

# 数値計算 : MATLAB

平井 慎一

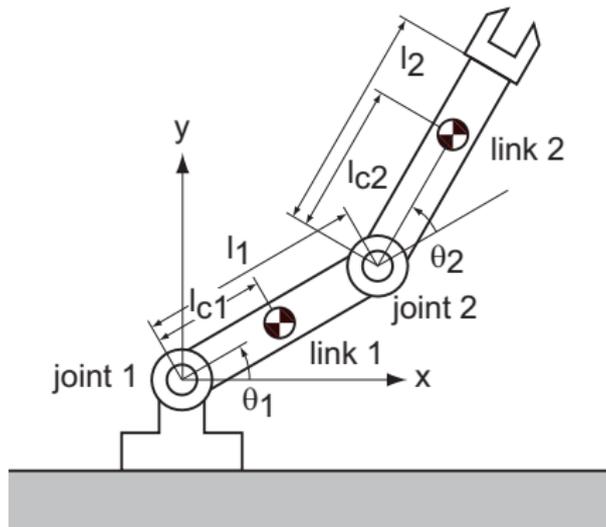
立命館大学 ロボティクス学科

# 講義の流れ

- 1 行列とベクトル
- 2 グラフ
- 3 常微分方程式
- 4 パラメータの引き渡し
- 5 まとめ

# 問題

2リンク開ループ機構を関節PID制御で駆動する。  
このときの動作をシミュレーションせよ。



# 問題

- Step 1. 動作を表す運動方程式を導く（機構学）
- Step 2. 運動方程式を数值的に解く
- Step 3. 数値解をグラフや動画で表す  
（ビジュアライゼーション）
- Step 4. シミュレートした運動を分析する

# 問題

連立一次方程式

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{bmatrix} \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

を解く.

↓

微分方程式

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{bmatrix} = \begin{bmatrix} \alpha_1(\theta_1, \theta_2, \omega_1, \omega_2) \\ \alpha_2(\theta_1, \theta_2, \omega_1, \omega_2) \end{bmatrix}$$

を解く.

# ベクトルと行列

## 列ベクトル

```
x = [ 2; 3; -1 ];
```

## 行ベクトル

```
y = [ 2, 3, -1 ];
```

## 行列

```
A = [ 4, -2, 1; ...  
      -2, 5, 2; ...  
      -2, 3, 2 ];
```

# ベクトルと行列

記号... は文が続くことを表す.

## 列ベクトル

```
x = [ 2; ...  
      3; ...  
     -1 ];
```

## 列ベクトル

```
x = [ 2; 3; -1 ];
```

# ベクトルと行列

## 乗算

```
p = A*x;
```

```
q = y*A;
```

```
>> p
```

```
p =
```

```
1
```

```
9
```

```
3
```

```
>>
```

# ベクトルと行列

## 乗算

```
p = A*x;
```

```
q = y*A;
```

```
>> q
```

```
q =
```

```
     4     8     6
```

```
>>
```

# 行列の操作

```
>> A
```

```
A =
```

```
     4     -2     1  
    -2     5     2  
    -2     3     2
```

```
>> A(3,2)
```

```
ans =
```

```
     3
```

# 行列の操作

```
>> A
```

```
A =
```

```
    4    -2     1  
   -2     5     2  
   -2     3     2
```

```
>> A(3,2) = 6;
```

```
>> A
```

```
A =
```

```
    4    -2     1  
   -2     5     2  
   -2     6     2
```

# 行列の操作

```
>> A
```

```
A =
```

```
     4     -2     1  
    -2     5     2  
    -2     3     2
```

```
>> A(3,:)
```

```
ans =
```

```
    -2     3     2
```

# 行列の操作

```
>> A
```

```
A =
```

```
     4     -2     1  
    -2     5     2  
    -2     3     2
```

```
>> A(:,2)
```

```
ans =
```

```
    -2  
     5  
     3
```

# 行列の操作

```
>> A
```

```
A =
```

```
    4    -2     1  
   -2     5     2  
   -2     3     2
```

```
>> A(:,2) = [ 0; 2; 1 ];
```

```
>> A
```

```
A =
```

```
    4     0     1  
   -2     2     2  
   -2     1     2
```

# 行列の操作

```
>> A
```

```
A =
```

```
     4     -2     1
    -2     5     2
    -2     3     2
```

```
>> A(3,:) = [ 3, -5, -1 ];
```

```
>> A
```

```
A =
```

```
     4     -2     1
    -2     5     2
     3     -5    -1
```

