

2020年度 生田システム講習会

MATLAB講習会（基礎編）

明治大学 生田メディア支援事務室
2020年9月15日

目次

- MATLABとは (MATLABの特徴)
- インターフェイスの紹介
- 基本操作／データ入力 (行列・数列)
- 基本操作／演算・関数 (演算子・行列計算・転置行列・文字列)
- 連立方程式の計算
- グラフ描画 (plot関数)
- スクリプトファイル作成
- 困ったときは
- まとめ
- MATLAB Campus-Wide License
- 参考文献

MATLABとは

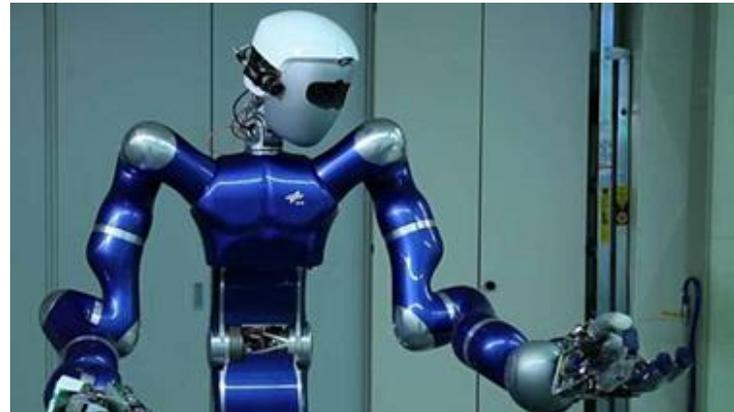
• 数値計算処理ソフト

理工学分野において、

データ解析・モデリング・シミュレーション等に使用される。



医用：造影剤注入ポンプの
流体動特性を解析



ロボット：自律ヒューマノイドロボットの
制御機能をシミュレーション

MATLABとは

・利用分野



航空産業・防衛産業



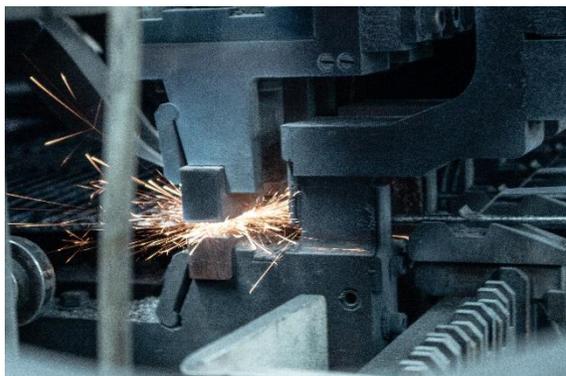
