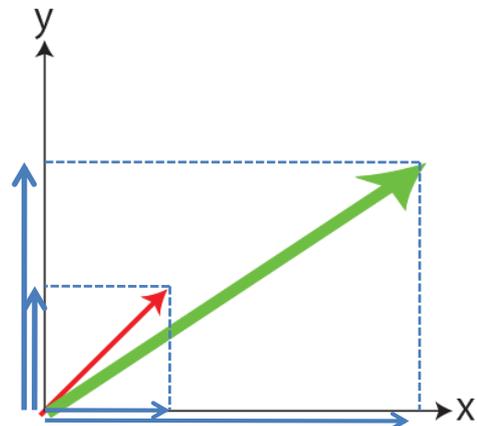


ほとんどすべての行列は、  
ベクトルを「引き延ばす」ものである (1)

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

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

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

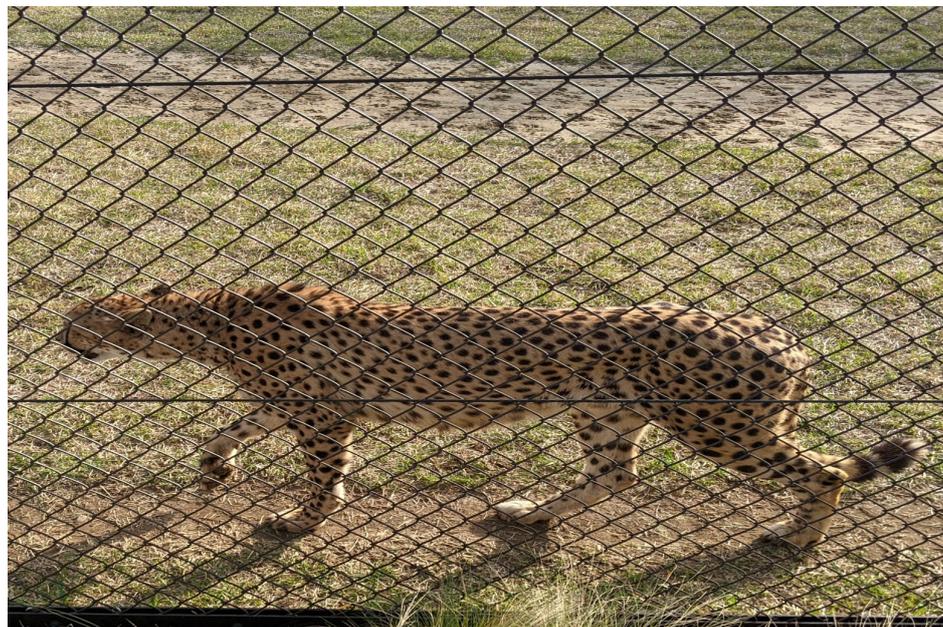


(作用)x軸成分を3倍、y軸成分を2倍に引き延ばす



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 (1)

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



(作用)x軸成分を3倍, y軸成分を2倍に引き延ばす



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 (2)

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \text{ は, x軸成分を3倍, y軸成分を2倍に引き延ばす}$$

では,

$$\mathbf{A} = \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix} \text{ の時は? . . . . よく分からない.}$$