# 行列の操作

```
>> A
```

```
A =
```

```
    4    -2     1  
   -2     5     2  
   -2     3     2
```

```
>> B = A([1,3],:);
```

```
>> B
```

```
B =
```

```
    4    -2     1  
   -2     3     2
```

# 行列の操作

```
>> A
```

```
A =
```

```
     4     -2     1  
    -2     5     2  
    -2     3     2
```

```
>> C = A(:, [2,1]);
```

```
>> C
```

```
C =
```

```
    -2     4  
     5    -2  
     3    -2
```

# 基本行操作

$A(3,:) = 5*A(3,:);$

3行目を5倍する

$A(1,:) = A(1,:) + 4*A(2,:);$

1行目に2行目の4倍を加える

$A([3,1],:) = A([1,3],:);$

1行目と3行目を交換する

## 連立一次方程式を解く

```
A = [ 4, -2, 1; ...  
      -2, 5, 2; ...  
      -2, 3, 2 ];
```

```
p = [ 1; 9; 3 ];
```

連立一次方程式  $Ax = p$  を解く

```
>> x = A\p;
```

```
>> x
```

```
x =
```

```
2
```

```
3
```

```
-1
```

```
>> A*x
```

# 連立一次方程式を解く

スクリプトファイル linear.m

```
A = [ 4, -2, 1; ...  
      -2, 5, 2; ...  
      -2, 3, 2 ];  
p = [ 1; 9; 3 ];  
x = A \ p;  
x
```

を作成し、実行せよ。

```
>> A*x
```

を実行し、解を確認せよ。

# 連立一次方程式を解く

- オペレータ \ は汎用であるが効率的ではない
- 係数行列が正定対称の場合は，コレスキー分解を使う
- 慣性行列は正定対称

## コレスキー分解

正定対称行列  $A$  は，上三角行列  $U$  を用いて

$$A = U^T U$$

と分解できる

$$Ax = p \implies U^T Ux = p \implies \begin{cases} U^T y = p \\ Ux = y \end{cases}$$

# コレスキークー分解

プログラム Cholesky.m

```
fprintf('Cholesky decomposition\n');
```

```
A = [ 4, -2, -2; ...  
      -2,  2,  0; ...  
      -2,  0,  3 ];
```

```
U = chol(A);
```

```
U
```

```
U'*U
```

# コレスキー分解

```
>> Cholesky  
Cholesky decomposition
```

```
U =
```

```
    2    -1    -1  
    0     1    -1  
    0     0     1
```

```
ans =
```

```
    4    -2    -2  
   -2     2     0  
   -2     0     3
```

# コレスキー分解

## プログラム

```
p = [ 6; -4; 0 ];  
y = U'\p;  
x = U\y;  
x
```

## 計算結果

```
x =  
    3  
    1  
    2
```

# グラフ

```
>> x = [0:10]'
```

```
x =
```

```
0
```

```
1
```

```
2
```

```
3
```

```
...
```

```
>> f = x.*x
```

```
f =
```

```
0
```

```
1
```