自動車



鉄道システム



医療機器



工作機械



鉱物・化学プラント



電子工学



生物科学

MATLABの特徴

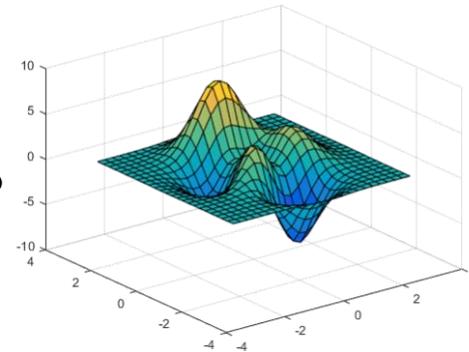
Matrix Laboratory

• 特徴

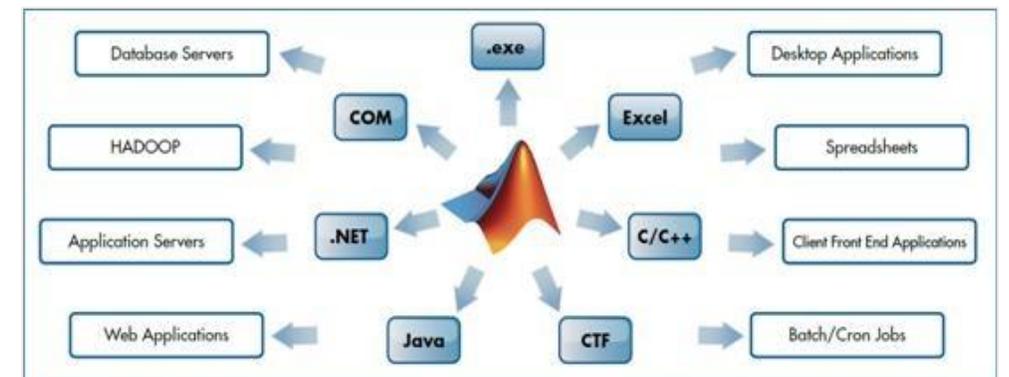
✓ データを**行列**として扱うことができる。

$$X = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

✓ **グラフ描画機能**を搭載している。



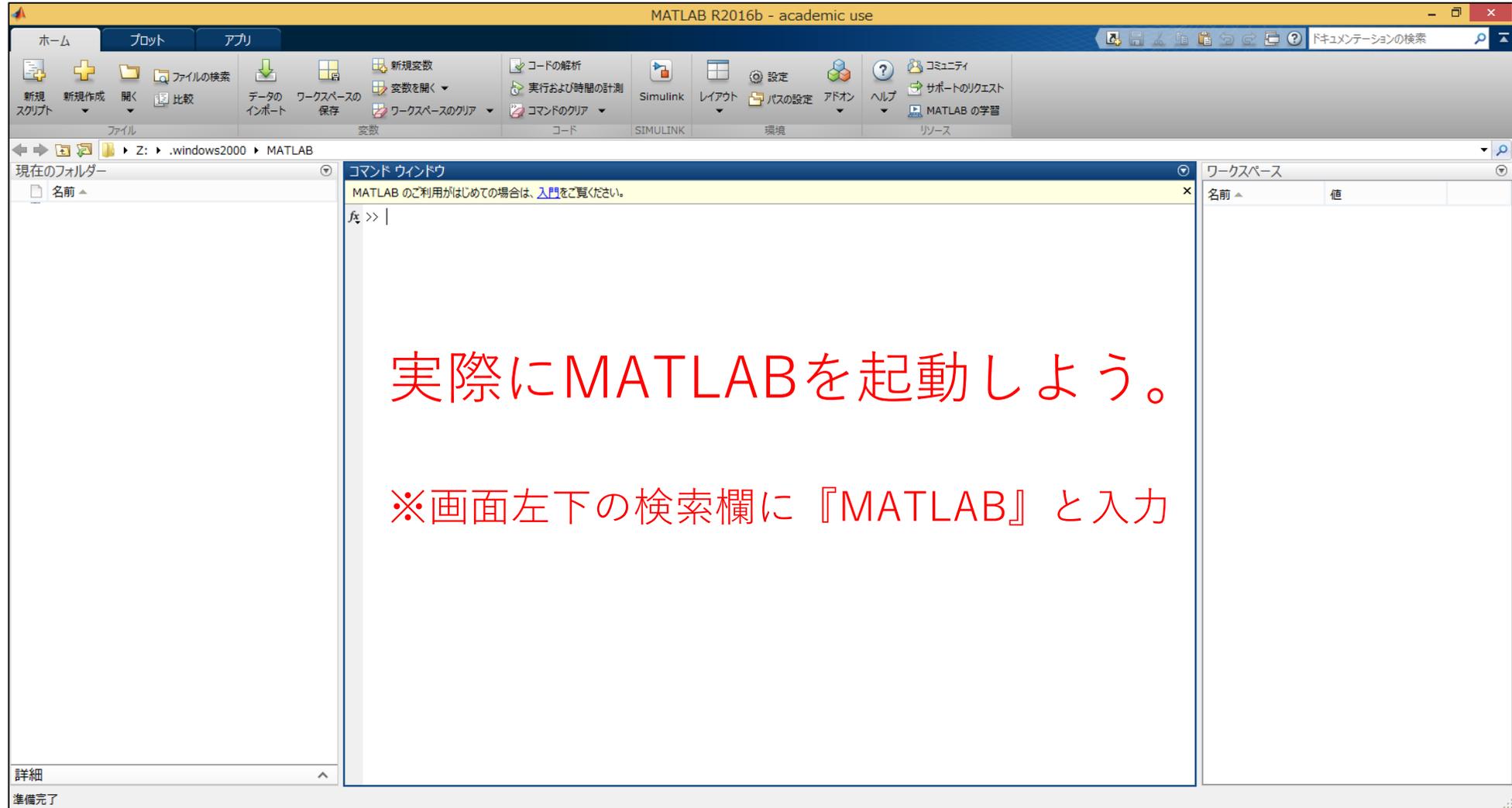
✓ **他アプリケーションとの連携**に
優れている。



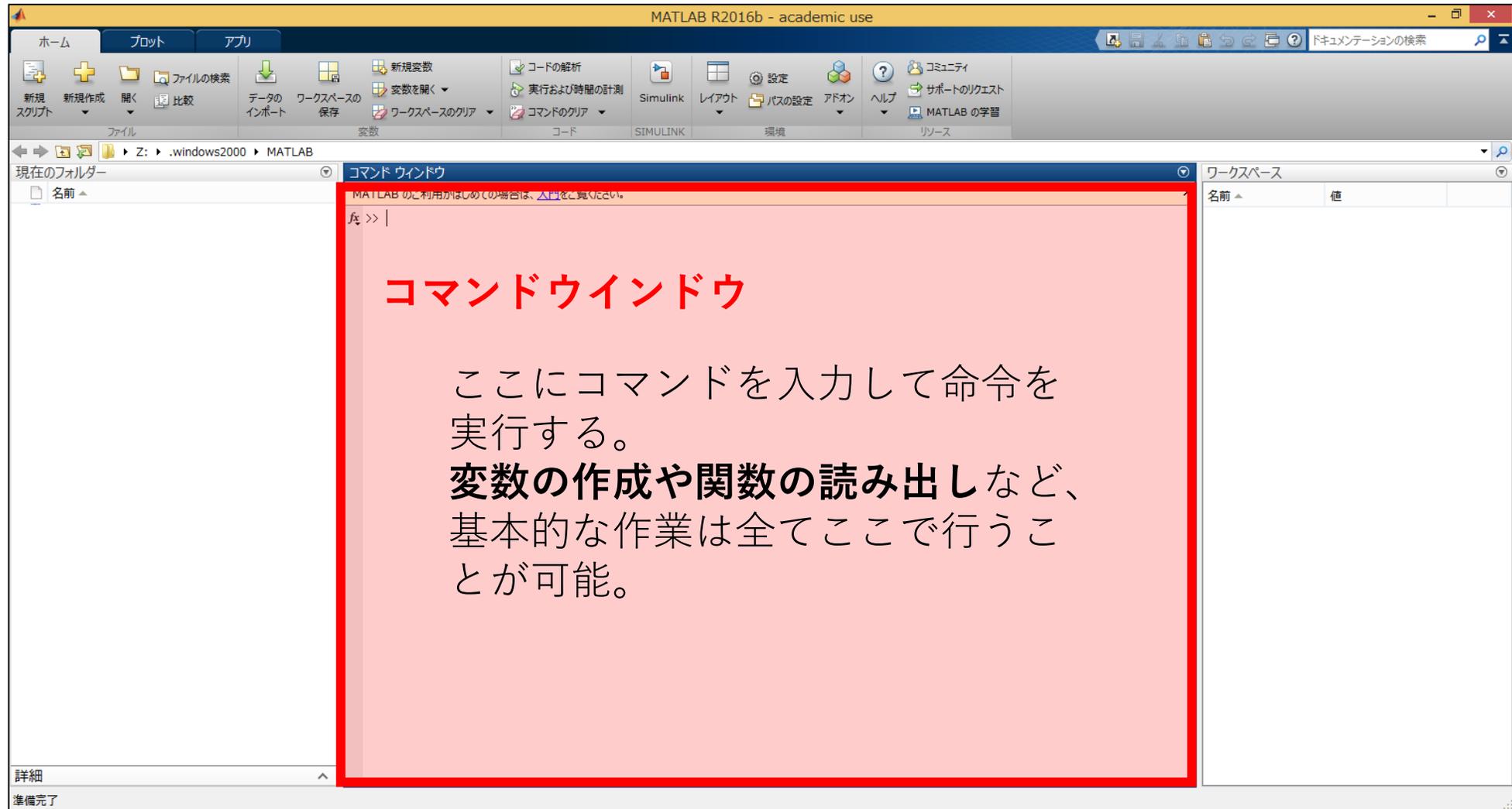
<https://jp.mathworks.com/help/matlab/ref/surf.html>

