

# **MATLAB講習**

外本・坂東研究室

# MATLABとは

米Mathwork社が開発する技術計算のための高性能言語

## 特徴

- 対話形式でひとつずつ命令を実行することも、命令をまとめてプログラムすることもできる
- 変数宣言が不要で、C言語などに比べて少ない労力で数値計算を実装しやすい
- ベクトルや行列計算に向いている
- 関数ライブラリ、グラフィックツールが豊富で、グラフ表示機能なども充実している

# 画面

The image shows the MATLAB R2023a - academic use interface. The top menu bar includes Home, Plots, Apps, Editor, Publish, and View. The Editor tab is active, showing a script named ULMC.m. The left sidebar displays the current folder (MATLAB Drive > Path integral control) and a list of files. The right sidebar shows the Workspace with variables like dt, dW, i, k, N, T, theta, u, x, and x0. The bottom status bar shows Zoom: 100%, UTF-8, CRLF, and ULMC.

⑤ ツールストリップ

③ エディター

④ 現在のフォルダー

② ワークスペース

① コマンドウィンドウ

```
function [E,J] = ULMC(funcEOM,funcU,funcV,funcphi,tspan,y0a,y0b)
Nsteps = size(tspan,2);
indims = ndims(y0a);
insize = size(y0a);

for jj = 1:Nsteps-1
    t0 = tspan(jj);
    h = tspan(jj+1)-tspan(jj);

    if indims == 1
        dw = sqrt(h)*gpuArray.randn(insize(1),1);
    elseif indims == 2
        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2));
    elseif indims == 3
        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2),insize(3));
    elseif indims == 4
        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2),insize(3),insize(4));
    end
```

名前	値
dt	0.0100
dW	[0.0778;-0.0549]
i	999
k	10
N	1000
T	10
theta	1.5708
u	1x999 double
x	2x1000 double
x0	[1;0]

>> RMW  
fx >>

Zoom: 100% UTF-8 CRLF ULMC 行 1 列 1

# ①コマンドウィンドウ

The image shows the MATLAB R2023a - academic use interface. The components are labeled as follows:

- ⑤ ツールストリップ (Toolstrip): Located at the top, containing tabs for Home, Plots, Apps, Editor, Publish, and View, along with various tool icons.
- ③ エディター (Editor): The central workspace for writing and editing code, showing a script named ULMC.m.
- ④ 現在のフォルダー (Current Folder): A panel on the left showing the file explorer for the current directory (MATLAB Drive > Path integral control).
- ② ワークスペース (Workspace): A panel on the right displaying the current workspace variables and their values.
- ① コマンドウィンドウ (Command Window): A panel at the bottom for entering and executing commands.

The Editor window displays the following code:

```
function [E,J] = ULMC(funcEOM,funcU,funcV,funcphi,tspan,y0a,y0b)
1
2 Nsteps = size(tspan,2);
3 indims = ndims(y0a);
4 insize = size(y0a);
5
6 for jj = 1:Nsteps-1
7     t0 = tspan(jj);
8     h = tspan(jj+1)-tspan(jj);
9
10    if indims == 1
11        dw = sqrt(h)*gpuArray.randn(insize(1),1);
12    elseif indims == 2
13        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2));
14    elseif indims == 3
15        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2),insize(3));
16    elseif indims == 4
17        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2),insize(3),insize(4));
```

The Workspace window shows the following variables:

名前	値
dt	0.0100
dW	[0.0778;-0.0549]
i	999
k	10
N	1000
T	10
theta	1.5708
u	1x999 double
x	2x1000 double
x0	[1;0]

The Command Window shows the following commands:

```
>> RMW
fx >>
```

# ①コマンドウィンドウ

## コマンドウィンドウとは？

- コマンドウィンドウは、MATLABでの計算やプログラムの実行結果、エラーメッセージの表示、変数の表示や操作などに利用されます。
- コマンドウィンドウはわざわざプログラムとして記述するほどでもない簡単な計算を行う際や変数の表示、操作を行う際に使用することが多いです。

# ①コマンドウィンドウ

## コマンドウィンドウでの基本的な操作

- コマンドウィンドウにコマンドを打ち込み, Enterキーを押すことでコマンドやプログラムを実行することができます.

```
>> 1+3
```

```
ans =
```

4

```
>> a=2;
```

```
>> b=3;
```

```
>> c=a+b
```

```
c =
```

5

```
>> pi
```

```
ans =
```

3.141592653589793

行の最後に  
「;」をつけると,  
結果が  
非表示になる

代入操作以外の実行結果は,  
「ans」という変数に格納される

# ①コマンドウィンドウ

```
>> a=[1 2 3]
```

```
a =
```

```
    1    2    3
```

```
>> b=[4;5;6]
```

```
b =
```

```
    4  
    5  
    6
```

列の区切りは「 」もしくは「,」  
行の区切りは「;」

```
>> A=[1 2;3 4]
```

```
A =
```

```
    1    2  
    3    4
```

大文字と小文字は  
区別される

# ②ワークスペース

The image shows the MATLAB R2023a interface with several components highlighted by green boxes and numbered:

- ⑤ ツールストリップ**: The top toolbar area containing icons for file operations, editing, and execution.
- ④ 現在のフォルダー**: The left sidebar showing the current folder structure, including '3\_Compare Sampling', 'Normal', 'Time step', 'mainNS.m', 'setup.m', 'setup\_NSSI.mat', and 'ULMC.m'.
- ③ エディター**: The central editor window displaying the code for 'ULMC.m'.
- ② ワークスペース**: The right sidebar showing the workspace variables and their values.
- ① コマンドウィンドウ**: The bottom-left window for entering commands.