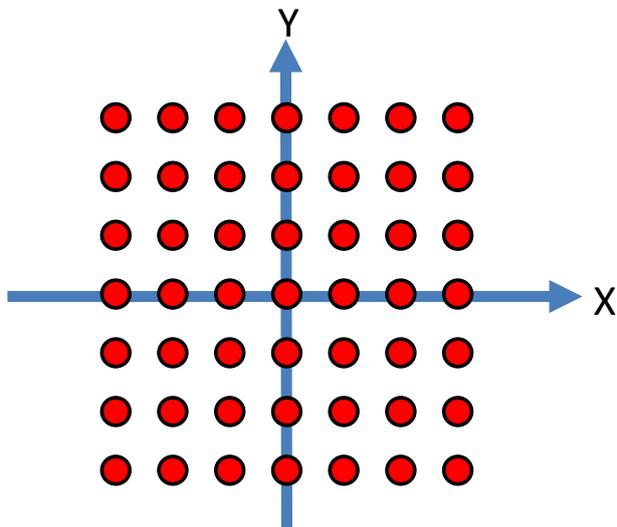


# 試してみる

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

で、平面上の点群はどう移動するか

X=-3~3, Y=-3~3の点群で検証



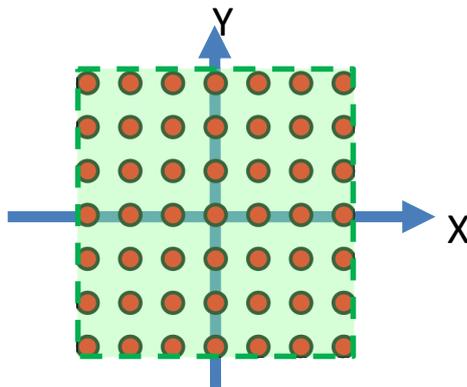
Scilabコード

```
A=[8,-2;3,1];  
s=[];  
t=[];  
for x=-3:3  
  for y=-3:3  
    r=A*[x;y];  
    s=[s,r(1)]; //x座標格納  
    t=[t,r(2)]; //y座標格納  
  end  
end  
plot(s,t,'o');
```