<https://jp.mathworks.com/discovery/anomaly-detection.html>

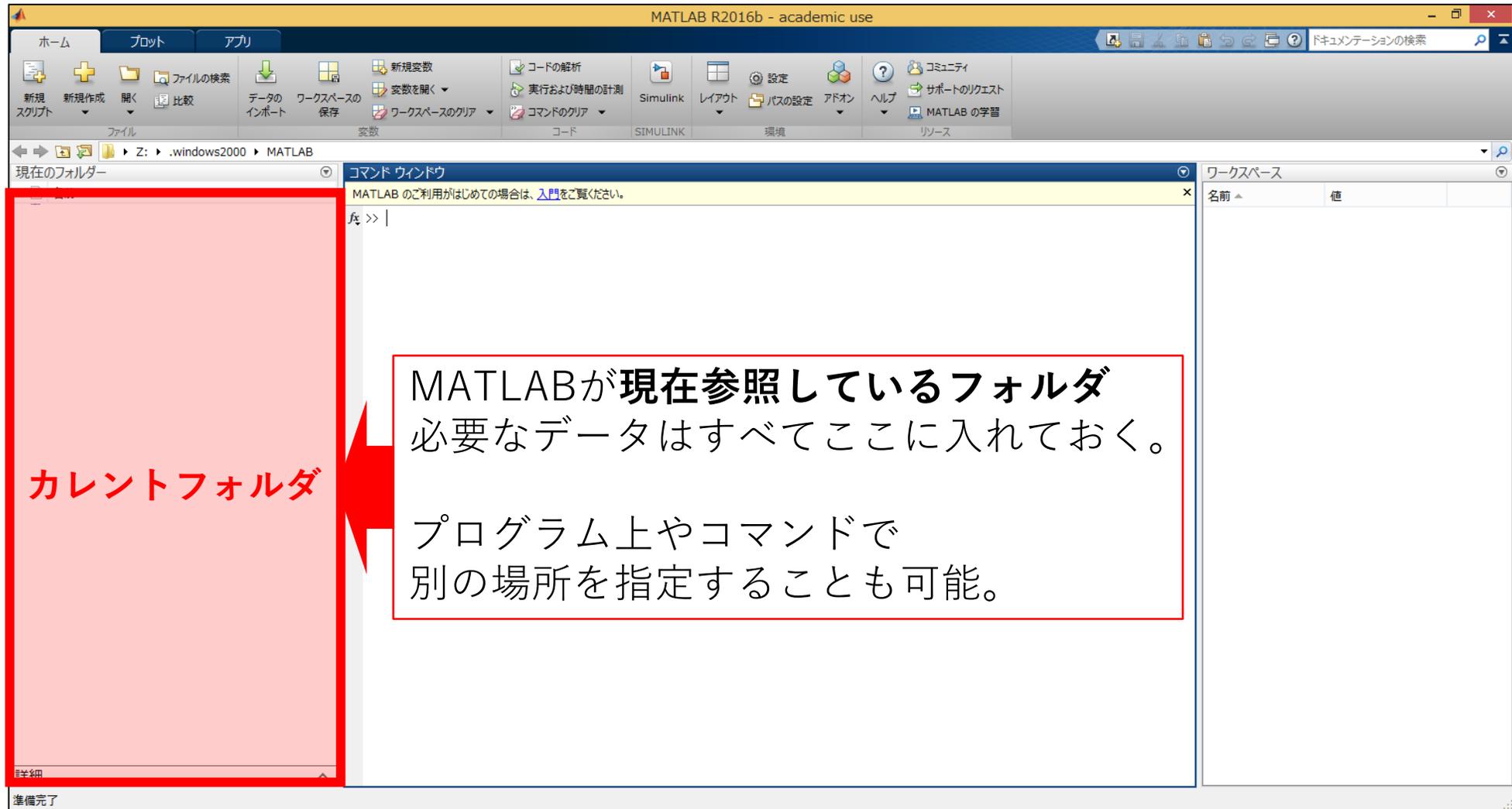
インターフェースの紹介



🚀 インターフェイスの紹介（コマンドウィンドウ）



🚩 インターフェイスの紹介（カレントフォルダ）



🚀 インターフェイスの紹介（ワークスペース）

コマンドウィンドウで入力した
変数や配列の一覧を表示する。

値が多い場合には
行列の数とデータ型を表示する。

ワークスペース

例題1

- 変数**a**の値を**5**と定義しよう。

コマンドウィンドウに『a=5』と入力し『Enterキー』を押す。

- 変数**b**の値を**3**と定義しよう。

コマンドウィンドウに『b=3』と入力し『Enterキー』を押す。

⇒ 上記操作によりプロンプト上で**a=5**, **b=3**と定義される。

⇒ データは**ワークスペース**（メモリ領域）に保存される。

※このデータはMATLAB終了時/**clear**コマンドが
実行されるまで保存される。

プログラム例

```
>> a = 5
```

```
a =
```

```
5
```

```
>> b = 3
```

```
b =
```

```
3
```

例題2

- 次の**行列**を入力しよう。

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

『`A=[1,2,3;4,5,6;7,8,9;]`』と入力して『Enterキー』を押す。

プログラム例

```
>> A = [1,2,3;4,5,6;7,8,9;]
```

```
A=
```

```
1 2 3
4 5 6
7 8 9
```

- ⇒ 行列の要素間は、**スペース**または**カンマ(,)**で区切る。
- ⇒ 行列の各行の終了は、**セミコロン(;)**で定義する。
- ⇒ 行列の要素全体は、**カギかっこ ([])**で囲み定義する。

例題3

- 次の**数列**を入力しよう。
 $B = \{0, 1, 2, 3, 4, 5\}$,
 $C = \{0, 2, 4, 6, 8, 10\}$,
 $D = \{10, 8, 6, 4, 2, 0\}$

『 $B = 0 : 1 : 5$ 』と入力して『Enterキー』を押す。

⇒ 数列を入力するには、

『**変数=最初の値 : 刻み値 : 最後の値**』と3つの数値を
コロン(:)で区切り記入する。

⇒ データは原則として、

1行n列の行列または**ベクトル**として格納される。

プログラム例

```
>> B = 0 : 1 : 5
```

```
B =  
  0 1 2 3 4 5
```

```
>> C = 0 : 2 : 10
```

```
C =  
  0 2 4 6 8 10
```

```
>> D = 10 : -2 : 0
```

```
D =  
 10 8 6 4 2 0
```

基本操作 / データ入力編 演習

基本操作 / データ入力編 演習①

演習1

- ・ 変数**a**の値を**8**、変数**b**の値を**6**と再定義し、
a × bの値が格納されるような変数**c**を定義しよう。

⇒ 再定義はコマンドウィンドウ上で、例題1と同様に定義してみよう。

⇒ 積を表す演算子は「*」、『**a*b**』のように表される。

基本操作 / データ入力編 演習①

演習1

- 変数**a**の値を**8**、変数**b**の値を**6**と再定義し、**a × b**の値が格納されるような変数**c**を定義しよう。

ワークスペース	
名前 ▲	値
a	8
b	6
c	48

プログラム例

```
>> a = 5
a =
    5
>> b = 3
b =
    3
```

再定義すると...



```
>> a = 8
a =
    8
>> b = 6
b =
    6
>> c = a * b
c =
    48
```

- ⇒ **a**、**b**それぞれ変わっていることが確認できる。
- ⇒ **c**が、再定義した**a**と**b**の積になっていることが確認できる。

基本操作 / データ入力編 演習②

演習2

- ・ 次の数列を入力しよう。

$$\mathbf{F} = \{0, \pi, 2\pi, 3\pi, 4\pi, 5\pi, 6\pi\}$$

⇒ 例題3同様に、数値と刻み値を**コロン(:)**で区切り、
入力しよう。

⇒ π は3.14...という無理数だが、
MATLABでは**pi**と入力すると...