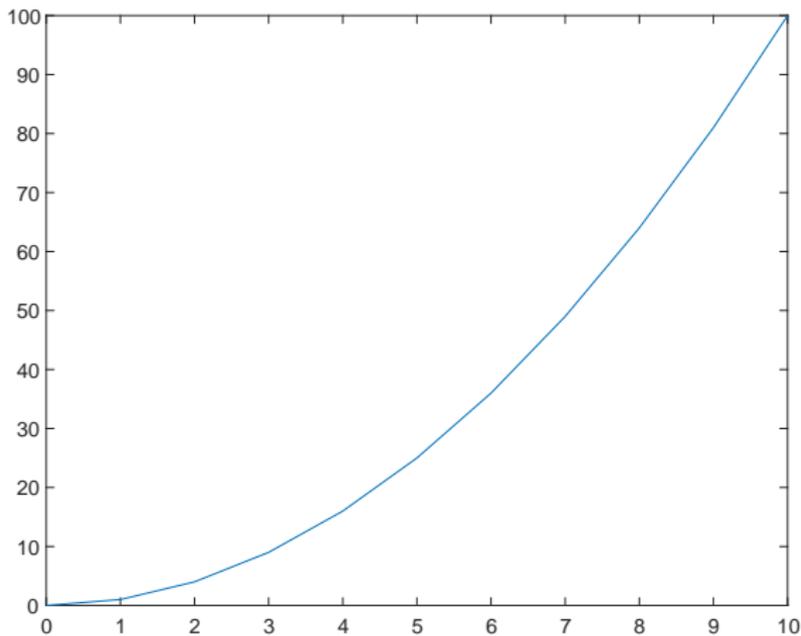
```
4
```

```
9
```

```
...
```

# グラフ

```
>> plot(x,f)
```



# 要素単位の演算

演算子 `.*` や `./` は、要素単位で乗算や除算を実行

$$\begin{bmatrix} 2 \\ 5 \\ -3 \end{bmatrix} .* \begin{bmatrix} 3 \\ -1 \\ -3 \end{bmatrix} = \begin{bmatrix} 6 \\ -5 \\ 9 \end{bmatrix}$$

$$\begin{bmatrix} 6 \\ -5 \\ 1 \end{bmatrix} ./ \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \\ 1/2 \end{bmatrix}$$

# グラフ

```
>> t = [0:0.1:10]'
```

```
t =
```

```
    0
```

```
  0.1000
```

```
  0.2000
```

```
  0.3000
```

```
  ...
```

```
>> x = sin(t)
```

```
x =
```

```
    0
```

```
  0.0998
```

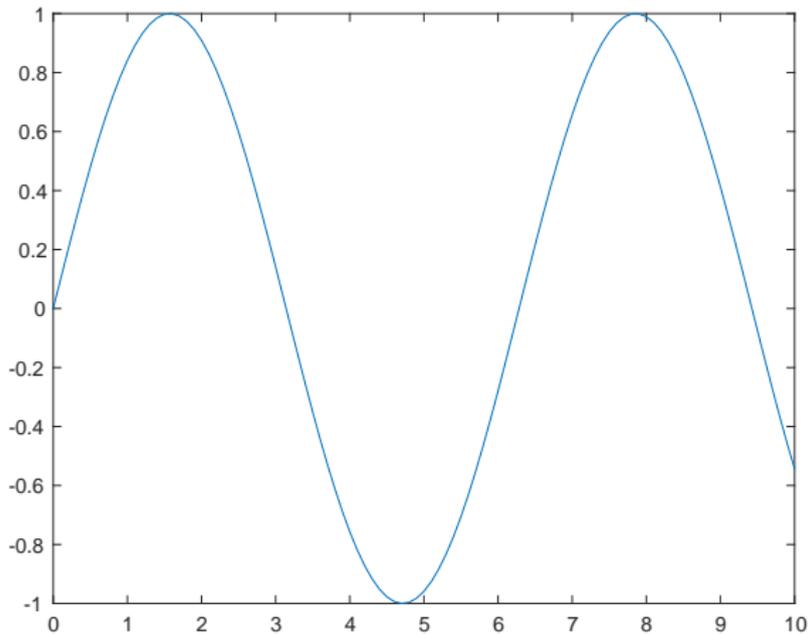
```
  0.1987
```

```
  0.2955
```

```
  ...
```

# グラフ

```
>> plot(t,x)
```



# ベクトル化関数

関数 **cos**, **sin**, **exp**, **log** 等は, ベクトルを引数とすることができる.

$$\sin \begin{bmatrix} 0 \\ \pi/6 \\ \pi/3 \end{bmatrix} = \begin{bmatrix} \sin(0) \\ \sin(\pi/6) \\ \sin(\pi/3) \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \\ \sqrt{3}/2 \end{bmatrix}$$

$$\exp \begin{bmatrix} 0 \\ \log 2 \\ \log 3 \end{bmatrix} = \begin{bmatrix} \exp(0) \\ \exp(\log 2) \\ \exp(\log 3) \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

# グラフ

ファイル draw\_graph.m

```
t = [0:0.1:10]';  
x = sin(t);  
plot(t,x);  
title('グラフ');      % グラフの表題  
xlabel('time');        % 横軸のラベル  
ylabel('position');   % 縦軸のラベル  
ylim([-1.5,1.5]);    % 縦軸の範囲  
saveas(gcf,'draw_sine_graph.png'); % グラフの保存
```