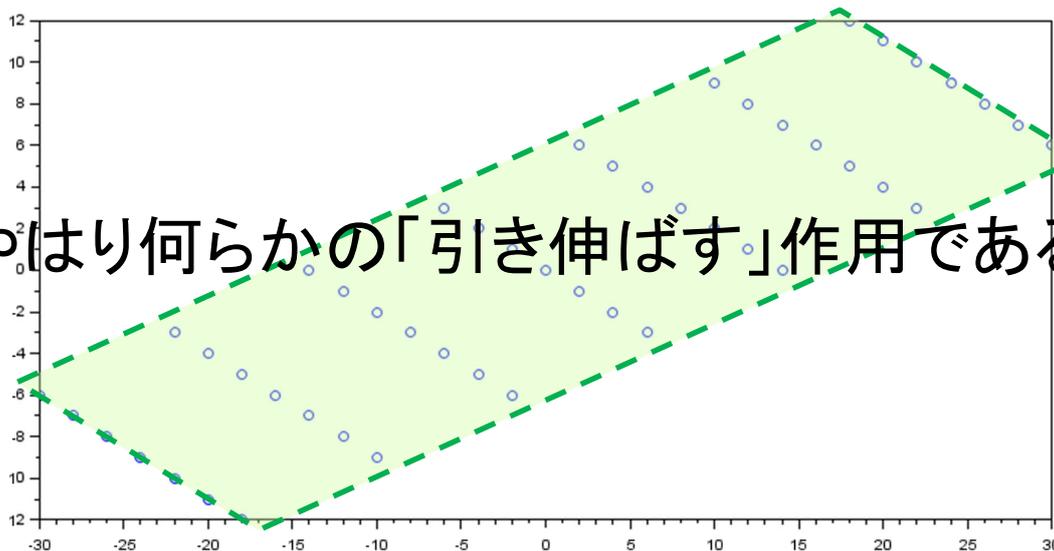


# 試してみる

変換前



変換後

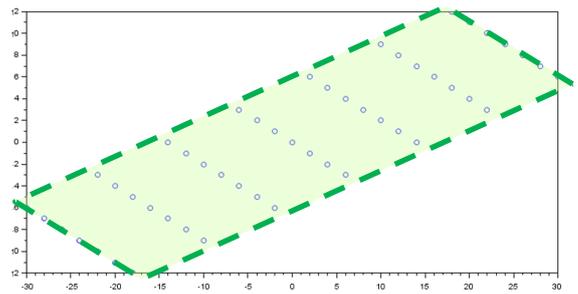


やはり何らかの「引き伸ばす」作用である



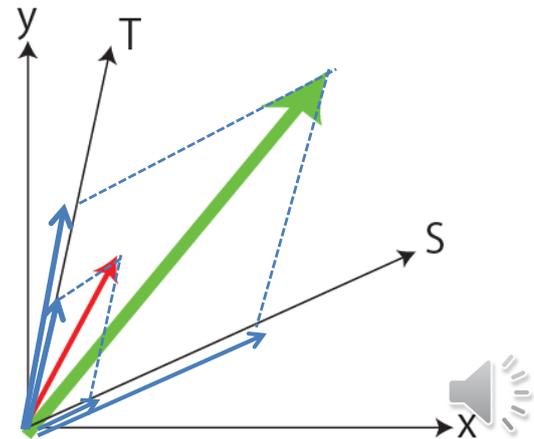
# ほとんどすべての行列は、 ベクトルを「引き延ばす」ものである (2)

$$\mathbf{A} = \begin{bmatrix} 8 & -2 \\ 3 & 1 \end{bmatrix} \quad \text{の作用は}$$



- 謎のS軸成分をs倍,
- 謎のT軸成分をt倍  
に引き延ばすことである

ただしもはや、  
このS,T軸は直交していない。



## 固有ベクトルと固有値

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

固有ベクトル, 固有値とは,  
謎のS, T軸, およびs,t倍  
のことである。

(求める手続き)

(1)  $\lambda$ 倍されるだけで方向不変のベクトルがあると仮定

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u}$$

(2) 式変形

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u} = \lambda\mathbf{I}\mathbf{u} \quad \begin{bmatrix} 8-\lambda & -2 \\ 3 & 1-\lambda \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \mathbf{0}$$
$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{u} = \mathbf{0}$$



# 固有ベクトルと固有値

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

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

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

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

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

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

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

この解 $\lambda_1, \lambda_2$ を固有値と呼び、  
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,  $k = 1/\sqrt{5}$

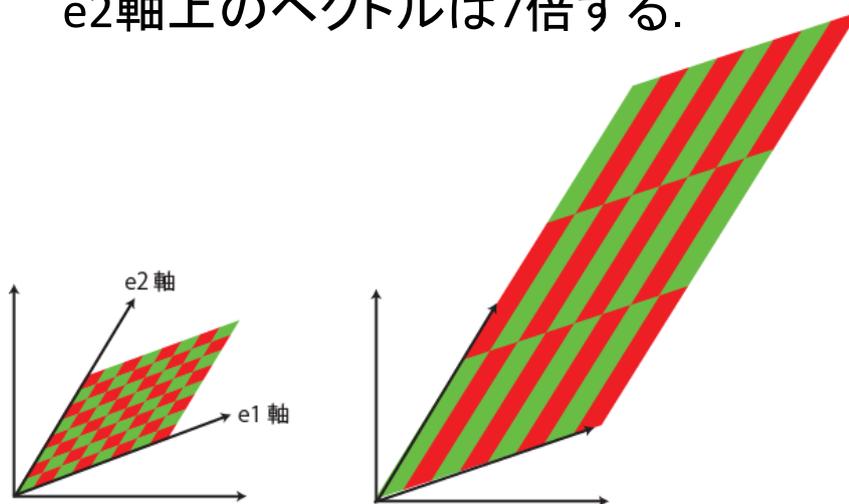
作用: e1軸上のベクトルは2倍,  
e2軸上のベクトルは7倍する。



# 固有ベクトルと固有値

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

作用:  $e_1$ 軸上のベクトルは2倍,  
 $e_2$ 軸上のベクトルは7倍する.



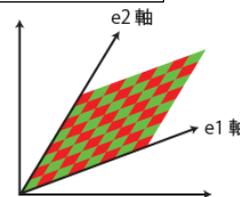
- やはり引き延ばす作用である
- 固有ベクトル上のベクトルは, 固有値倍される



# 行列と座標変換

- 引き延ばす作用である
- 固有ベクトル上のベクトルは, 固有値倍される

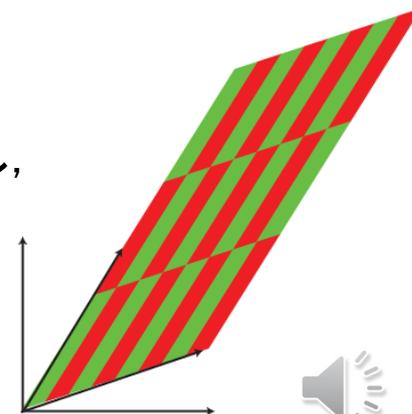
わかりにくい...



