MATLAB入門セミナー
（基礎からデータ解析、可視化まで）
2016.11.29(火)

2016 年度 MATLAB TA
渡邉 郁弥　　木村 駿介

自己紹介

私たちMATLAB TA は

MATLAB と Simulink の使用を支援します

MATLAB Office Hour

MathWorks
Accelerating the pace of engineering and science

TAがMATLAB/Simulinkに関するインストールから
実践的な使い方までどんな質問にも対応.

場所：南3号館2階リフレッシュルーム

連絡先：sim_edu@citl.titech.ac.jp

Twitter
はじめました
@MATLAB_titech

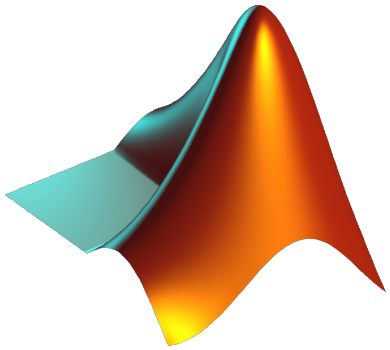
実施時間（3 Quarter）

月曜日	火曜日	水曜日	木曜日	金曜日
13:20 ～ 16:35	13:20 ～ 14:35	10:45 ～ 12:15	13:20 ～ 16:35	13:20 ～ 16:35

講習会・Office Hourの最新情報は下記リンク先へ

http://www.citl.titech.ac.jp/index.php/learning-environment/simulation-edu/

MATLABとは？



MATLAB

科学技術計算に特化した数値計算ソフトウェア

何ができるの？

例えば

- 行列演算
- 数値シミュレーション
- 信号処理（画像処理）
- 可視化（グラフ化）

など

世界中の大学・企業で
利用されている！

複雑な科学技術計算を誰でも簡単に扱うことができる！

講習会の流れ

本講習会は基本的に以下の3ステップの流れで行います.
①機能の説明
②例題を実演
③演習問題

質問は随時受け付けます.
わからなくなったらいつでも聞いてください！

Outline

基本的な演算と変数
- 四則演算
- 数学関数
- 変数について
- ベクトル・行列の定義
- ベクトル・行列の要素へのアクセス
- ベクトル・行列の演算と関数

スクリプトと関数
- スクリプトファイル
- 関数の定義
- 分岐と繰り返し

可視化
- 2次元プロット
- 3次元プロット

データの読み込み・解析
- データの読み込み
- 最小二乗法
- データの書き込み

Outline

基本的な演算と変数
- 四則演算
- 数学関数
- 変数について
- ベクトル・行列の定義
- ベクトル・行列の要素へのアクセス
- ベクトル・行列の演算と関数

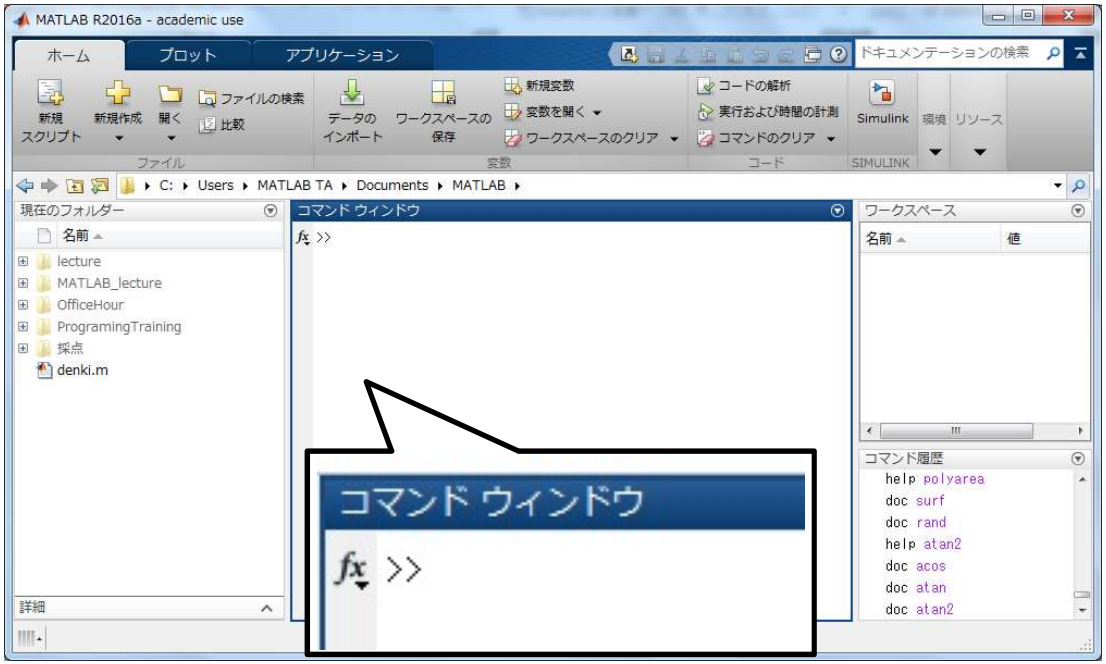
スクリプトと関数
- スクリプトファイル
- 関数の定義
- 分岐と繰り返し

可視化
- 2次元プロット
- 3次元プロット

データの読み込み・解析
- データの読み込み
- 最小二乗法
- データの書き込み

コマンドウィンドウについて

まずはMATLABを電卓として使ってみよう



コマンドウィンドウ上で
コマンドを入力することで操作

打ち間違えて実行，再度実行したい場合，
方向キー↑↓で入力したコマンドの履歴を使用可能

四則演算

基本的な加減乗除記号

足し算：+　　引き算：−
掛け算：＊　　割り算：/
累乗　：x^y

例題：以下の計算をしてみよう
(1) 3 + 5
(2) 4 – 9
(3) 2 * 3
(4) 1 / 3
(5) 2^10
(6) (3+2i)*i

※複素数を扱いたい場合，
MATLABではi, jの両方を虚数単位
として使用可能

数学関数

よく使う数学関数
三角関数　　:sin, cos, tan
逆三角関数　　:asin, acos, atan
指数・対数関数:exp, log, log10, log2

など，多数用意されている.
よく使う計算は大抵用意されているので，探せば出てくる.