The workspace table is as follows:

名前	値
dt	0.0100
dW	[0.0778; -0.0549]
i	999
k	10
N	1000
T	10
theta	1.5708
u	1x999 double
x	2x1000 double
x0	[1;0]

The command window shows the following commands:

```
>> RMW  
fx >>
```



## ②ワークスペース

- ワークスペースには、現在のMATLABセッションにおいて定義された**すべての変数が表示されます**。
- ワークスペースにある変数はclearを行うか、MATLABを終了しない限り残り、**以降の計算に使用可能です**。
- ワークスペースにある変数を**MATLAB終了後に保持する場合、データを保存する必要があります**。

# ③エディター

The image shows the MATLAB R2023a interface with several components highlighted by green boxes and numbered labels:

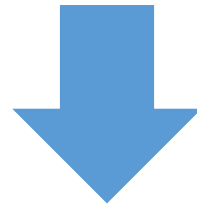
- ⑤ ツールストリップ** (Toolstrip): Located at the top of the interface, containing tabs for Home, Plots, Apps, Editor, Publish, and View. It also includes various icons for file operations and execution.
- ④ 現在のフォルダー** (Current Folder): Located on the left side, showing the current directory structure. The file `ULMC.m` is selected.
- ③ エディター** (Editor): The central area where the code for `ULMC.m` is written. The code is as follows:

```
1 function [E,J] = ULMC(funcEOM,funcU,funcV,funcphi,tspan,y0a,y0b)
2 Nsteps = size(tspan,2);
3 indims = ndims(y0a);
4 insize = size(y0a);
5
6 for jj = 1:Nsteps-1
7     t0 = tspan(jj);
8     h = tspan(jj+1)-tspan(jj);
9
10    if indims == 1
11        dw = sqrt(h)*gpuArray.randn(insize(1),1);
12    elseif indims == 2
13        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2));
14    elseif indims == 3
15        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2),insize(3));
16    elseif indims == 4
17        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2),insize(3),insize(4));
```
- ② ワークスペース** (Workspace): Located on the right side, showing the current workspace variables. The variables and their values are:

名前	値
dt	0.0100
dW	[0.0778;-0.0549]
i	999
k	10
N	1000
T	10
theta	1.5708
u	1x999 double
x	2x1000 double
x0	[1;0]
- ① コマンドウィンドウ** (Command Window): Located at the bottom left, showing the command prompt. The current command is `>> RMW`.

### ③エディター

実際の作業ではいちいちコマンドウィンドウに打ち込むのは面倒だし、編集ができない



「\*\*\*\*\*.m」(M-ファイル)を作ると、  
ファイルの内容がコマンドラインから実行できる。  
様々な処理をまとめて、ひとつの関数として定義することができる

手順

1. エディターを開く
2. 関数を書く
3. 保存して実行, 確認する

# ④現在のフォルダー

The image shows the MATLAB R2023a interface with several components highlighted by green boxes and numbered:

- ⑤ ツールストリップ**: The top toolbar area containing icons for file operations, editing, and execution.
- ③ エディター**: The central code editor window displaying the `ULMC.m` script.
- ② ワークスペース**: The workspace window on the right showing variables and their values.
- ① コマンドウィンドウ**: The command window at the bottom left for entering MATLAB commands.
- ④ 現在のフォルダー**: The current folder browser on the left, showing the file structure of the current directory.

The code in the editor is as follows:

```
function [E,J] = ULMC(funcEOM,funcU,funcV,funcphi,tspan,y0a,y0b)
1
2 Nsteps = size(tspan,2);
3 indims = ndims(y0a);
4 insize = size(y0a);
5
6 for jj = 1:Nsteps-1
7     t0 = tspan(jj);
8     h = tspan(jj+1)-tspan(jj);
9
10    if indims == 1
11        dw = sqrt(h)*gpuArray.randn(insize(1),1);
12    elseif indims == 2
13        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2));
14    elseif indims == 3
15        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2),insize(3));
16    elseif indims == 4
17        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2),insize(3),insize(4));
```

The workspace variables are:

名前	値
dt	0.0100
dW	[0.0778;-0.0549]
i	999
k	10
N	1000
T	10
theta	1.5708
u	1x999 double
x	2x1000 double
x0	[1;0]

The command window shows:

```
>> RMW
fx >>
```

## ④現在のフォルダー

現在のフォルダーとは、MATLABで作業中のフォルダーを示しています。MATLABではこの現在のフォルダーからファイルを読み込んだり、書き込んだりします。

- プログラムの保存



- フォルダーの作成  
振り返りができるように、  
日付ごとにフォルダーを作成し、  
前回のフォルダーを  
コピーしてから編集するとよい。

名前	更新日
200427	2020/0
200428	2020/0
200429	2020/0
200430	2020/0
200501	2020/0
200502	2020/0
200504	2020/0

# ⑤ ツールストリップ

The image shows the MATLAB R2023a - academic use interface. The top toolbar is highlighted with a red box and labeled ⑤ ツールストリップ (Toolstrip). The left sidebar is labeled ④ 現在のフォルダー (Current Folder). The central editor window is labeled ③ エディター (Editor) and contains the following code:

```
function [E,J] = ULMC(funcEOM,funcU,funcV,funcphi,tspan,y0a,y0b)
1
2 Nsteps = size(tspan,2);
3 indims = ndims(y0a);
4 insize = size(y0a);
5
6 for jj = 1:Nsteps-1
7     t0 = tspan(jj);
8     h = tspan(jj+1)-tspan(jj);
9
10    if indims == 1
11        dw = sqrt(h)*gpuArray.randn(insize(1),1);
12    elseif indims == 2
13        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2));
14    elseif indims == 3
15        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2),insize(3));
16    elseif indims == 4
17        dw = sqrt(h)*gpuArray.randn(insize(1),insize(2),insize(3),insize(4));
18    end
19    y = funcU(t0,y0a,dw);
20    y0a = y;
21    y0b = funcV(t0,y0b,dw);
22    E(jj) = funcphi(t0,y0a,y0b);
23    J(jj) = funcphi(t0,y0a,y0b);
24 end
```