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

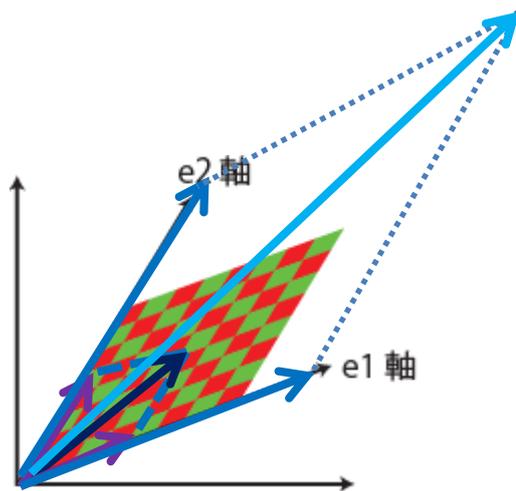
ように分解すればわかりやすいはず??



# 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



## (3) 合成して元に戻す操作, から考える

行列の作用を,

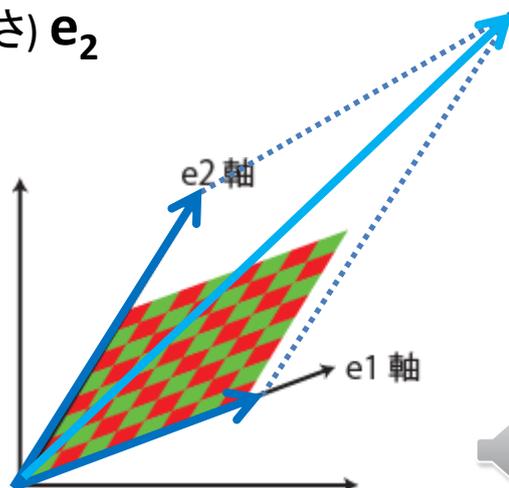
- (1) 引き延ばし軸での成分表示に変換し,
- (2) 各成分を引き延ばし,
- (3) 合成して元に戻す

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

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

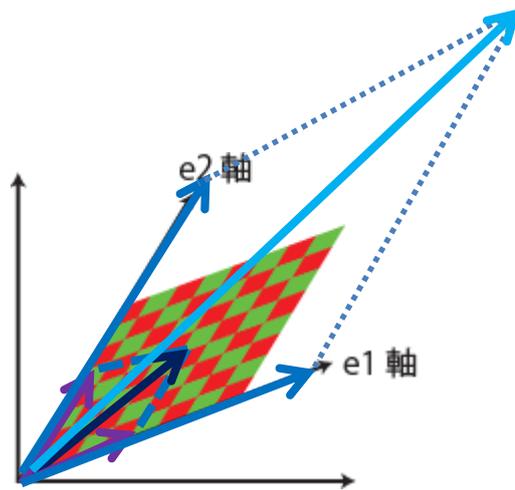
$$\mathbf{P} = \begin{bmatrix} e_1 & e_2 \end{bmatrix} \text{ において}$$

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



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



## (1) 引き延ばし軸での成分表示

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

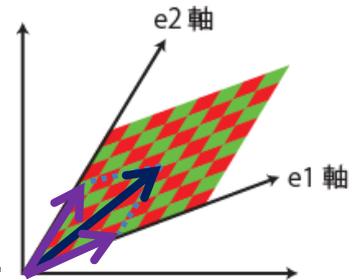
(3)「合成」が,

$$\mathbf{P} \begin{bmatrix} \mathbf{e}_1 \text{軸成分} \\ \mathbf{e}_2 \text{軸成分} \end{bmatrix}$$