例題:以下の計算をしてみよう
(1) sin(pi/2)
(2) atan(1)
(3) exp(1)
(4) exp(i*pi/2)

参考:オイラーの公式

$$e^{i\theta} = \cos\theta + i\sin\theta$$

変数について

数値を保存するために，変数を利用することができる.

例：

```
>> a = cos(pi/4);
>> b = sin(pi/4);
>> z = a + i*b;
>> theta = log(z)
```

- MATLABでは変数利用時にデータ型を指定する必要が無い（指定も可能）.
- 行の最後にセミコロンをつけないと，現在の変数の値を表示
- 変数に代入しない場合はans という変数に格納

ベクトル・行列の定義

以下の例を試してみよう.

例 :

```
>> x = [1; 2; 3]
>> A = [2 0 0; 1 2 3; 0 0 0]
>> A*x
```

$$Ax = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 2 & 3 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

ベクトル・行列は [ ] で囲んで定義する.
スペースまたはカンマで右隣の要素へ, セミコロンで改行.

特殊なベクトル・行列の定義

よく使うベクトル・行列は以下の関数・表現を使って簡単に得られる.

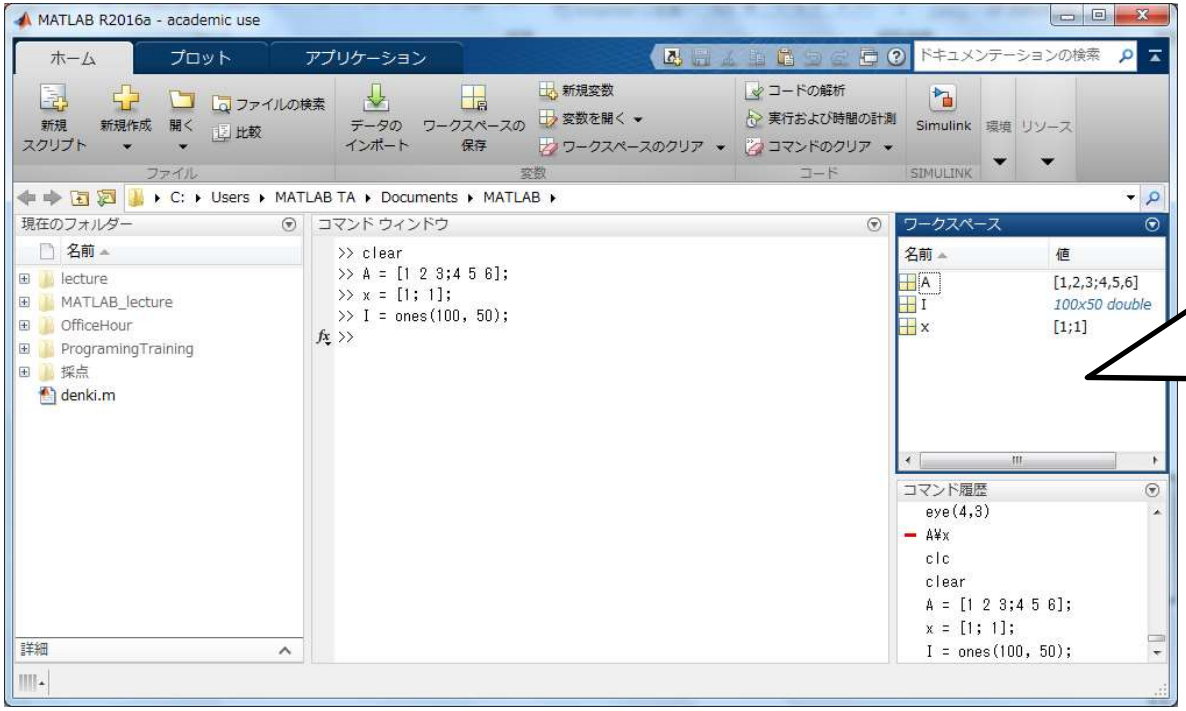
等間隔ベクトル	:[a:b:c]
Aからbまでn個のベクトル	:linspace(a, b, n)
n 次元の単位行列	:eye(n)
m 行n 列の零行列	:zeros(m, n)
全要素が1の行列	:ones(m, n)

MATLAB で所望の動作を見つける基本

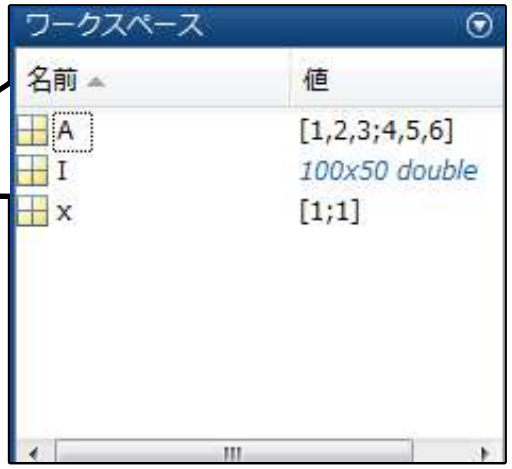
➡ Google 先生で検索

例:3行4列の乱数行列を作る方法を調べる
「matlab 乱数」で検索→MathWorks のページへ(実行例)

ワークスペースについて



現在定義されている変数情報は
ワークスペースで確認可能



要素数が多いと「サイズと型」の表示となる
ダブルクリックで変数の中身を確認できる
（コマンドウィンドウ上で確認するためには
コマンドウィンドウで変数名を入力）

ベクトル・行列の要素へのアクセス

行列の要素を参照したい場合は以下のようにすればよい.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \end{bmatrix}$$

2行3列目　A(2, 3) = 6

2から4行目の1列目　A(2:4,1) = $\begin{bmatrix} 4 \\ 7 \\ 10 \end{bmatrix}$

3行目全部　A(3, :) = [7 8 9]

コロンを使うとまとまった要素を参照する.
コロンのみを使うとすべての要素を参照する.

※注意
MATLABでは行列の要素は1から数え始める.
C言語の配列では0から始まりなので，混乱しないよう注意.

ベクトル・行列の演算と関数

よく利用するベクトル・行列の演算と関数は以下のとおり.