⇒ 有理数として定義されている！

```
>> pi  
  
ans =  
  
3.1416
```

演習2

- 次の数列を入力しよう。

$$F = \{0, \pi, 2\pi, 3\pi, 4\pi, 5\pi, 6\pi\}$$

*を抜いて、**6pi**とすると...

プログラム例

```
>> F = 0:pi:6*pi
```

```
F =
```

```
0    3.1416    6.2832    9.4248    12.5664    15.7080    18.8496
```

```
>> F = 0:pi:6pi
```

```
F = 0:pi:6pi
```

↑

エラー：式が無効です。乗算演算子の不足、区切り文字の不足や不一致、またはその他の構文エラーをチェックしてください。行列を作成するには、小かっこの代わりに大かっこを使用してください。

もしかして:

```
>> F = 0:pi:6*pi
```

基本操作／演算・関数編

演算子	説明
+	加算
-	減算
*	乗算
/	割り算
^	べき乗
' ※	共役転置
.'	非共役転置

関数	説明
sin()	正弦
cos()	余弦
tan()	正接
atan()	逆正接
log()	自然対数
log10()	常用対数

関数	説明
exp()	指数関数
sqrt()	平方根
inv()	逆行列
abs()	絶対値
real()	複素数の実部
imag()	複素数の虚部

定数	説明
pi	円周率(=3.14...)
i,j	虚数単位

※ 『』 はシングルクォーテーション

⇒ MATLAB上で演算を行うには、上記のような**演算子**や**関数**が用いられる。

基本操作 / 演算・関数編 ① 演算子

例題4

・ 次の数値を求めてみよう。

$$\sin\left(\frac{\pi}{2}\right)$$

例題5

・ 次の数値を求めてみよう。

$$\sqrt{2}$$

プログラム例

```
>> sin(pi/2)
```

```
ans =
```

```
1
```

```
>> sqrt(2)
```

```
ans =
```

```
1.4142
```

プログラム例

```
>> A = [1,2,-3;2,8,-7;];  
>> B = [-3,5,6;-1,9,8;];  
  
>> A+B  
  
ans =  
  
    -2     7     3  
     1    17     1
```

例題6

・以下の行列を計算してみよう。

①行列 **A**, **B** を入力

$$A = \begin{bmatrix} 1 & 2 & -3 \\ 2 & 8 & -7 \end{bmatrix}, \quad B = \begin{bmatrix} -3 & 5 & 6 \\ -1 & 9 & 8 \end{bmatrix}$$

※出力を伴うプログラムの行の終わりに**セミコロン (;)**を挿入することで、出力を非表示にする。

②行列の足し算 **A + B** をMATLAB上で計算

『A+B』 と入力して『Enterキー』を押す。

プログラム例

例題6の続き

・以下の行列を計算してみよう。

③行列 A の転置行列を計算

『A.』 と入力して『Enterキー』を押す。

④行列の掛け算 $A \times B^T$ をMATLAB上で計算

『A*B.』 と入力して『Enterキー』を押す。

⇒単なる転置の場合は(.')、共役転置の場合は(')を用いる。

※) スカラーと行列の積の場合は(*)を用いる。

```
>> A.'
```

```
ans =
```

```
1 2  
2 8  
-3 -7
```

```
>> A * B.'
```

```
ans =
```

```
-11 -7  
-8 14
```

例題7

- ・ 次の**文字列と値**を入力しよう。

S = "Figure."

N = 1

プログラム例

```
>> S = "Figure."
```

```
S =
```

```
    "Figure."
```

```
>> N = 1
```

```
N =
```

```
    1
```

⇒ 文字列は、**ダブルクォーテーションマーク** (“) で囲んで表す。

例題7の続き

・文字列と数字を連結させて、「Figure.1」と表記させよう。

⇒MATLABには、文字列を扱う関数も用意されている。

⇒文字列の連結には**strcat**関数を用いる。

⇒ただし、**strcat(文字列,文字列)**とする必要があるが、Nは数値であるため、エラーになる。

```
>> strcat(S,N)
```

```
エラー: string/strcat (line 37)
```

```
入力は、文字ベクトル、文字ベクトルのセル配列、または文字列配列でなければなりません。
```

⇒**num2str(N)**とすると、数値を文字列として扱うことができる。

プログラム例

```
>> S = "Figure."
```

```
S =
```

```
    "Figure."
```

```
>> N = 1
```

```
N =
```

```
    1
```

例題7の続き

- ・ 文字列と数字を連結させて、「Figure.1」と表記させよう。

⇒その他にも関数が多くあり、ファイルからデータを入出力する際のファイル名を扱う場合に便利。

- ・ **strcat**: 文字列の連結
- ・ **strsplit**: 指定した記号で文字列を分割
- ・ **strcmp**: 文字列同士の比較

プログラム例

```
>> strcat(S,num2str(N))  
  
ans =  
  
    "Figure.1"
```

基本操作 / 演算 ・ 関数編 演習

基本操作 / 演算・関数編 演習

演習3

- ・ 次の数値を求めてみよう。

$$\log 10$$

演習4

- ・ 次の数値を求めてみよう。

$$e^2$$

基本操作 / 演算・関数編 演習

プログラム例

演習3

- ・ 次の数値を求めてみよう。

$\log 10$

```
>> log(10)
```

```
ans =
```

```
2.3026
```

```
>> exp(2)
```

```
ans =
```

```
7.3891
```

演習4

- ・ 次の数値を求めてみよう。

e^2

連立方程式の計算

連立方程式の計算は、行列の考え方を利用することで解くことができる。

例

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 2 \\ x_2 + x_3 = 4 \\ x_1 + 3x_2 + 5x_3 = 1 \end{cases}$$

上記のような連立方程式は以下のように行列で記述することができる。

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix}$$

連立方程式の計算に応用すると・・・

$$C = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix}, d = \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