で出来るのだから, (1)はその逆のはず.  
すなわち

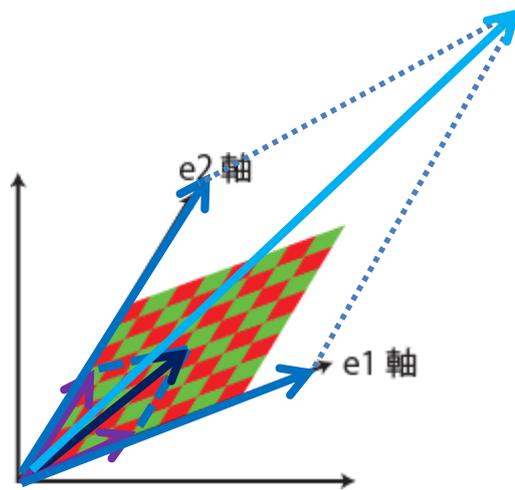
$$\mathbf{P}^{-1} \begin{bmatrix} x \text{軸成分} \\ y \text{軸成分} \end{bmatrix}$$

により引き延ばし軸での成分表示ができる



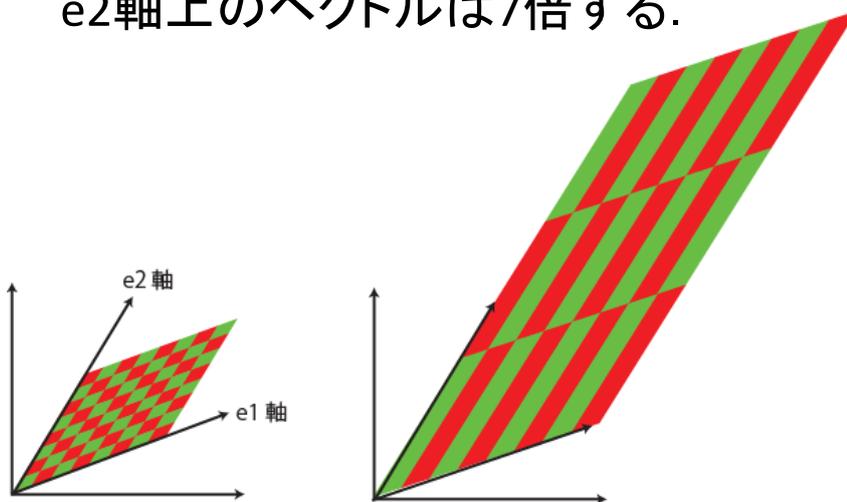
行列Tの作用は次の3段階に分解できる.

- (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



(再) 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
- 固有ベクトル上のベクトルが, 固有値倍される



## (2) 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,

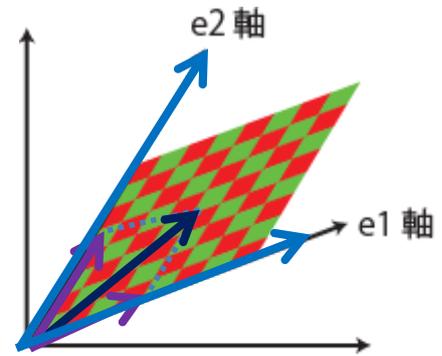
(2) 各成分を引き延ばし,

(3) 合成して元に戻す

各成分を

固有ベクトル $e_1$ 軸に沿って固有値 $\lambda_1$ 倍,  
固有ベクトル $e_2$ 軸に沿って固有値 $\lambda_2$ 倍する.

この操作は,



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



## まとめると

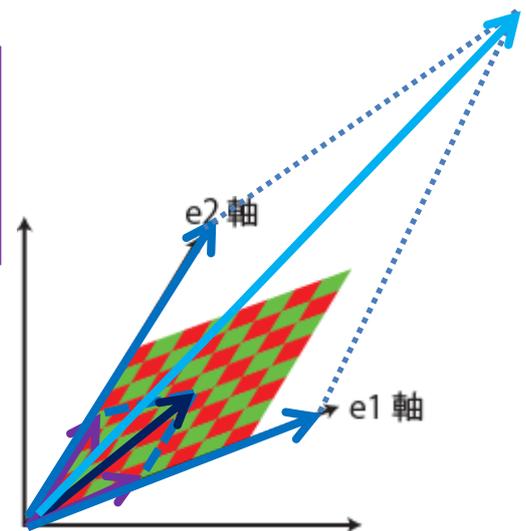
行列 $T$ の作用は次の3段階に分解できる.