要素同士の演算	: A.*B, A./Bなど演算子の前に . (ドット)をつける
転置	: A.'
共役転置	: A'
逆行列	: inv(A)
連立方程式の解	: A¥b
固有値・固有ベクトル	: [V, D] = eig(A) (Vに固有ベクトル, Dに固有値)
p-ノルム	: norm(x, p)
行列のサイズ	: [m, n] = size(A)
サイズの最大値	: length(A)
要素の最大値	: max(A)
要素の総和	: sum(A)

演習問題

1. 300 より小さい9 の倍数を横に並べた列A を作成せよ

$$A = [9, 18, 27, \ldots, 279, 288, 297]$$

2. Aと同じサイズですべての要素を3 とした列B を作成せよ

$$B = [3, 3, 3, \ldots, 3, 3, 3]$$

1. A の各要素をBの各要素で除算した列C を作成せよ

$$C = [3, 6, 9, \ldots, 93, 96, 99]$$

2. C を変形（次元の変更）して3 行 11 列の行列D を作成せよ

$$D = \begin{bmatrix} 3 & 12 & \cdots & 84 & 93 \\ 6 & 15 & \cdots & 87 & 96 \\ 9 & 18 & \cdots & 90 & 99 \end{bmatrix}$$

3. D の奇数行目と奇数列目の要素のみを集めた行列E を作成せよ

$$E = \begin{bmatrix} 3 & 21 & \cdots & 75 & 93 \\ 9 & 27 & \cdots & 81 & 99 \end{bmatrix}$$

Outline

基本的な演算と変数
- 四則演算
- 数学関数
- 変数について
- ベクトル・行列の定義
- ベクトル・行列の要素へのアクセス
- ベクトル・行列の演算と関数

スクリプトと関数
- スクリプトファイル
- 関数の定義
- 分岐と繰り返し

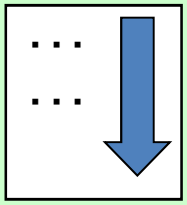
可視化
- 2次元プロット
- 3次元プロット

データの読み込み・解析
- データの読み込み
- 最小二乗法
- データの書き込み

スクリプトと関数

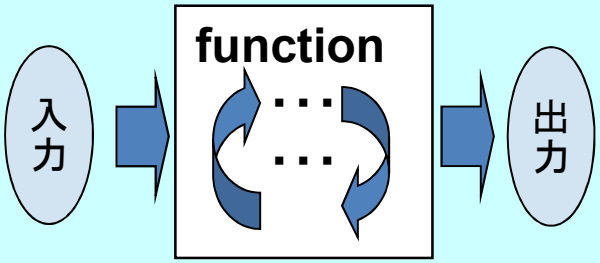
同じ処理を毎回毎回コマンド入力するのは面倒
⇒スクリプトや関数に処理をまとめて保存することができる！

スクリプトファイル（xxx.m）



・コマンド入力を記述，逐次的に処理
・変数の変更状況はワークスペース内で保存される
・メインプログラム＋インクルードファイルのイメージ

関数ファイル（function_name.m）



・関数は一般に入力変数、出力変数を持つ
・入出力変数は関数内で独立
・入出力変数を介してスクリプトとデータ交換

スクリプトエディタの起動

新規スクリプトをクリック
してエディタを起動

スクリプトファイルの保存・実行

スクリプト内にコマンドを記述したら，〇〇.mという形式で保存

エディタタブ内の実行をクリック

コマンドでスクリプトを実行するときは，
>> 〇〇(保存したファイル名，拡張子なし)
と入力して実行する．

実行のショートカットキーは
F5
中断は
Ctrl + c (command + c)

例：円周と面積の計算

```
r = 6;
circ = 2 .* pi .* r
area = pi .* r.^2
```

応用：r = 1:3 としてみよう

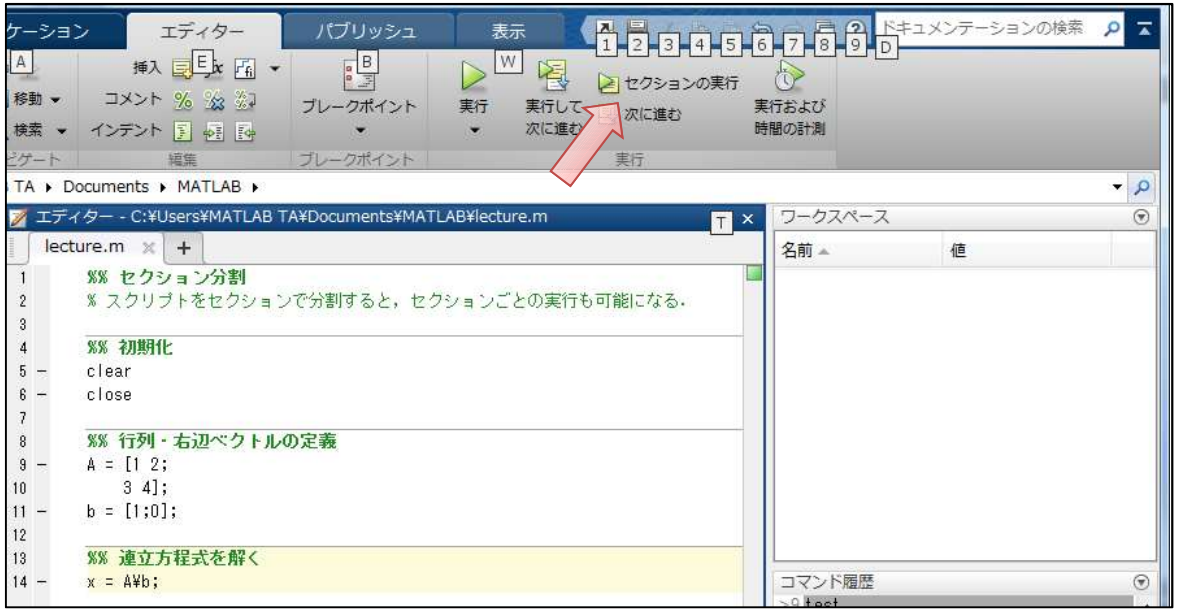
このファイルをmy_circle.m と保存
実行ボタンかコマンドウィンドウで
>> my_circle

コメントとセクション分割

%記号を使うことでコメントを入力できる
何をしたか忘れないように，できるだけコメントを残すことを推奨
一括コメントアウト　　ctrl + r (command + /)
一括コメントアウト解除　ctrl + t (command + /)