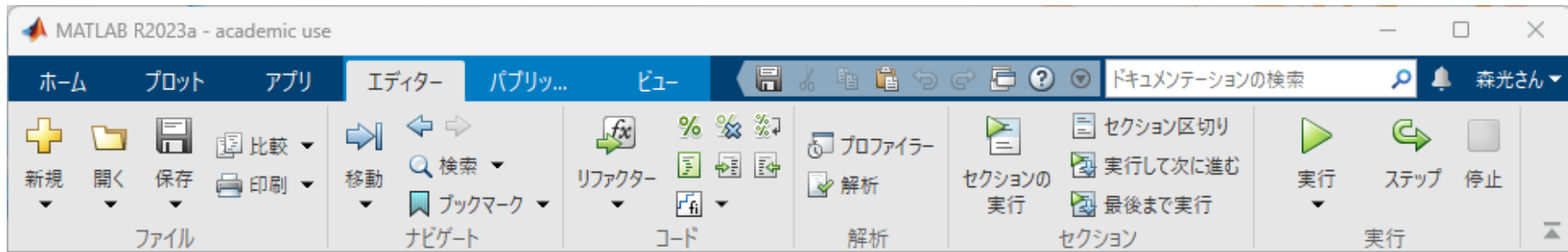
The right sidebar is labeled ② ワークスペース (Workspace) and contains the following variables:

名前	値
dt	0.0100
dW	[0.0778;-0.0549]
i	999
k	10
N	1000
T	10
theta	1.5708
u	1x999 double
x	2x1000 double
x0	[1;0]

The bottom window is labeled ① コマンドウィンドウ (Command Window) and contains the following text:

```
>> RMW
fx >>
```

# ⑤ ツールストリップ



- MATLAB上部にあるメニューバー
- ファイルを開いたり, プログラムを実行したりするのに使うことが多い

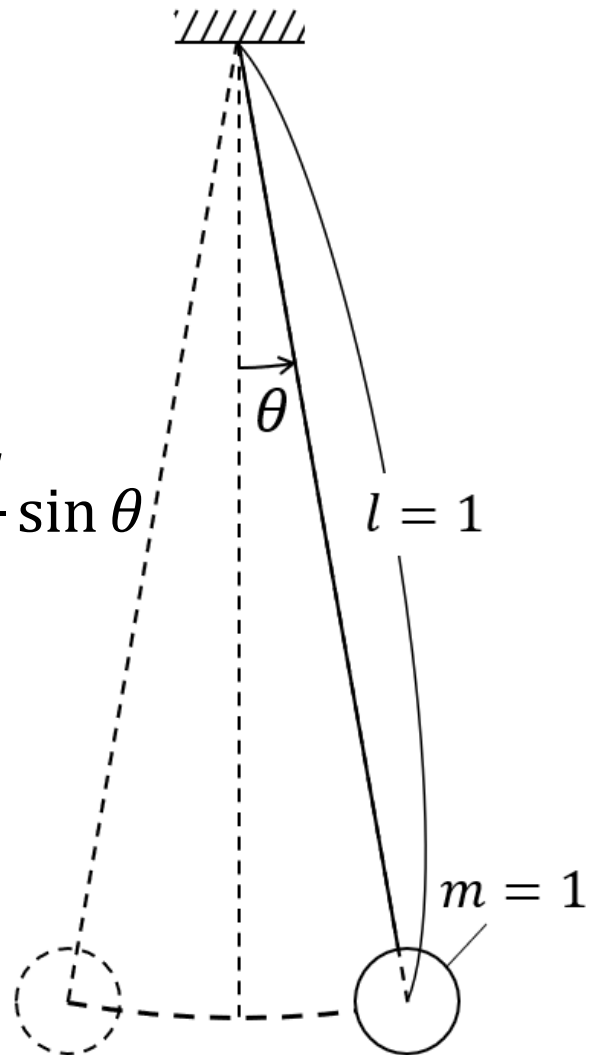
# 実際のプログラムの例

常微分方程式 (Ordinary Differential Equation, ODE) を解く

例) 単振子の非線形微分方程式をODEの  
数値解法で解く

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l} \sin \theta \quad \xrightarrow{\text{一階微分方程式に変換}} \begin{cases} \frac{d\theta}{dt} = \dot{\theta} \\ \frac{d\dot{\theta}}{dt} = -\frac{g}{l} \sin \theta \end{cases}$$

非線形微分方程式は解析的に解く方法が  
存在しないため数値解法を用いる





# 実際のプログラムの例

```
1 %% Reset
2 clear % clear variables
3 close % close figure windows
4
5 %% Input problem settings
6 l = 1; % pendulum length[m]
7 g = 9.80655; % gravitational acceleration[m/s^2]
8 T = [0,5]; % simulation time[s]
9 x0 = [1;0]; % initial condition
10
11 %% Simulate
12 [tspan,x] = ode45(@(t,x) eq_pend(t,x,l,g),T,x0); % solve ODE
13
14 %% plot
15 plot(tspan,x) % plot a result
```