(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

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



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

$$\mathbf{Ax} =$$



# 行列の対角化を数式で導出

行列の対角化を数式で導出する。  
まず、2つの固有値を $\lambda_1, \lambda_2$ 、固有ベクトルを $e_1, e_2$ とする。

2つの式を「まとめて」書くと次のようになる。

$[e_1, e_2]$ を $P$ 、固有値を対角成分に持つ行列を $T$ と書き、左辺の $P$ を右辺に移項すると

(こちらのほうが簡単！  
この式が持つ意味は前述のとおり)



## レポート課題 (1)

$$A = \begin{bmatrix} 3 & -3 \\ 1 & 7 \end{bmatrix}$$

の作用について、 $xy$ 平面上の点群( $X=-3\sim 3, Y=-3\sim 3$ )  
がどのように移動するか、例と同様に試してみることに

またこの移動が、固有ベクトル、固有値の観点から妥当  
であることを確認すること



## 重要な応用： $A^n$

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

$$\begin{aligned} \because \mathbf{T}^n &= \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \dots \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \\ &= \begin{bmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{bmatrix} \end{aligned}$$

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 $A^n$

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

行列の固有値 $\lambda$ の絶対値が引き延ばしの倍率だから、

固有値の絶対値が

- 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： $A^n$

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

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

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

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

$$A^3 = \mathbf{P} \mathbf{T}^3 \mathbf{P}^{-1} = \begin{bmatrix} 1/\sqrt{10} & 2/\sqrt{5} \\ 3/\sqrt{10} & 1/\sqrt{5} \end{bmatrix} \begin{bmatrix} 2^3 & 0 \\ 0 & 7^3 \end{bmatrix} \begin{bmatrix} 1/\sqrt{10} & 2/\sqrt{5} \\ 3/\sqrt{10} & 1/\sqrt{5} \end{bmatrix}^{-1}$$
$$= \dots = \begin{bmatrix} 410 & -134 \\ 201 & -59 \end{bmatrix} \quad \text{固有値が大きいのでどんどん大きくなる}$$

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)

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

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

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

$$\therefore u_x = ju_y \quad \mathbf{e}_1 = k \begin{bmatrix} j \\ 1 \end{bmatrix} \quad \text{大きさを1とすれば例えば, } k = 1/\sqrt{2}$$

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

$$\mathbf{e}_2 = k \begin{bmatrix} 1 \\ j \end{bmatrix} \quad \text{大きさを1とすれば例えば, } k = 1/\sqrt{2}$$



(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

注意:ここで導入する行列は導入編用で, シミュレーションとしては不正確です.

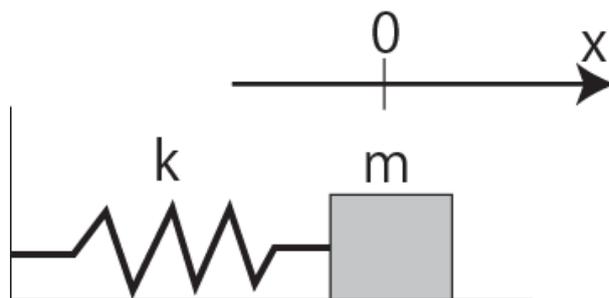
おもりの挙動をシミュレートしたい

```
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
```