%%記号を使うことで
セクション分割も可能

セクション分割すると，
セクションごとに分けて
スクリプトを実行できる．
ctrl + enter
(command + enter)



関数ファイルの作り方

```
function [out1, out2, … ] = func(in1, in2, …)

処理の内容

end
```

関数もスクリプトファイルと同様に .m 形式で保存する.
関数ファイルは関数名と同じファイル名にすること！

例 : 極形式への変換

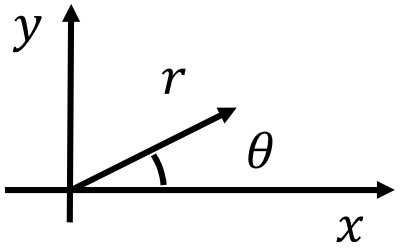
```
function [r, theta] = phasor(x, y)
        r = sqrt(x.^2+y.^2);
        theta = atand(y./x);
end
```

phasor.m で保存

$$r = \sqrt{x^2 + y^2}$$
$$\theta = \tan^{-1}\left(\frac{y}{x}\right)$$

関数の実行

```
>>[a,b] = phasor(2, 1)
```



制御構文(if文)

考え方はC言語と同じ.

MATLABではend までがひとつのブロックとなる.

よく使う論理演算子

論理和(OR)	|
論理積(AND)	&
否定(NOT)	~

よく使う比較演算子

等号	==
不等式	> や >= など
不等号	~=

```
if (条件式)
        処理内容
elseif (条件式)
        処理内容
else
        処理内容
end
```

制御構文(for文)

```
for (変数名) = (ベクトル)

    処理内容

end
```

forの隣で定義したベクトルの要素をひとつずつ網羅するように繰り返す.

例 : 1から5まで足す

```
n = 0;
for id = 1:5

    n = n + id;

end
```

例 : 0から10まで偶数だけ足す

```
n = 0;
for id = 0:2:10

    n = n + id;

end
```

制御構文(while文)

```
while (条件式)

    処理内容

end
```

条件式をみたしている間はブロック内を繰り返す.
無限ループに注意.

例:5 でない間繰り返す
(5になるまで繰り返す)

```
n = 0;
while n ~= 5

    n = n + 1;

end
```

例:5になるまで繰り返す
（ならない→無限ループ）

```
n = 0;
while n ~= 5

    n = n + 2;

end
```

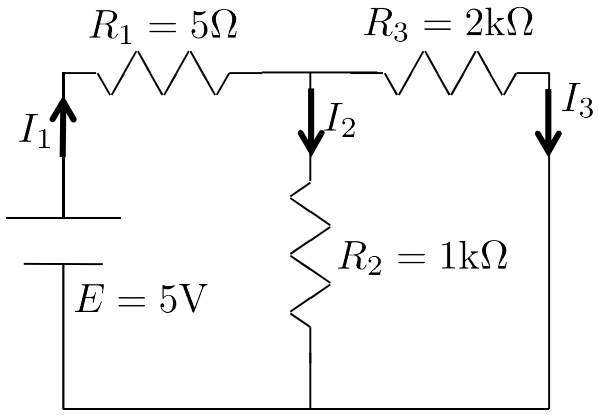
Ctrl + c (command + c)

演習問題

1. 引数 n に対して，次の値を計算する関数をつくりなさい．

$$f(n) = \sum_{k=1}^{n} g(k), \quad g(x) = \begin{cases} x & (x = 1, 3, \ldots) \\ x^2 & (x = 2, 4, \ldots) \end{cases}$$

2. 次の直流回路の各枝路に流れる電流を求めなさい．



キルヒホッフの法則から

$$\begin{aligned} 0 &= I_1 - I_2 - I_3 \\ E &= R_1 I_1 + R_2 I_2 \\ E &= R_1 I_1 + R_3 I_3 \end{aligned}$$

ヒント：I_3 を消去すると

$$E = (R_1 + R_3) I_1 - R_3 I_2$$

であるから，

$$\begin{bmatrix} R_1 & R_2 \\ R_1 + R_3 & -R_3 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} E \\ E \end{bmatrix}$$

が成り立つ．

その他小ネタ

- スクリプトの実行 F5 キー
- ドラッグで選択した部分のみを実行したい場合 F9 キー
- 選択した部分（またはカーソルがある行）をコメントアウト
  ctrl + r (command + /)
- 選択した部分のコメントアウトの解除
  ctrl + t (command + /)
- 実行する際には現在のフォルダ（パス）を実行するファイルと合わせる必要があります．（パス違いによって出てくるポップアップは「フォルダの変更」で大丈夫です）
- 自主学習は「MATLAB Academy」で検索

Outline

基本的な演算と変数
- 四則演算
- 数学関数
- 変数について
- ベクトル・行列の定義
- ベクトル・行列の要素へのアクセス
- ベクトル・行列の演算と関数

スクリプトと関数
- スクリプトファイル
- 関数の定義
- 分岐と繰り返し

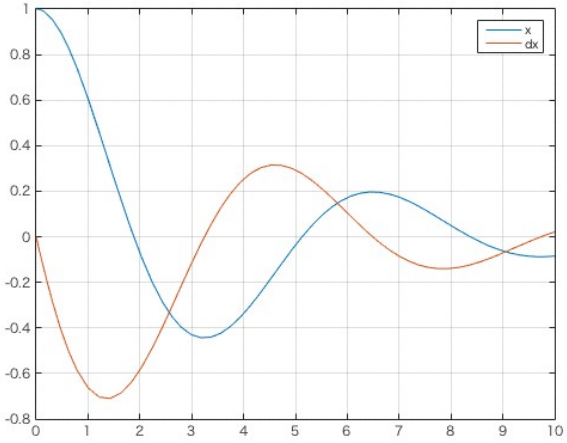
可視化
- 2次元プロット
- 3次元プロット

データの読み込み・解析
- データの読み込み
- 最小二乗法
- データの書き込み

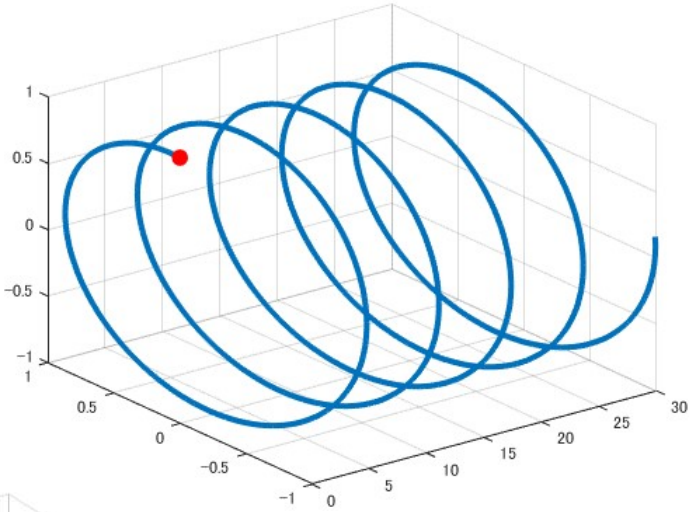
可視化

MATLABは強力なデータ可視化機能を持っている.