とおくと、求めたい x_1, x_2, x_3 は以下のように求めることができる。

$$Cx = d \leftrightarrow x = C^{-1}d$$

連立方程式の計算

例題8

行列を利用して、次の連立方程式の x_1, x_2, x_3, x_4 を求めてみよう。

$$\begin{cases} 3x_1 - 2x_2 + 5x_3 + 4x_4 = 5 \\ x_2 + 2x_3 + 4x_4 = 11 \\ 2x_1 + 3x_2 - x_3 - 3x_4 = -2 \\ 5x_1 - 3x_2 + 2x_3 + 2x_4 = 9 \end{cases}$$

MATLABの計算手順

⇒ $Ex = d$ のように E, d を定義。

⇒ 逆行列を求める関数「**inv()**」を利用して

$x = E^{-1}d$ を計算する。

プログラム例

```
>> E = [3,-2,5,4; 0,1,2,4; 2,3,-1,-3; 5,-3,2,2;]
```

```
>> d = [5;11;-2;9;]
```

```
>> x = inv(E)*d
```

```
x =
```

```
2.0000
```

```
1.0000
```

```
-3.0000
```

```
4.0000
```

連立方程式の計算 演習

連立方程式の計算 演習①

演習5

行列を利用して、次の連立方程式の x_1, x_2, x_3, x_4, x_5 を求めてみよう。

$$\begin{cases} 4x_1 + x_2 & & & & = 5 \\ x_1 & & + 2x_3 & & = -3 \\ & 2x_2 & & - 4x_4 & = -2 \\ & & 4x_3 & & + x_5 = -4 \\ 2x_1 + 2x_2 & & & + x_4 - 3x_5 & = -7 \end{cases}$$

🚩 連立方程式の計算 演習①

プログラム例

演習5

行列を利用して、次の連立方程式の x_1, x_2, x_3, x_4, x_5 を求めてみよう。

$$\begin{cases} 4x_1 + x_2 & = 5 \\ x_1 + 2x_3 & = -3 \\ 2x_2 - 4x_4 & = -2 \\ 4x_3 + x_5 & = -4 \\ 2x_1 + 2x_2 + x_4 - 3x_5 & = -7 \end{cases}$$

```
>> G=[4,1,0,0,0;1,0,2,0,0;0,2,0,-4,0;0,0,4,0,1;2,2,0,1,-3];
>> c = [5;-3;-2;-4;-7];
>> x = inv(G)*c
```

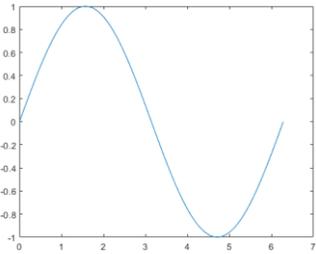
x =

```
1.0000
1.0000
-2.0000
1.0000
4.0000
```

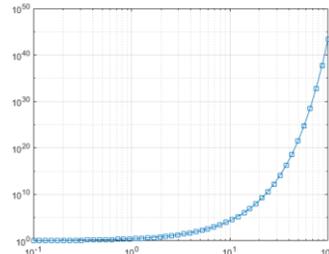
グラフ描写

グラフィックス関数によって、データを様々な形式で表現することができる。

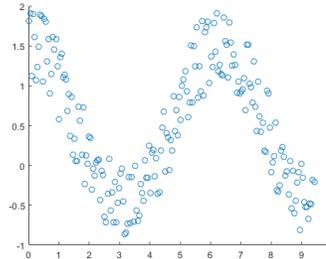
2次元グラフィックス



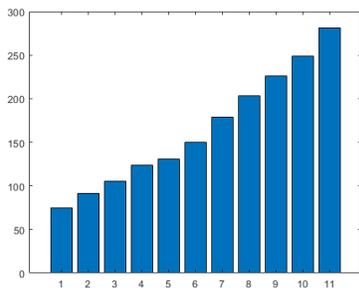
plot関数(x-yグラフ)



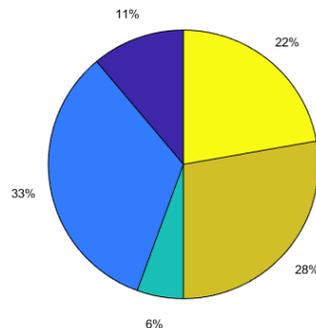
loglog関数(対数グラフ)



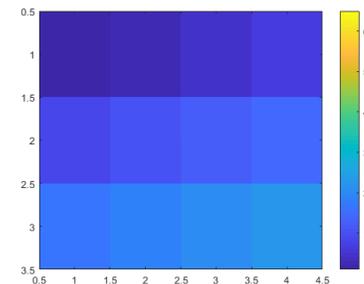
scatter関数(散布図)



bar関数(棒グラフ)

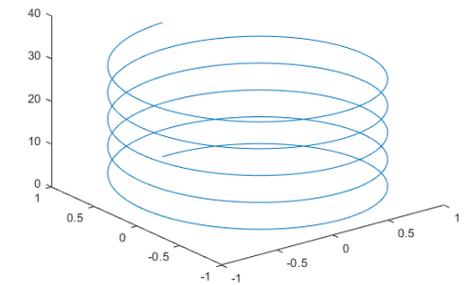


pie関数(円グラフ)

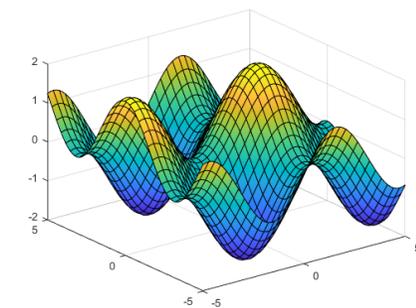


image関数(画像)

3次元グラフィックス



plot関数(x-y-zグラフ)



fsurf関数(3次元サーフェイス)

グラフ描写 / plot関数

今回は、**plot関数**を利用してグラフを描画する。

plot関数の定義 「plot(x, y, '色 マーカー 線種）」

- xが横軸、yが縦軸。
- x と y は、変数に限らず、ベクトルでも自動的にプロットされる。
(ただし、x と y のベクトルの大きさは常に一致していなければならない)
- グラフの色と線種に関するコマンドは右の表参照。