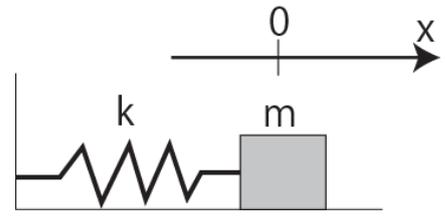
```
for time= 0:dt:10 //時刻
    F=-k*x; //ばねによって生じる力
    a=F/m; //生じる加速度
    v= v+a*dt; //速度
    x= x+v*dt; //位置
    record = [record,x]; //記録(※テクニック:ベクトルが伸びていく)
```

```
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
    
```

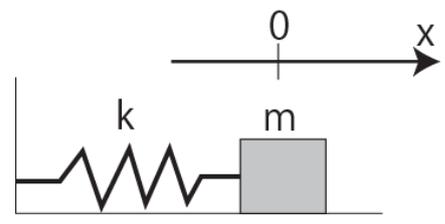
位置, 速度, 加速度を並べた「状態ベクトル」 $x$ を定義  $x = \begin{bmatrix} x \\ v \\ a \end{bmatrix}$

上の関係から,  $dt$ 時間後の新たな位置, 速度, 加速度は

$$\begin{bmatrix} X_n \\ V_n \\ a_n \end{bmatrix} = \begin{bmatrix} \phantom{X_{n-1}} \\ \phantom{V_{n-1}} \\ \phantom{a_{n-1}} \end{bmatrix} \begin{bmatrix} X_{n-1} \\ V_{n-1} \\ a_{n-1} \end{bmatrix} = \mathbf{A} \mathbf{x}$$



# 制御における行列



```

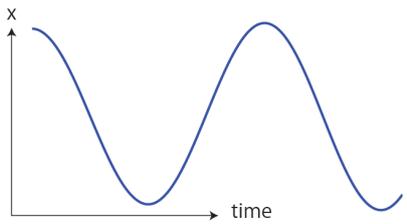
Scilabコード
m=1.0; //重さ
k=1.0; //ばね定数
x=1.0; //初期位置
v=0; //初期速度
a=-k/m*x; //初期加速度
dt=0.01; //時間刻み
record=[]; //記録用

state=[x;v;a];

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);
    
```

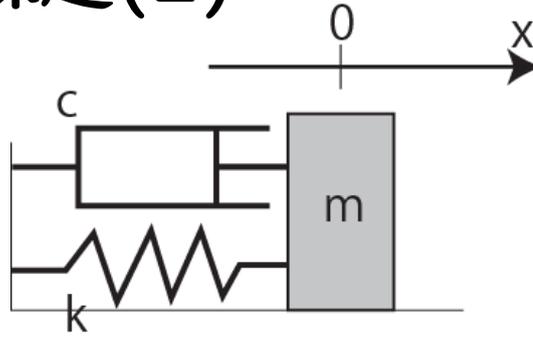


$$\begin{bmatrix} X_n \\ V_n \\ a_n \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 \\ 0 & 1 & dt \\ -k/m & 0 & 0 \end{bmatrix} \begin{bmatrix} X_{n-1} \\ V_{n-1} \\ a_{n-1} \end{bmatrix} = \mathbf{A} \mathbf{x}_{n-1} = \dots = \mathbf{A}^n \mathbf{x}_0$$

- 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=∞)収束するか発散するか予測可能!



# レポート課題(2)



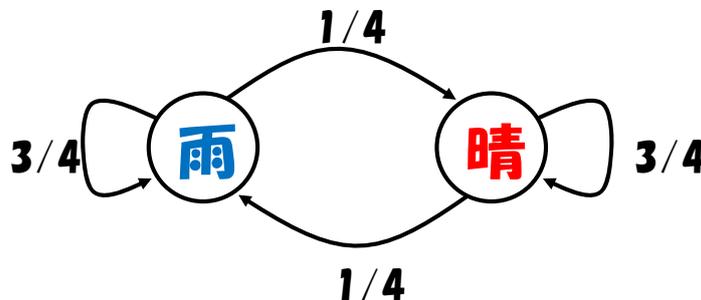
- ダンパを加えた際の行列を考え、同様のシミュレーションプログラムを書け
- 行列Aの固有値の絶対値が1よりも小さいこと、すなわち位置が収束することを確認し、コメントに記せ ( Scilabでは固有値はspec()で求められる)