ファイル draw\_graph.m を実行すると、グラフを描き、描いたグラフをファイルに保存する。

# 常微分方程式を数值的に解く

ファンデルポール (van der Pol) 方程式

$$\ddot{x} - 2(1 - x^2)\dot{x} + x = 0$$

⇓

$$\begin{cases} \dot{x} = v \\ \dot{v} = 2(1 - x^2)v - x \end{cases}$$

⇓

$$\mathbf{q} = \begin{bmatrix} x \\ v \end{bmatrix}, \quad \dot{\mathbf{q}} = \mathbf{f}(t, \mathbf{q}) = \begin{bmatrix} v \\ 2(1 - x^2)v - x \end{bmatrix}$$

# 常微分方程式を数值的に解く

関数  $f(t, \mathbf{q})$  を定義するファイル `van_der_Pol.m`

ファイルの名前 "`van_der_Pol`" と関数の名前 "`van_der_Pol`" を一致させる

```
function dotq = van_der_Pol (t, q)
    x = q(1);
    v = q(2);
    dotx = v;
    dotv = 2*(1-x^2)*v - x;
    dotq = [dotx; dotv];
end
```

# 常微分方程式を数值的に解く

関数ファイル `van_der_Pol.m` を作成せよ.

```
>> q = [ 2; 0]
```

```
>> van_der_Pol(0,q)
```

を実行せよ. 作成した関数を用いることができる.

# 常微分方程式を数值的に解く

スクリプトプログラム `van_der_Pol_solve.m`

```
interval = 0.00:0.10:10.00;  
qinit = [ 2.00; 0.00 ];  
[time, q] = ode45(@van_der_Pol, interval, qinit);
```

ファイル `van_der_Pol_solve.m` を作成し, 実行せよ.

# 常微分方程式を数值的に解く

時刻  $t$  と変数  $x$  の関係をグラフで表す

```
plot(time, q(:,1), '-');
```

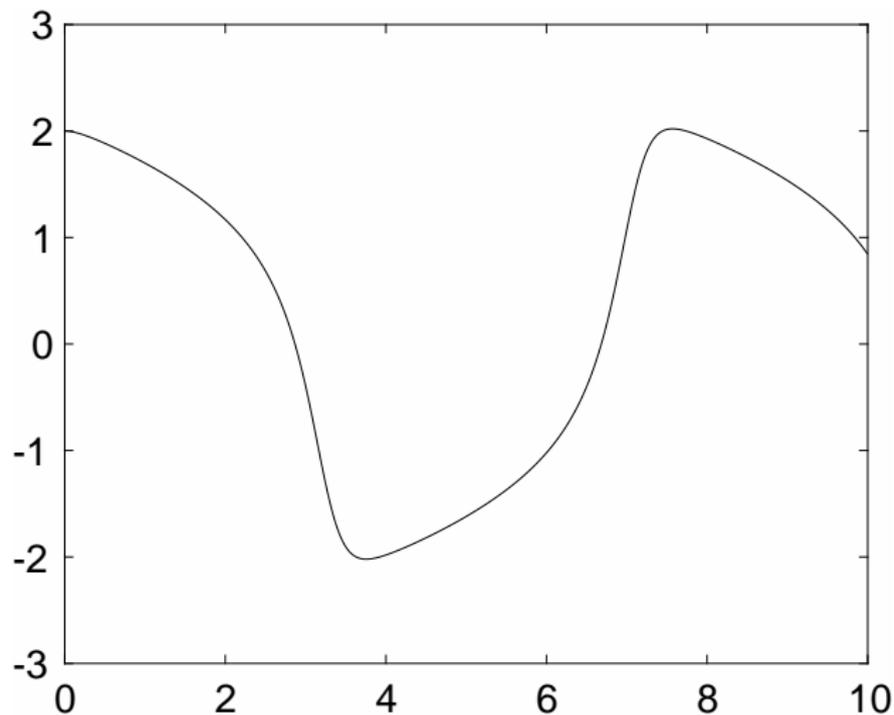
時刻  $t$  と変数  $v$  の関係をグラフで表す

```
plot(time, q(:,2), '-');
```