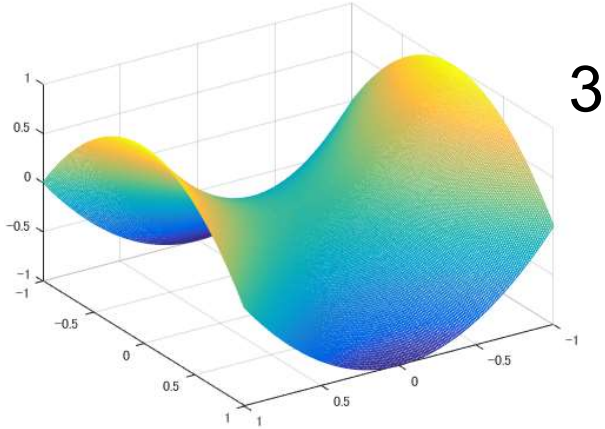
MATLABを使うと簡単に下のようなグラフが描ける！



2Dグラフ



3Dグラフ



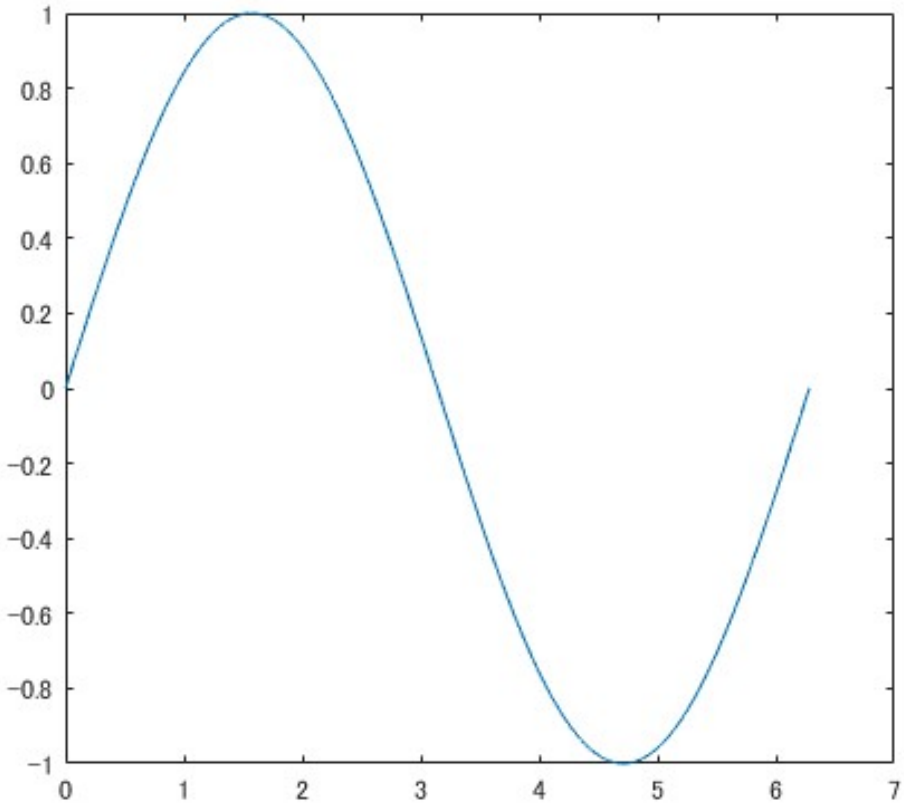
3D表面グラフ

2次元プロット

2次元プロット用の関数
plot(x1, y1, x2, y2, …, オプション)

例.

```
>> x = [0:0.01:2*pi];
>> y = sin(x);
>> plot(x, y);
```

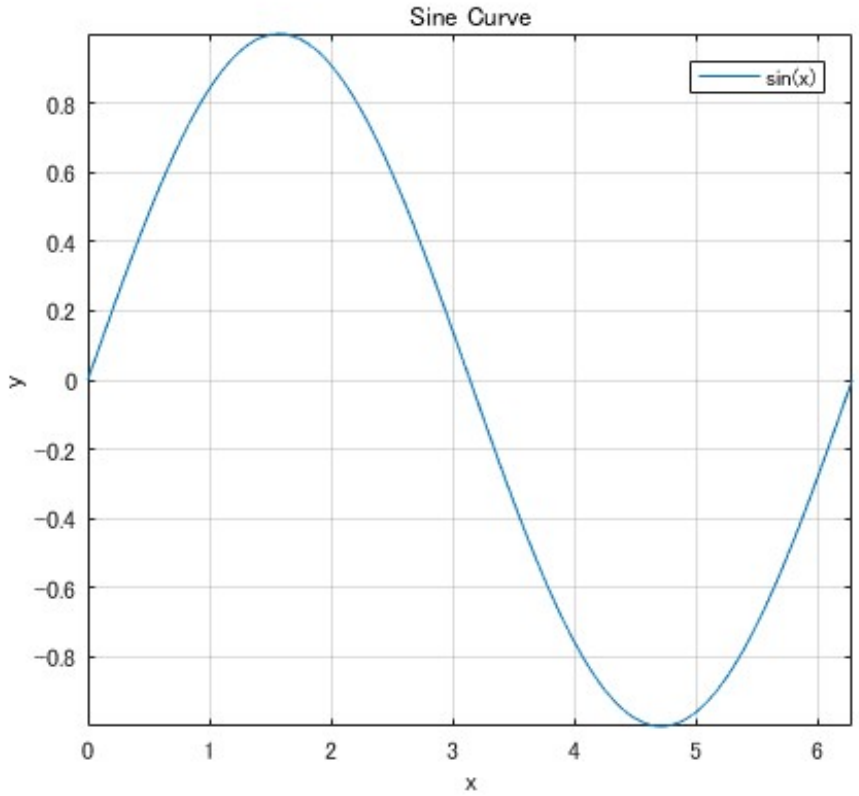


グラフの装飾

グラフを出したあとにコマンドを入力することでグラフを装飾できる.