変数を全部削除し、  
開いている図を閉じる

最初に使用する  
変数を定義する

“ode45”というMATLABにある  
ODEを数値計算する関数を使う

# 実際のプログラムの例

```
14 %% plot
15 plot(tspan,x) % plot a result
16 legend("angle","angular velocity") % show legends
17 xlabel("time [s]") % show a horizontal axis label
18 ylabel("state") % show a vertical axis label
19
20 %% save
21 save("result.mat","tspan","x"); % save a result (numerical data)
22 savefig("result.fig"); % % save a result (figure data)
23
24 %% function
25 function dxdt = eq_pend(t,x,l,g) % write ODE
26 dxdt = [x(2);-g/l*sin(x(1))];
27 end
```

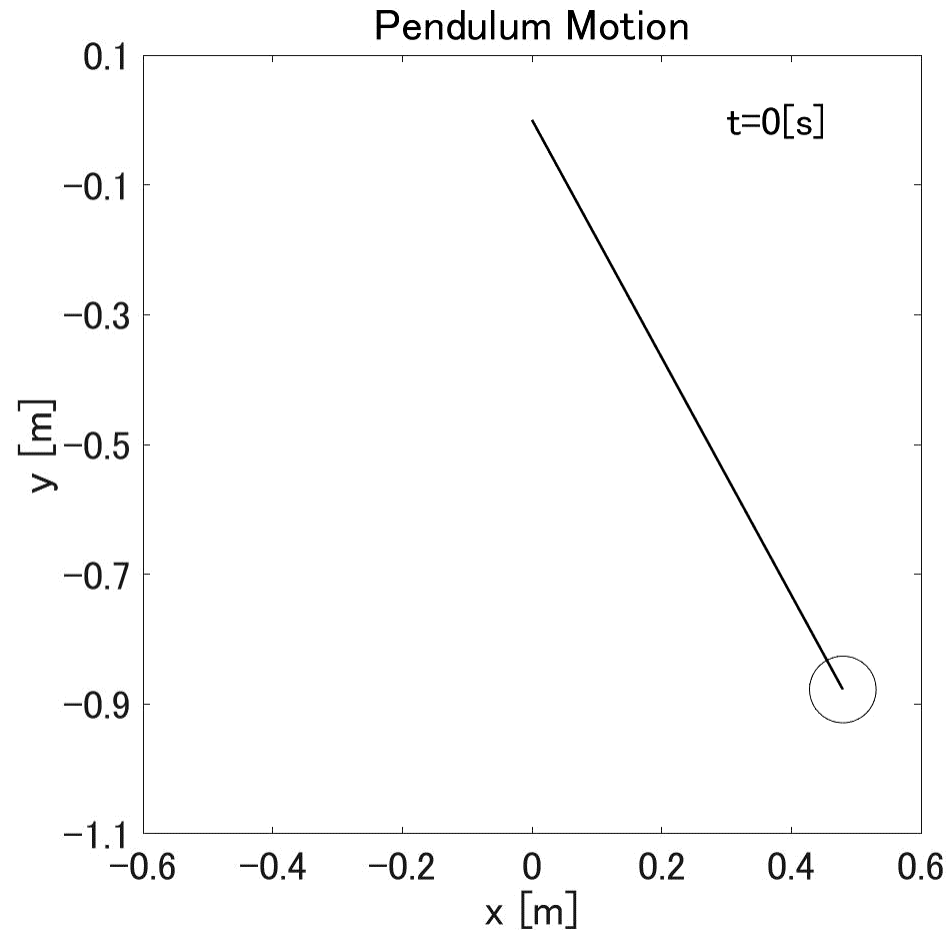
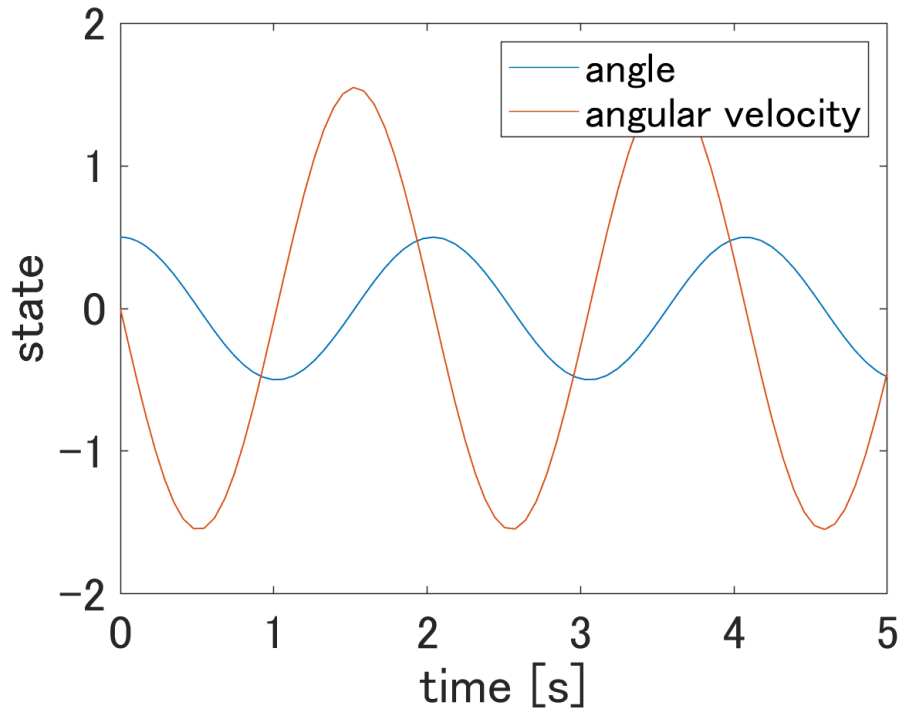
← 得られた結果を使って  
図を作成する

← 計算結果, 図を  
保存する

← 解く関数を  
記述している

# 実際のプログラムの例

## 結果



# 実際のプログラムの例

## ode45の説明

```
[tspan,x] = ode45(@(t,x) eq_pend(t,x,l,g),T,x0);  
  
function dxdt = eq_pend(t,x,l,g)  
dxdt = [x(2);-g/l*sin(x(1))];  
end
```

[評価点, 解] = ode45(解を求める関数, 積分区間, 初期条件)

### 【解を求める関数】

- 関数ハンドル(@をつける)を用いる.
- @の後ろにある()の中の引数(t,x)をそのままに関数として渡す.  
odeのプログラムの中で代入し, 計算を行う.
- ()の中にない引数(l,g)はここで値を代入する.