コマンド	色
y	黄
m	赤紫
c	水
r	赤
g	緑
b	青
w	白
k	黒

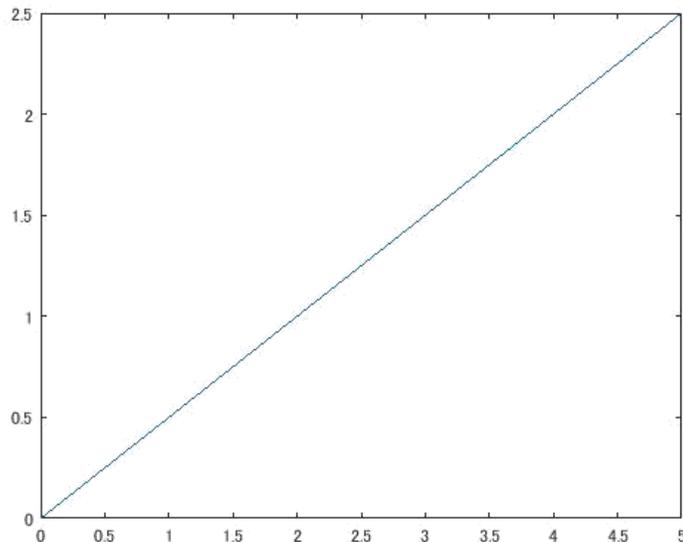
コマンド	線種
-	実線
:	点線
-.	鎖線
--	破線

コマンド	マーカー
.	テン
o	円
x	x印
+	+印
*	*印
s	四角形
d	ダイヤ型
v	三角形 (下)
^	三角形 (上)
<	三角形 (左)
>	三角形 (右)
p	五角形
h	六角形

グラフ描写 / plot関数 例題 (1)

例題9

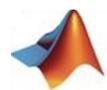
x が $0 \leq x \leq 5$, 刻み幅は 0.5 ($x = 0:0.5:5$) のとき、 $y = \frac{1}{2}x$ のグラフを描いてみよう。



プログラム例

```
>> x = 0:0.5:5;  
>> y = 1 / 2 * x;  
  
>> plot(x,y)
```

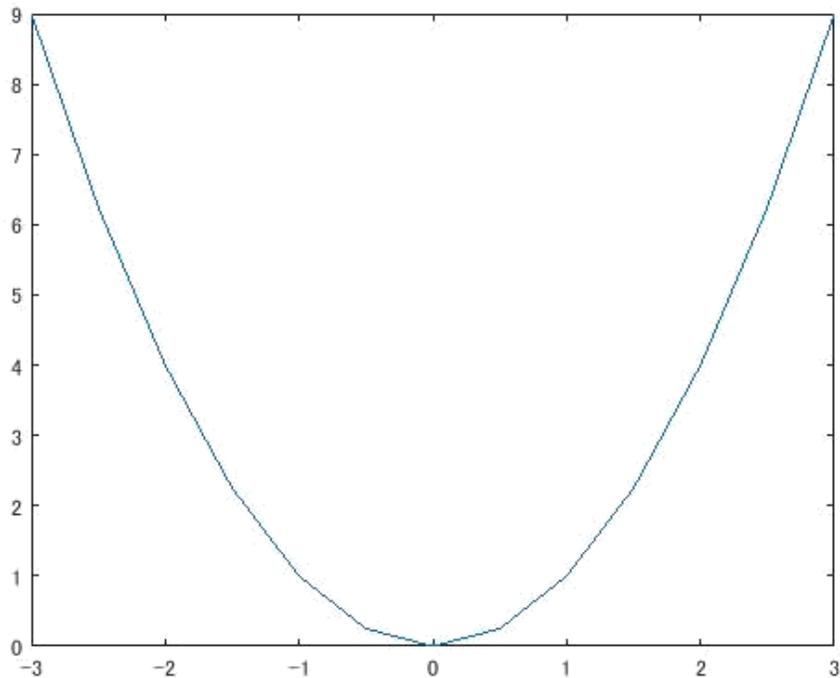
⇒ 関数**plot**により、データの最大値、最小値に応じて座標を自動的に生成し、 x - y グラフを図示できる。



グラフ描写 / plot関数 例題 (2)

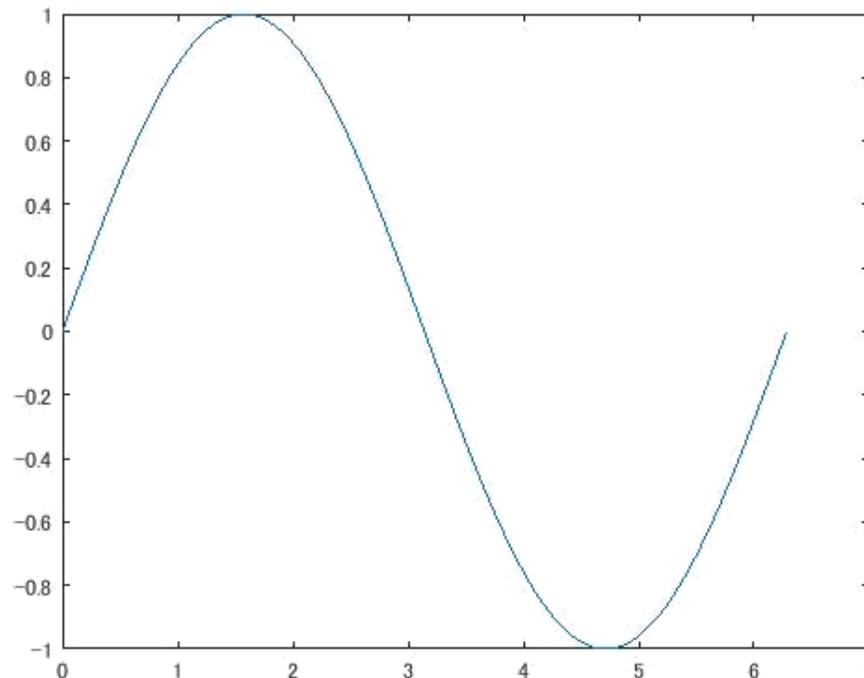
例題10

x が $-3 \leq x \leq 3$, 刻み幅が 0.5 ($x = -3:0.5:3$) のとき、 $y = x^2$ のグラフを描いてみよう。



例題11

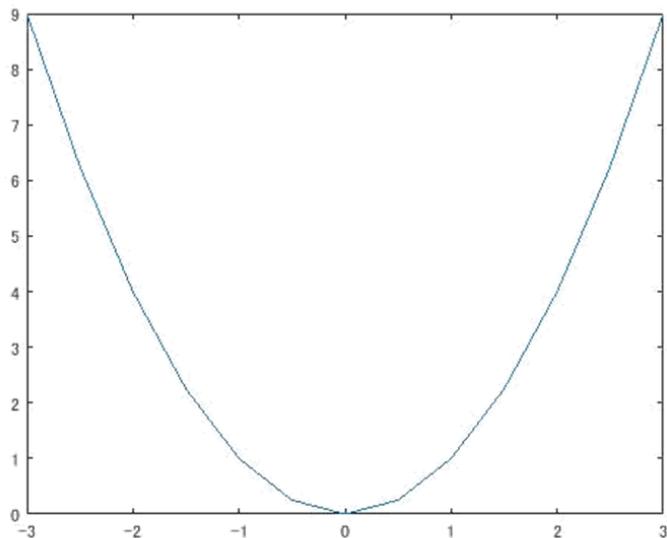
t が $0 \leq t \leq 2\pi$, 刻み幅は 0.01 ($t = 0:0.01:2\pi$) のとき、 $y = \sin t$ のグラフを描いてみよう。