例. (先ほどのコードに続けて)

```
>> grid on
>> title('Sine Curve')
>> xlabel('x')
>> ylabel('y')
>> legend('sin(x)')
>> axis tight
```

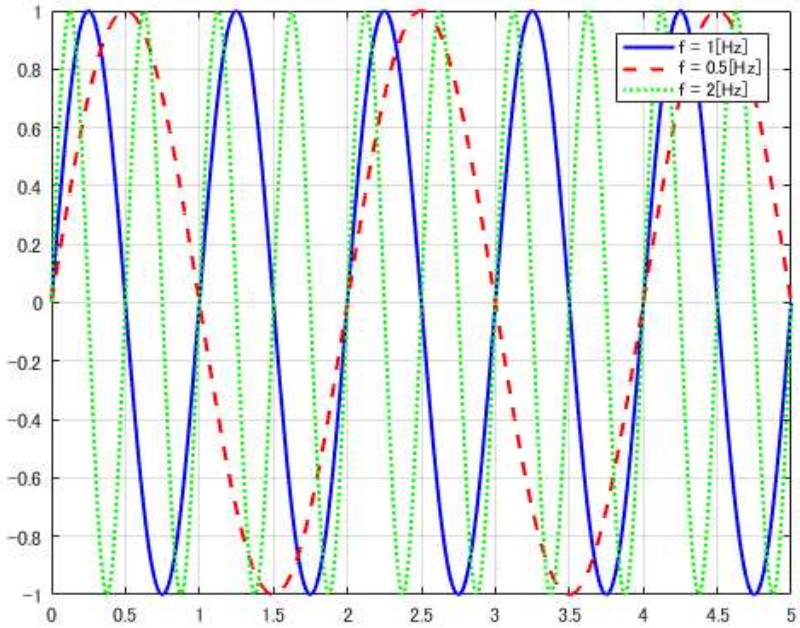


線種の変更

plot関数にオプションを指定することで線種を変えることもできる.

例.

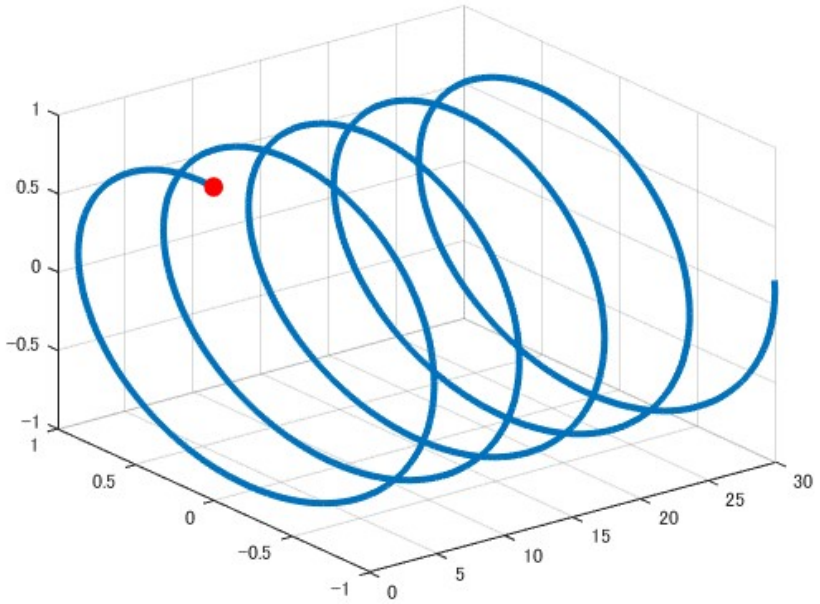
```
>> t = [0:0.01:5];
>> y1 = sin(2*pi*t);
>> y2 = sin(2*pi*0.5*t);
>> y3 = sin(2*pi*2*t);
>> plot(t, y1, 'b', 'LineWidth', 1.5);
>> grid on
>> hold on
>> plot(t, y2, 'r--', 'LineWidth', 1.5);
>> plot(t, y3, 'g:', 'LineWidth', 1.5);
>> legend('f = 1[Hz]', 'f = 0.5[Hz]', 'f = 2[Hz]')
```



3次元プロット

3次元プロットも2次元プロットと同じ要領でできる.
plot3(x, y, z, オプション)

```
>> x = [0:0.01:30];
>> y = sin(x);
>> z = cos(x);
>> plot3(x, y, z, 'LineWidth', 3);
>> grid on
>> hold on
>> plot3(x(1,1), y(1,1), z(1, 1), 'r.', 'MarkerSize', 30)
```



meshgridについて

表面プロットを行うにあたり，どのようにデータを与えるか
を理解する.

データは行列内に保存されているので，行番号や列番号を
データが存在する座標に変換する必要がある.
⇒[X, Y] = meshgrid(xgv, ygv)

X

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

Y

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

(-2, 0)

データ

3	4	0	2	1
7	6	5	3	3
1	1	1	1	1
1	3	6	4	2
0	0	5	6	4

例：meshgridによる3次元データの作成

$z(x, y) = x^2 - y^2$ という関数に対応する3次元データを作ってみよう.

```
>> [X, Y] = meshgrid([-2:2], [-2:2]);
>> Z = X.^2 – Y.^2;
```

X

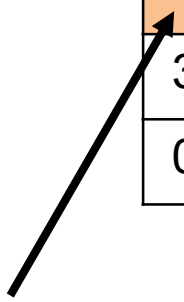
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2