# 実際のプログラムの例

## ode45の説明

```
[tspan,x] = ode45(@(t,x) eq_pend(t,x,l,g),T,x0);  
  
function dxdt = eq_pend(t,x,l,g)  
    dxdt = [x(2);-g/l*sin(x(1))];  
end
```

[評価点, 解] = ode45(解を求める関数, 積分区間, 初期条件)

### 【積分区間】

- 積分区間を $[t_0, t_f]$ の順に指定する.
- $[t_0, \dots, t_f]$ のように3つ以上指定した場合, 指定した点でのみ値を返す.

# 関数の調べ方

関数の説明はMathWorksのサイトで調べると出てくる



製品
ソリューション
アカデミア
サポート
コミュニティ
イベント

MATLAB を入手する



ヘルプ センター

ode45

ヘルプセンター

🔍

三

ドキュメンテーション

例

関数

アプリ

ビデオ

回答集

📄

評価版

📦

製品の更新

ode45

R2023a

ページ内をすべて折りたたむ

構文

```
[t,y] = ode45(odefun,tspan,y0)
[t,y] = ode45(odefun,tspan,y0,options)
[t,y,te,ye,ie] = ode45(odefun,tspan,y0,options)
sol = ode45( __ )
```

説明

[t,y] = ode45(odefun,tspan,y0)

例

[t,y] = ode45(odefun,tspan,y0) は、tspan = [t0 tf] のときに、初期条件 y0 を使用して、微分方程式系  $y' = f(t, y)$  を t0 から tf まで積分します。解の配列 y の各行は、列ベクトル t に返される値に対応します。

すべての MATLAB® ODE ソルバーは、 $y' = f(t, y)$  の形式の方程式系、あるいは質量行列  $M(t, y)y' = f(t, y)$  を含む問題を解くことができます。すべてのソルバーは類似した構文を使用します。ode23s ソルバーは、質量行列が定数である場合にのみ、これを含む問題を解くことができます。ode15s および ode23t は、特異質量行列をもつ方程式、つまり微分代数方程式 (DAE) を解くことができます。odeset の Mass オプションを使用して質量行列を指定します。

ode45 は汎用的な ODE ソルバーであり、ほとんどの問題に対して最初に試すべきソルバーです。ただし、問題がスティッフである場合や高い精度が要求される場合、他の ODE ソルバーの方が適していることもあります。詳細については、ODE ソルバーの選択を参照してください。

[t,y] = ode45(odefun,tspan,y0,options)

例

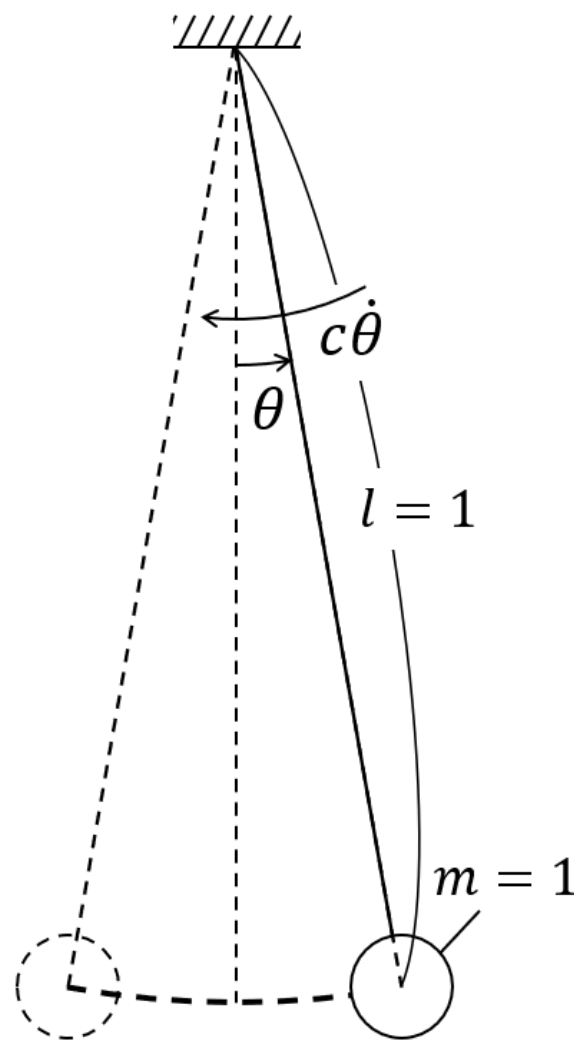
[t,y] = ode45(odefun,tspan,y0,options) は options (関数 odeset を使用して作成された引数) で定義された積分設定も使用します。たとえば、AbsTol オプションおよび RelTol オプションを使用して絶対許容誤差と相対許容誤差を指定したり、Mass オプションを使用して質量行列を指定することができます。

# 演習

先ほどの振り子に摩擦抵抗を追加した  
右の図のような振り子の時間履歴をプロット  
してください. ただし, シミュレーション条件は  
以下のようにします.