グラフ描写 / plot関数 例題 (2)

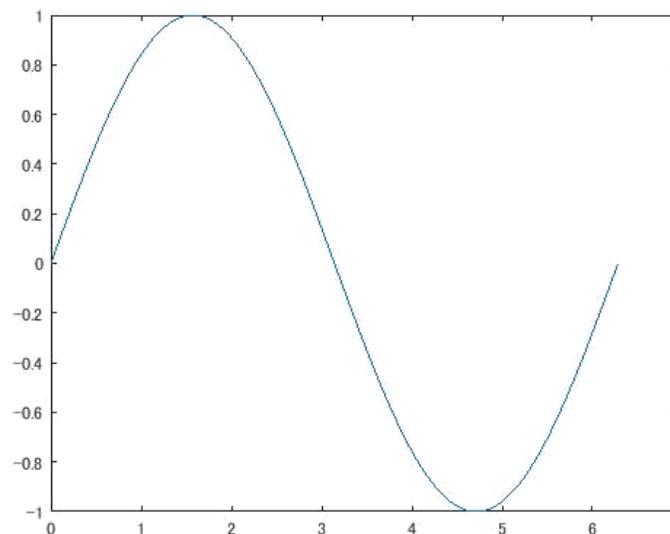
プログラム例

```
>> x = -3:0.5:3;  
>> y = x.*x  
  
>> plot(x,y)
```



プログラム例

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



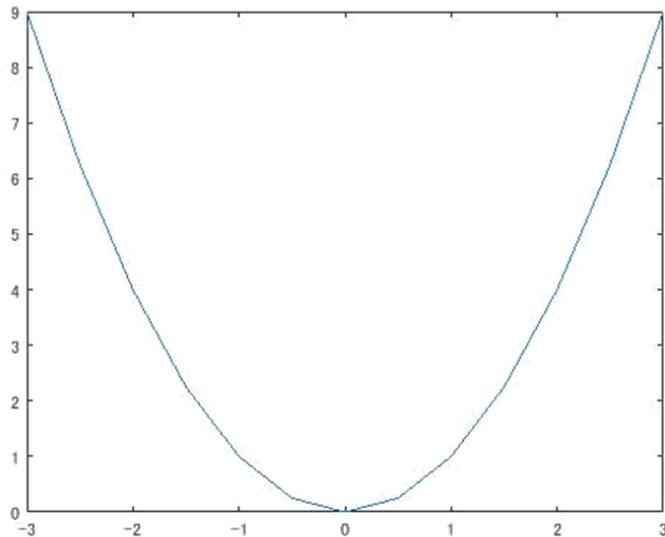
🦉 グラフ描写 / plot関数 例題 (2)

プログラム例

```
>> x = -3:0.5:3;  
>> y = x.*x
```

```
>> plot(x,y)
```

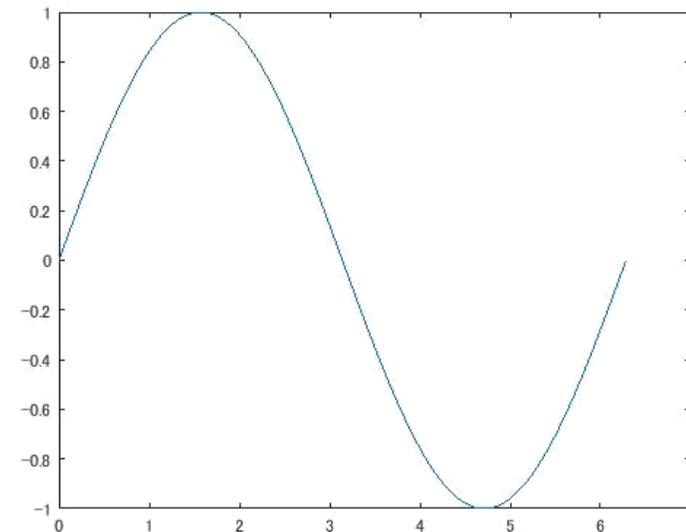
行列の乗算ではなく要素同士の乗算であることを示すために「.」を入れる必要がある。



プログラム例

```
>> t = 0:0.01:2*pi;  
>> y = sin(t)
```

```
>> plot(t,y)
```



📊 グラフ描画 拡張表現

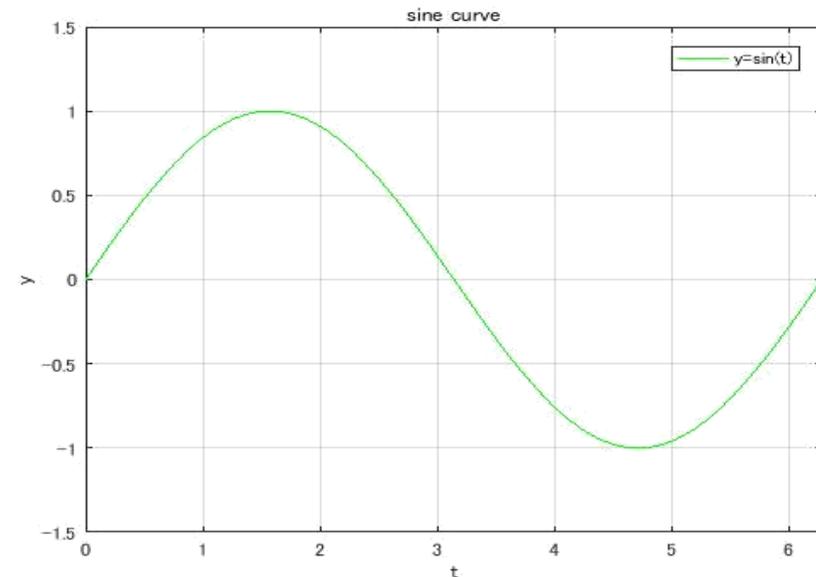
グラフにグリッドライン・タイトルなどを付けたい場合は、以下の関数を使用する。

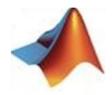
関数	説明
title("")	グラフのタイトル
xlabel("")	x軸のラベル
ylabel("")	y軸のラベル
grid	グリッドライン
axis()	軸のスケージング
legend("")	凡例
subplot	複数のaxisを作成

※title("")、xlabel("")、ylabel("")、legend("")の関数の()内は'(シングル quotes)が2つ
例) title('sine curve')