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。



## 情報理論における行列の例

- ある日晴れた: 次の日も晴れる確率は $3/4$ , 雨になる確率は $1/4$
  - ある日雨: 次の日も雨の確率は $3/4$ , 晴れる確率は $1/4$
- これは「状態遷移図」によって表せる。(マルコフ過程と呼ぶ)



- ある日晴れた⇒次の次の日も晴れる確率は？

(回答)

次の日晴れて, 次の次の日晴れる

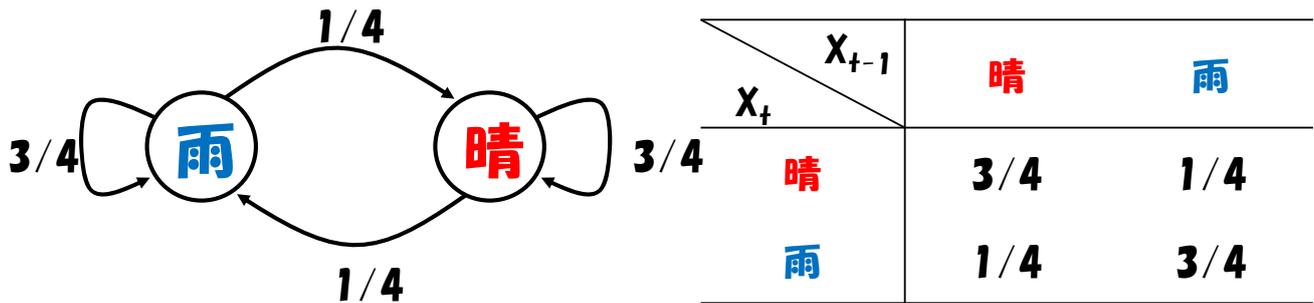
次の日雨になって, 次の次の日晴れる

合わせて,

では, 次の次の次の日だと? 100日後だと?



# 情報理論における行列の例



状態遷移図を行列で表すことで、ある日の天気確率分布( $X_{t-1}$ )から次の日の天気確率分布( $X_t$ )を得ることができる。

次の日の天気確率分布ベクトル

遷移確率行列

ある日の天気確率分布ベクトル

$$\begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \begin{bmatrix} 3/4 & 1/4 \\ 1/4 & 3/4 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix}$$

# 情報理論における行列の例

ある日晴れた: 確率分布ベクトルは  $[1, 0]$  なので、次の日は、

$$\begin{bmatrix} P_{\text{晴}} \\ P_{\text{雨}} \end{bmatrix} = \begin{bmatrix} \phantom{0} & \phantom{0} \\ \phantom{0} & \phantom{0} \end{bmatrix}$$

その次の日は、

$$\begin{bmatrix} P_{\text{晴}} \\ P_{\text{雨}} \end{bmatrix} = \begin{bmatrix} \phantom{0} & \phantom{0} \\ \phantom{0} & \phantom{0} \end{bmatrix}$$

一般にN日後は、

$$\begin{bmatrix} P_{\text{晴}} \\ P_{\text{雨}} \end{bmatrix} = \begin{bmatrix} \phantom{0} & \phantom{0} \\ \phantom{0} & \phantom{0} \end{bmatrix}$$

100日後は？無限日後は？

**$A^N$ を求める。**

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

$$\left\| \begin{bmatrix} 3/4 - \lambda & 1/4 \\ 1/4 & 3/4 - \lambda \end{bmatrix} \right\| = 0$$



**$A^N$ を求める。**

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

=	
=	

これによって、n日後の晴れと雨の確率分布をもとめられる

## 無限日後に起きること

$$\lim_{n \rightarrow \infty} \begin{bmatrix} 1^n & 0 \\ 0 & 1/2^n \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix} \text{より}$$

--	--

--	--

初日の、晴れの確率と雨の確率の合計値は明らかに1だから、

--

つまり、この遷移行列に従う場合、  
初日の天気に関わらず無限日後は晴れと雨の確率は等しくなる。