Y

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

Z

0	-3	-4	-3	0
3	0	-1	0	3
4	1	0	1	4
3	0	-1	0	3
0	-3	4	-3	0

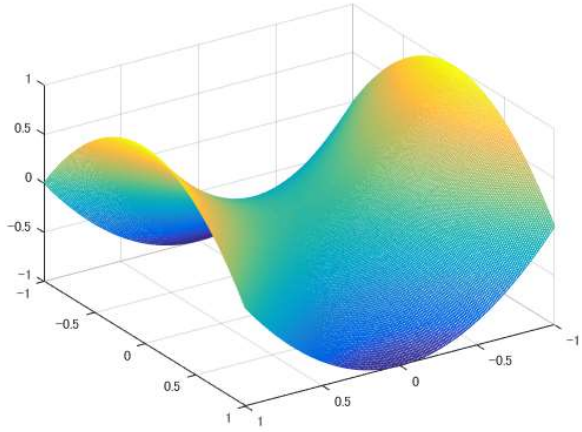


$$2^2 - 0^2 = 4$$

3次元表面プロット

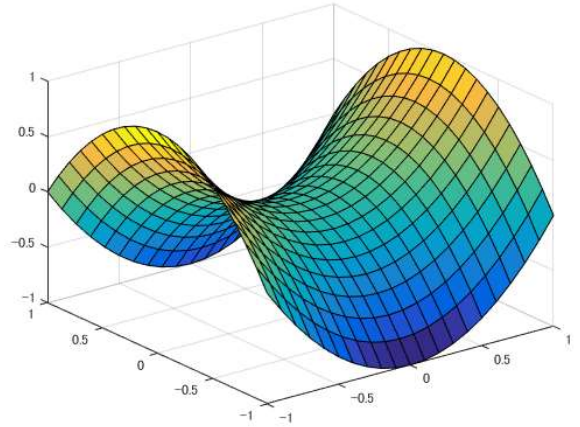
例.

```
>> [x,y] = meshgrid(-1:0.01:1,-1:0.01:1);
>> z =x.^2-y.^2;
>> mesh(x, y, z);
```



例.

```
>> [x,y] = meshgrid(-1:0.1:1,-1:0.1:1);
>> z =x.^2-y.^2;
>> surf(x, y, z);
```



演習問題

1. 単位円を表示しなさい.
   (ヒント:角度をパラメータに使うと簡単)
2. 次の関数をmeshで表示しなさい.
   (ヒント:meshgridを使おう)

$$z = p(x, y) = \frac{1}{2\pi} \exp\left(-\frac{x^2 + y^2}{2}\right)$$

Outline

基本的な演算と変数
- 四則演算
- 数学関数
- 変数について
- ベクトル・行列の定義
- ベクトル・行列の要素へのアクセス
- ベクトル・行列の演算と関数

スクリプトと関数
- スクリプトファイル
- 関数の定義
- 分岐と繰り返し

可視化
- 2次元プロット
- 3次元プロット

データの読み込み・解析
- データの読み込み
- 最小二乗法
- データの書き込み

データの読み込み

数値・テキストデータの読み込みは以下の関数が用意されている.

load	ワークスペース内変数を保存したMATLAB用データ(.mat)を読み込み
xlsread	Excelデータを読み込み
csvread	カンマ区切りファイル(.csv)を読み込み
dlmread	区切りテキストファイルを読み込み
fread	ファイルポインタとサイズを指定して読み込み

例：Excelデータの読み込み

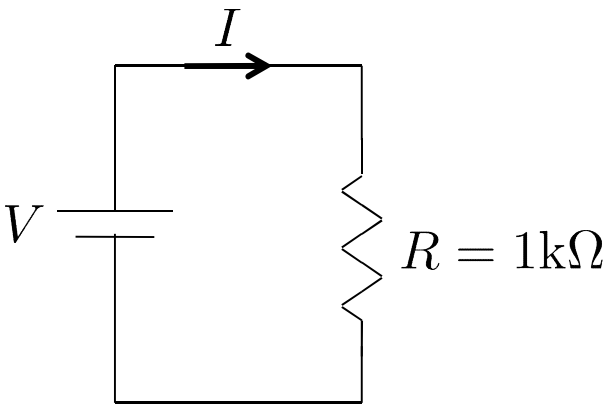
電圧・電流特性の測定実験データがExcelデータとして保存されているとき，このデータをグラフ化してみよう．

```
% Excelデータの読み込み
Data = xlsread('VI_data.xlsx');
V = Data(:,1);
I = Data(:,2);

% 表示
plot(V, I, 'o');
grid on
xlabel('Voltage [V]');
ylabel('Current [A]');
title('V-I Characteristic');
```

VI_data.xlsx

0	0
0.2	0.000224
0.4	0.000331
0.6	0.000535
0.8	0.000919
…	…



最小二乗法

測定データ$\{(x_i, y_i)\}_{i=1}^{n}$を元にして，ある関数$y = f(x)$に
フィッティングさせたい.

この問題を解決するために，一般的に以下の評価関数を考える.

$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(f(\theta, x_i) - y_i)^2$: 誤差の2乗和

ここで，θはパラメータである.

$J(\theta)$を最小化するようなθを求めれば，問題が解決！

線形モデル

ここでは，簡単のために生成する関数の形を線形モデルに制限して考えよう．

$$f(\theta, x) = \sum_{j=1}^{b} \theta_j \phi_j(x) = \theta^{\mathsf{T}} \phi(x), \quad \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_b \end{bmatrix}, \quad \phi(x) = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_b(x) \end{bmatrix}$$

このモデルを用いると，評価関数は以下のように表せる．

$$J(\theta) = \frac{1}{2} \sum_{i=0}^{n} \left(\theta^{\mathsf{T}} \phi(x_i) - y_i \right)^2 = \frac{1}{2} \| \Phi\theta - y \|^2, \quad \Phi = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_b(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \cdots & \phi_b(x_n) \end{bmatrix}$$