- 摩擦抵抗  $c = 0.3$
- 評価点は0から10秒までの間を0.1秒刻み
- 初期状態は  $\theta = 1$ ,  $\dot{\theta} = -0.1$

Tips: 0から10までを0.1で刻んだ行列を  
作るには, "0:0.1:10"と入力するとよい

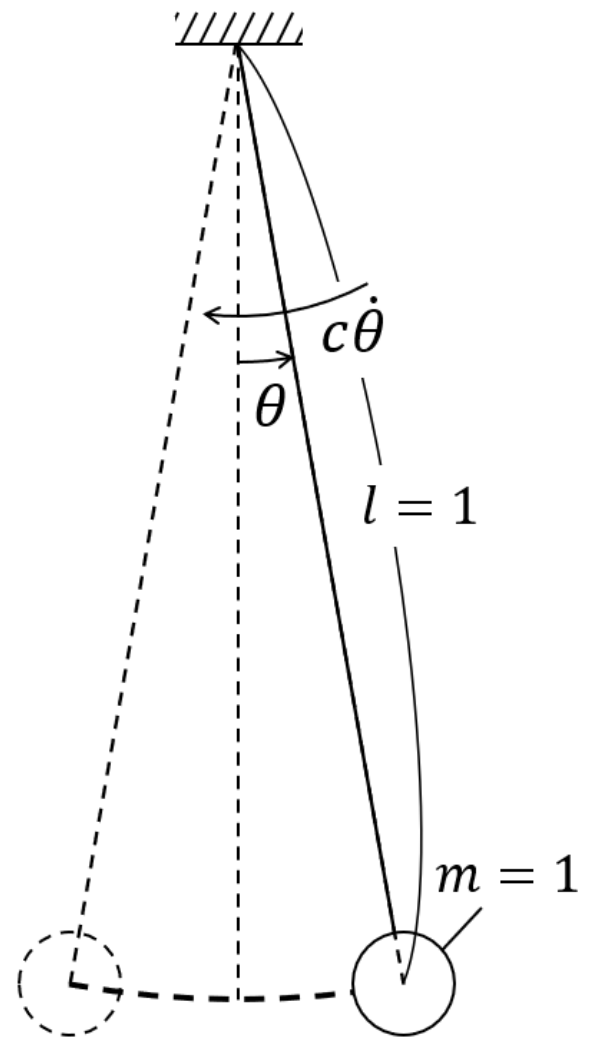
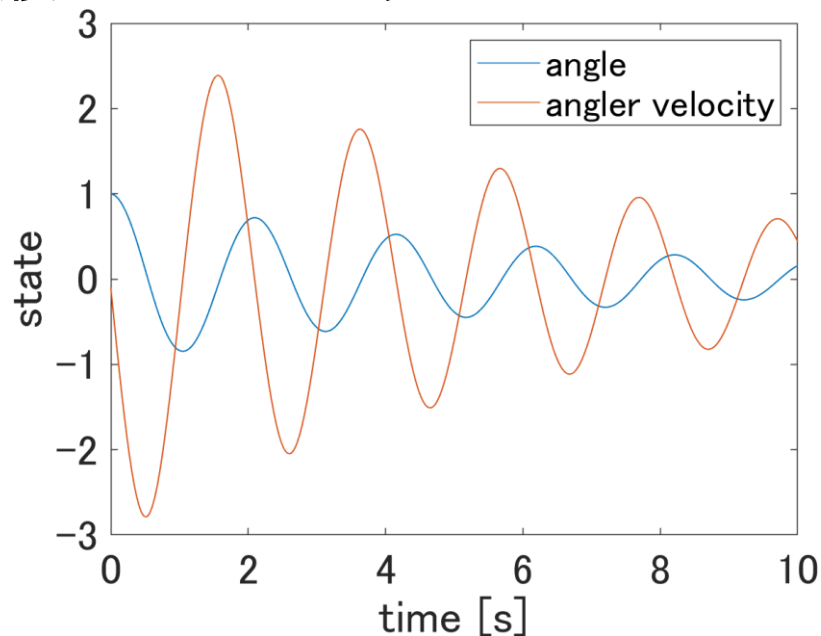


# 演習

ODEは次のようになる.

$$\begin{cases} \frac{d\theta}{dt} = \dot{\theta} \\ \frac{d\dot{\theta}}{dt} = -c\dot{\theta} - \frac{g}{l} \sin \theta \end{cases}$$

時間履歴は次のようになる.





# 参考になるサイト

- MathWorksのサイト
  - 関数の説明のほか、参考となるプログラムも例として公開されている
- Qiita
  - プログラムが説明付きで見つかる
- GitHub
  - 膨大な数のプログラムが公開されている
  - ソースコードを見れない(暗号化されている)場合がある
  - 詳しくは塩塚君に聞いて
- ChatGPT
  - 書いてほしいプログラムを説明してMATLABで書いてというと、書いてもらえる
  - よくある問題は正確に書いてくれるが、マイナーな問題は無理
  - ChatGPTに練習問題を作ってもらうこともできる

# MATLAB課題

- 全6問を1人1人に割り当てる
- MATLABでプログラムを作成し、その結果をプレゼンにまとめてゼミで発表する
- 発表時間は1人8分程度で発表後に質疑応答を行う
- 問題は以下の6つ
  - Van der Pol方程式の解析
  - Lorenz方程式の解析
  - バネ・マス系に振子を取り付けたものの運動
  - 二重振子の運動
  - 倒立振子の制御
  - 2自由度ばねマスダンパ系の運動

# Van der Pol方程式の解析

非線形力学系であるVan der Pol方程式は次式で表される.