- '-' 実線
- '--' 破線
- '-.' 一点破線
- ':' 点線

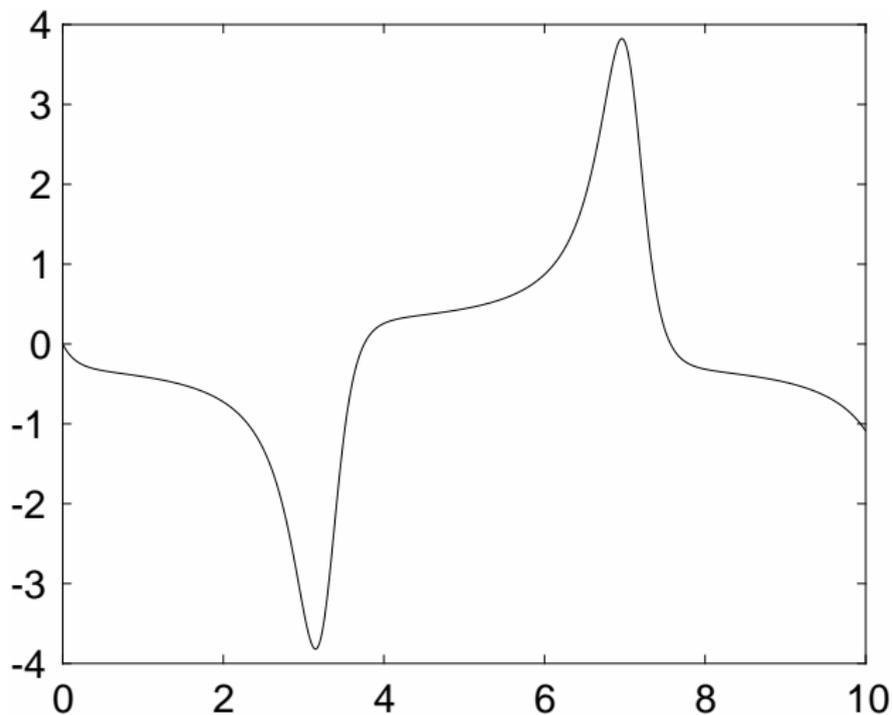
# 常微分方程式を数値的に解く

時刻  $t$  と変数  $x$  のグラフ



# 常微分方程式を数值的に解く

時刻  $t$  と変数  $v$  のグラフ



# パラメータを有する常微分方程式

微分方程式

$$\ddot{x} + b\dot{x} + 9x = 0$$

$b$  はパラメータ

↓

$$\dot{x} = v$$

$$\dot{v} = -bv - 9x$$

# 大域変数

## 関数

```
function dotq = damped_vibration_global (t, q)
    global b;
    x = q(1); v = q(2);
    dotx = v; dotv = -b*v - 9*x;
    dotq = [dotx; dotv];
end
```

## プログラム

```
global b;
interval = [0,10];
qinit = [2.00;0.00];
b = 1.00;
[time,q] = ode45(@damped_vibration_global,interval,qinit)
```

# 入れ子関数

時刻, 状態変数ベクトル, パラメータを引数とする関数

```
function dotq = damped_vibration_param (t, q, b)
    x = q(1); v = q(2);
    dotx = v; dotv = -b*v - 9*x;
    dotq = [dotx; dotv];
end
```

プログラム

```
interval = [0,10];
qinit = [2.00;0.00];
b = 1.00;
damped_vibration = @(t,q) damped_vibration_param (t,q,b);
[time,q] = ode45(damped_vibration,interval,qinit);
```

# 大域変数 vs 入れ子関数

## 大域変数

プログラムが単純

大域変数が他の変数と重複する恐れがある

## 入れ子関数

プログラムがやや複雑

パラメータの値が変わるたびに、関数定義を再実行する必要

他の変数と重複しない

# まとめ

## MATLAB による数値計算

- ベクトルと行列の計算
- 連立一次方程式を解く
- 常微分方程式を数値的に解く
- パラメータの引き渡し

# 付録

## 違いに注意

- 、 コンマ 要素の区切り
- ； セミコロン 文の最後，列ベクトル (dotq, qinit)
- ⋮ コロン 等間隔の要素列 (interval)
- 。 ピリオド

# レポート (MATLAB Grader)

MATLAB grader 「MATLAB の使い方の練習」

締切：2026年5月11日（月曜）01:00 AM

- ① 「スクリプト」欄に解答を入力
- ② 「スクリプトを実行」をクリックし、解答を実行
- ③ 「出力」欄で解答を確認
- ④ 「提出」をクリックし、解答を提出  
(解答の提出回数の制限に注意)

<https://jp.mathworks.com/> の「サイト内検索」で  
「単位行列」や「ゼロ行列」を検索せよ。