評価関数を最小化するようなθを求めれば，線形モデルのパラメータが得られる．

$\frac{\partial}{\partial\theta} J(\theta) = \Phi^{\mathsf{T}}\Phi\theta - \Phi^{\mathsf{T}} y = 0$ になればよいので，

$$\theta^* = \left(\Phi^{\mathsf{T}}\Phi \right)^{-1} \Phi^{\mathsf{T}} y$$

MATLABによる最小二乗法

$$y = \theta_1 x^2 + \theta_2 x + \theta_3 = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix} \begin{bmatrix} x^2 \\ x \\ 1 \end{bmatrix}$$ にフィッティング

```
%% 測定データの生成
% パラメータ設定
a_data = 2; b_data = 3; c_data = 1;

% y = ax^2 + bx + cにノイズをのせる
x_data = [0:0.1:2]';
y_data = a_data.*x_data.^2 +
b_data.*x_data + c_data +
0.5*randn(size(x_data));

%% 測定データの表示
plot(x_data, y_data, 'b*');
hold on
```

```
%% パラメータの計算
% 計画行列Phiの生成
Phi = [x_data.^2, x_data, ones(size(x_data))];

% パラメータを擬似逆行列で計算
Theta = Phi¥y_data;

% 得られたThetaを使って計算
y_est = Theta(1).*x_data.^2 + Theta(2).*x_data +
Theta(3);

% 関数値の表示
plot(x_data, y_est, 'r')
```

多項式フィッティング

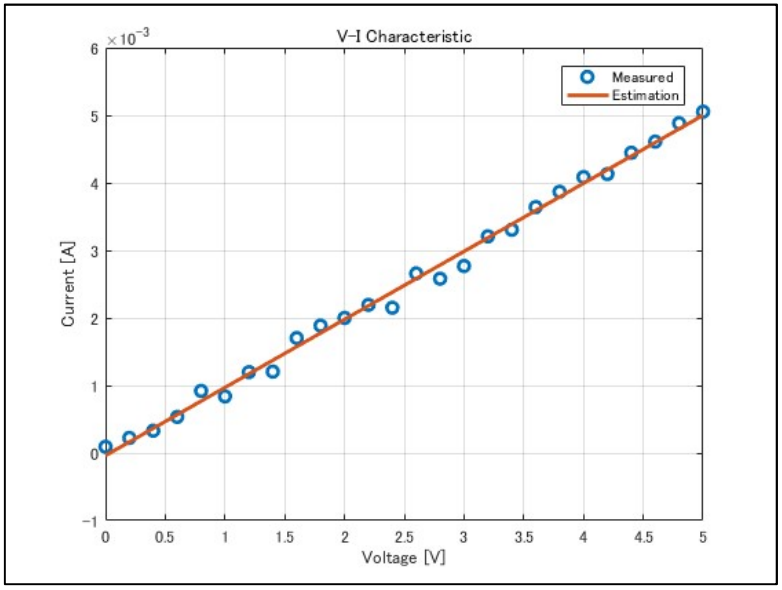
多項式フィッティングの場合は関数が用意されている.

$$y = \theta_1 x^n + \theta_2 x^{n-1} + \cdots + \theta_n x + \theta_{n+1}$$

theta = polyfit(x, y, n) : x, yを用いてn次の多項式フィッティング
Y = polyval(theta, x) ：thetaを用いてxの多項式を計算

```
% Excelデータの読み込み
Data = xlsread('VI_data.xlsx');
V = Data(:,1);
I = Data(:,2);

% I = aV + bの1次式にフィッティング
theta = polyfit(V, I, 1);
% パラメータthetaを用いて多項式を計算
I_est = polyval(theta, V);
```



データの書き込み

数値・テキストデータの書き込みは以下の関数が用意されている.

save	ワークスペースの変数をMATLAB用データ(.mat)としてそのまま保存
xlswrite	Excelデータとして保存
csvwrite	カンマ区切りファイル(.csv)として保存
dlmwrite	区切りテキストファイルとして保存
fwrite	ファイルポインタとサイズを指定して保存

例：多項式フィッティングの結果を書き込み

```
% Excelデータの読み込み
Data = xlsread('VI_data.xlsx');
V = Data(:,1);
I = Data(:,2);

% I = aV + bの1次式にフィッティング
theta = polyfit(V, I, 1);
% パラメータthetaを用いて多項式を計算
I_est = polyval(theta, V);

% データ書き込み
SaveData = [V I I_est]; % 保存するデータをひとつの行列にまとめる
xlswrite('VI_estimation.xlsx', SaveData); % データを保存
```

演習問題

斜方投射した物体(質点とみなす)の座標を計測した.

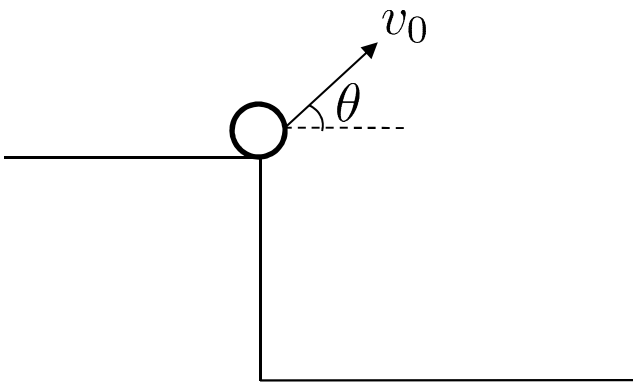
1. 計測データ(xy_data.xlsx)を読み込み，グラフ化せよ.
2. 放物線の式にフィッティングし，グラフ化せよ.
3. 2 で求めたパラメータから投射した角度と初速度を推定せよ.

ヒント：放物線の式

$$
\begin{aligned}
x &= v_0 \cos\theta \\
y &= v_0 \sin\theta - \frac{1}{2} g t^2 + y_0
\end{aligned}
$$

より

$$
y = x \tan\theta - \frac{g}{2 v_0^2 \cos^2\theta} x^2 + y_0
$$



$g = 9.81\mathrm{m/s^2}$ とする.

MATLAB Office Hour

TAがMATLAB/Simulinkに関する質問に対応します！
インストールから実践的な使い方までお気軽にどうぞ！

場所：南3号館2階リフレッシュルーム
* 火曜のみ ものつくりセンター中2階

連絡先：sim_edu@citl.titech.ac.jp

Twitter始めました！
@MATLAB_titech

実施時間（4 Quarter）

月曜日	*火曜日	水曜日	木曜日	金曜日
13:20 ～ 16:35	13:20 ～ 16:35	10:45 ～ 12:15	13:20 ～ 16:35	13:20 ～ 16:35

講習会・Office Hourの最新情報は下記リンク先へ

http://www.citl.titech.ac.jp/index.php/learning-environment/simulation-edu/