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

この方程式について以下の問いに答えよ.

1. Van der Pol方程式に4つの異なる(自分で決めた)初期状態を与えて, 数値的に解け. ただし, odeを用いること.
2. 平面をメッシュ状に分割し, 各メッシュの点における速度, 加速度を計算せよ.

# Lorenz方程式の解析

右の連立方程式はLorenz方程式という.

この Lorenz方程式に関して以下の問いに答えよ. ただし, 必要なパラメーター, シミュレーション条件は各自で設定すること.

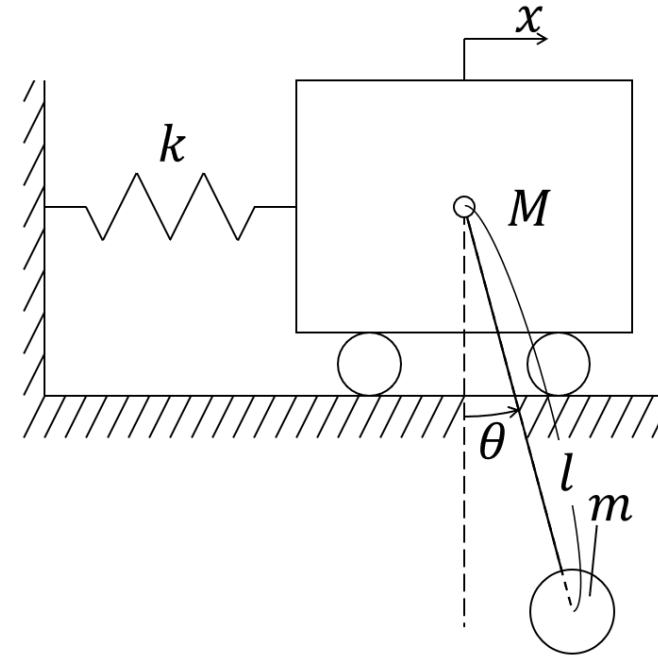
1. Lorenz方程式をodeを用いて解け.
2. 1の結果をxy, yz, zx平面にそれぞれプロットせよ.
3. 時間と共に1の結果がどう変化するかを3次元空間上に示した動画を作成せよ.

$$\begin{cases} \frac{dx}{dt} = -px + py \\ \frac{dy}{dt} = -xz + rx - y \\ \frac{dz}{dt} = xy - bz \end{cases}$$

# バネ・マス系に振子を取り付けたものの運動

図に示すようなバネ・マス系に振子を取り付けたものを考える. 必要なパラメーター, シミュレーション条件は各自で設定し, ode を用いること.

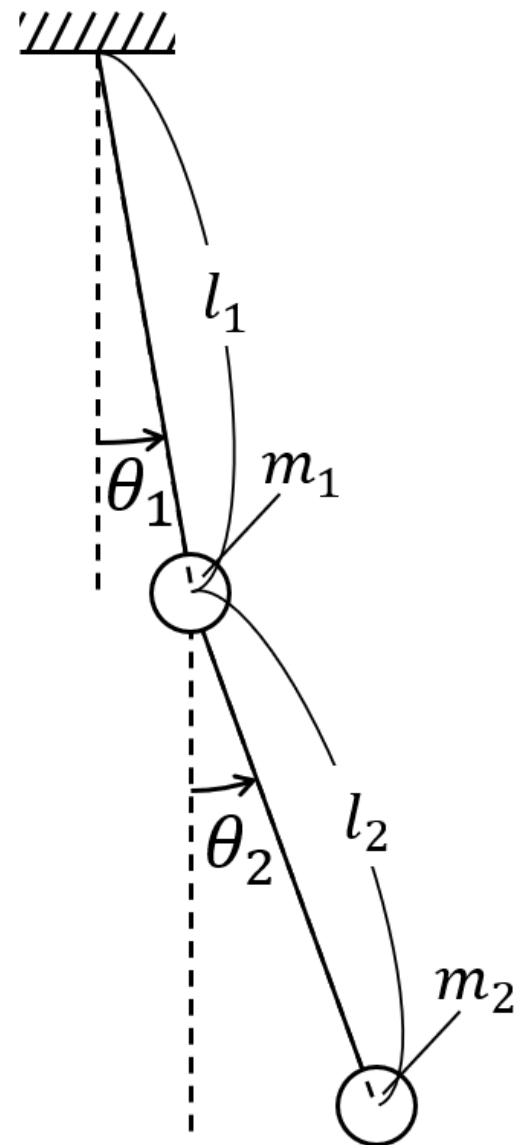
1. 状態方程式を求めよ.
2. 適当な初期状態を与え, 状態方程式を ode で解き, その結果をグラフで示せ.
3. 2の運動の様子を動画に示せ.



# 二重振り子の運動の様子

図に示すような二重振り子を考える. その際に糸はたるまないものとし, 各物理量や必要なシミュレーション条件は各自で与えてよい.

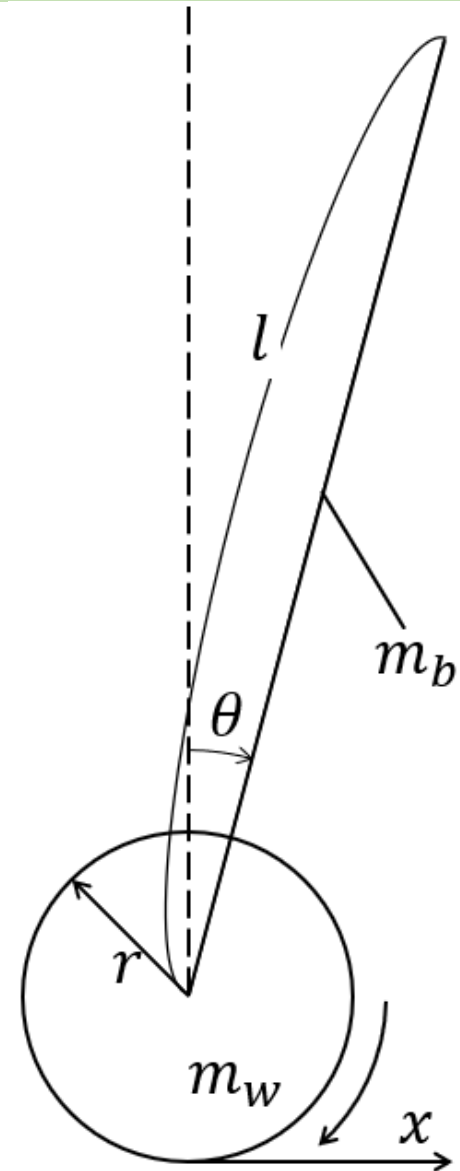
1. 二重振り子の運動方程式を求めよ.
2. 1つの初期値において, 求めた運動方程式をodeを用いて解き, それぞれの角度, 角速度の変化をグラフで示せ.
3. 2における初期値を微小にずらした初期値を100通り用意し, 時間履歴をodeで求め, 二重振り子の運動の様子を示す動画を作成せよ. その際に, 1つ1つが見分けのつくように工夫すること.



# 倒立振子の運動の様子

図に示すような倒立振子を考える. 必要なパラメータは各自で設定し, odeを用いること.

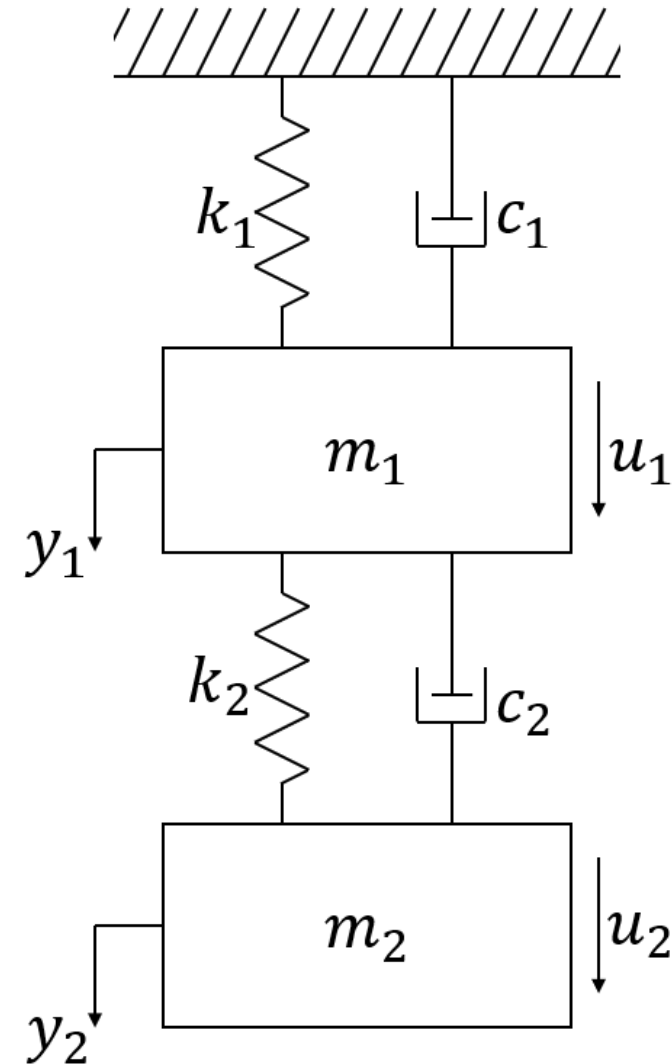
1. 倒立振子の状態方程式を求めよ.
2. 平衡点( $\theta = 0$ )まわりで線形化した状態方程式を求めよ. また, ある初期値からの制御問題を考え, 平衡点への制御入力を求め, その制御入力を加えた際の状態履歴を求めよ.
3. 2の際の倒立振子の様子を表す動画を作成せよ.
4. 線形化前の非線形な運動方程式に2で求めた制御入力を加えた際の状態履歴を求めよ.



# 2自由度系の振動の様子

図のような2自由度のバネ・マス・ダンパ系を考える. 必要なパラメーター, シミュレーション条件は各自で設定せよ.

1. このシステムの状態方程式を導出せよ.
2. ある初期値からの状態履歴を求め, グラフで示せ. ただし, odeを用いて状態方程式を解くこと.
3. 2の状態履歴を用いて, バネ・マス・ダンパ系の運動の様子を表す動画を作成せよ.





# スライド作成に当たって

- 課題には画像や動画の出力を求める問題があるが、ただこれをこなすだけでなく、プレゼンを行うのを念頭に置き、聴衆がわかりやすいようにする。
- 画像や動画を出力する際には、軸のキャプション等文字の大きさに注意する。
- 課題で挙げたもののほかにも各自で追加で検証や画像、動画の作成を行い、よりわかりやすく内容の深いプレゼンにできるとよい。
- 質疑応答の際に予想される質問等に関しては、あらかじめ説明用のスライドを後ろに用意しておくと、答えやすい。
- MATLAB課題やプレゼンでわからないことがあったり、アドバイスが欲しかったりする場合は、先輩に遠慮せずに聞いて下さい！