例題12

t が $0 \leq t \leq 2\pi$, 刻み幅は 0.01 ($t = 0:0.01:2\pi$) のとき、 $y = \sin t$ のグラフを描いてみよう。その後、下図のように、体裁を整えてみよう。このとき、線の色を緑にしてみよう。

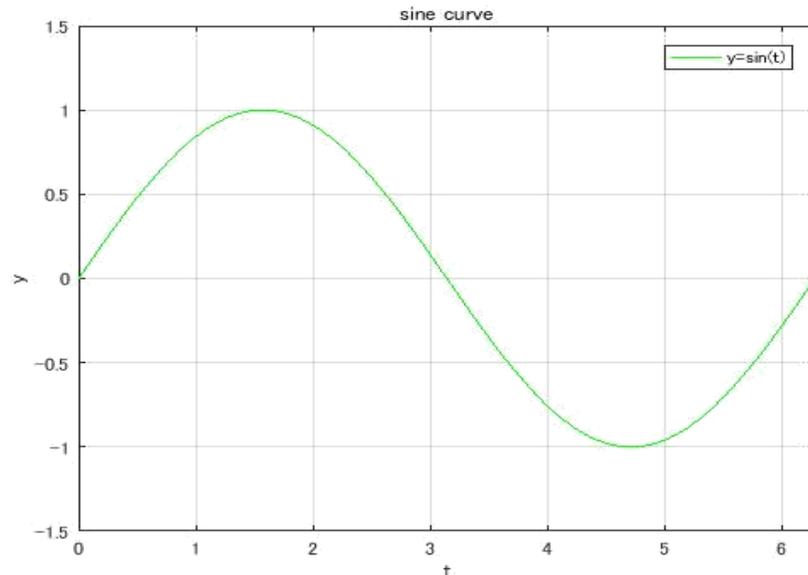




グラフ描画 拡張表現（タイトル・ラベル）

例題12の続き

t が $0 \leq t \leq 2\pi$, 刻み幅は 0.01 ($t = 0:0.01:2\pi$) のとき、 $y = \sin t$ のグラフを描いてみよう。
その後、下図のように、体裁を整えてみよう。



プログラム例

```
>> t=0:0.01:2*pi;  
>> y=sin(t);  
>> plot(t,y,'g')  
>> title('sine curve')  
>> xlabel('t')  
>> ylabel('y')  
>> grid on  
>> legend('y=sin(t)')  
>> axis([0 2*pi -1.5 1.5])
```

※ plot(t,y,'g') 入力後に出力されるグラフウィンドウは開いたまま、赤枠部分を入力しよう。

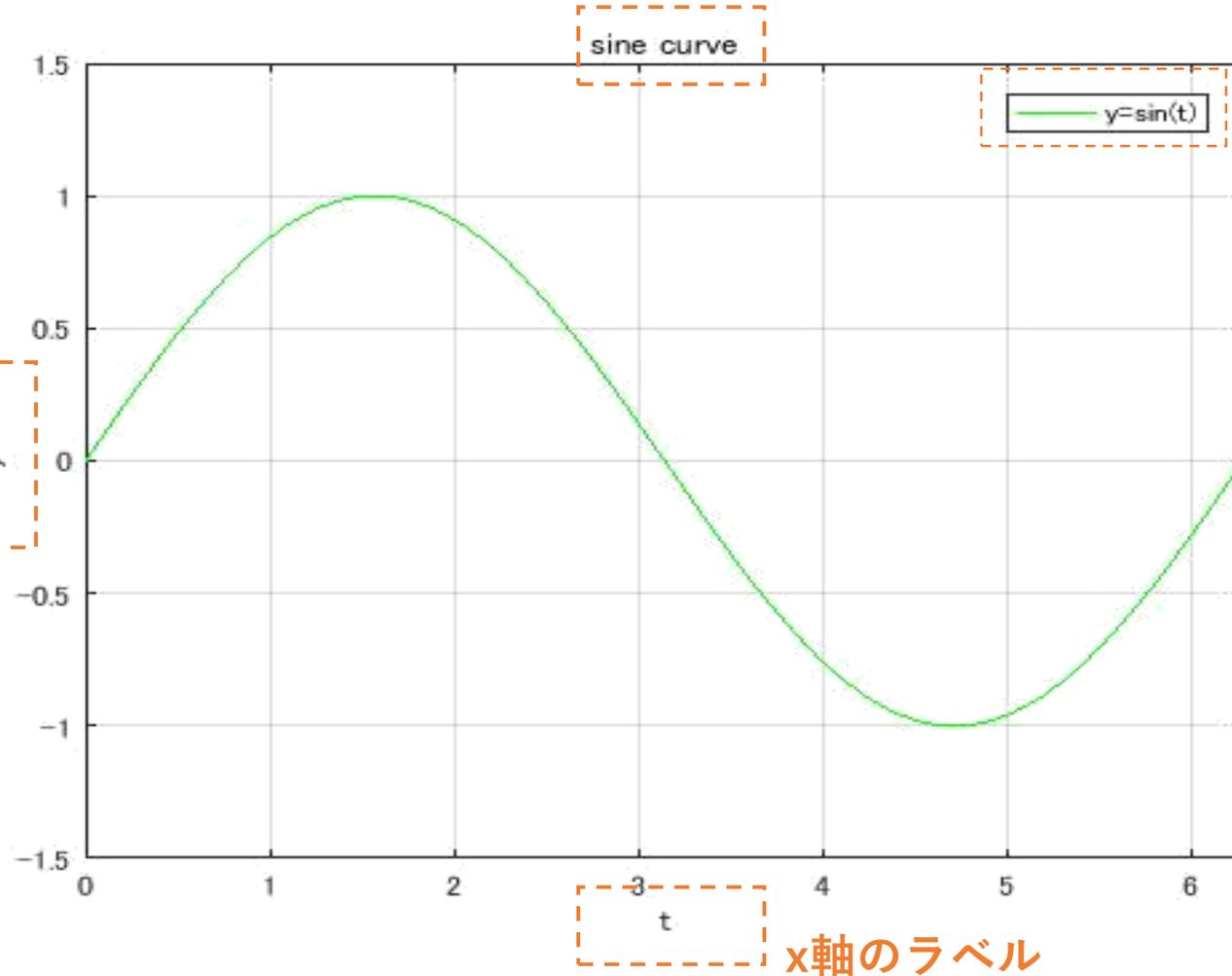
📊 グラフ描画 拡張表現

上部のグラフタイトル

y軸のラベル

凡例

軸の範囲を
x軸 : $0 \sim 2\pi$
y軸 : $-1.5 \sim 1.5$ に変更



x軸のラベル

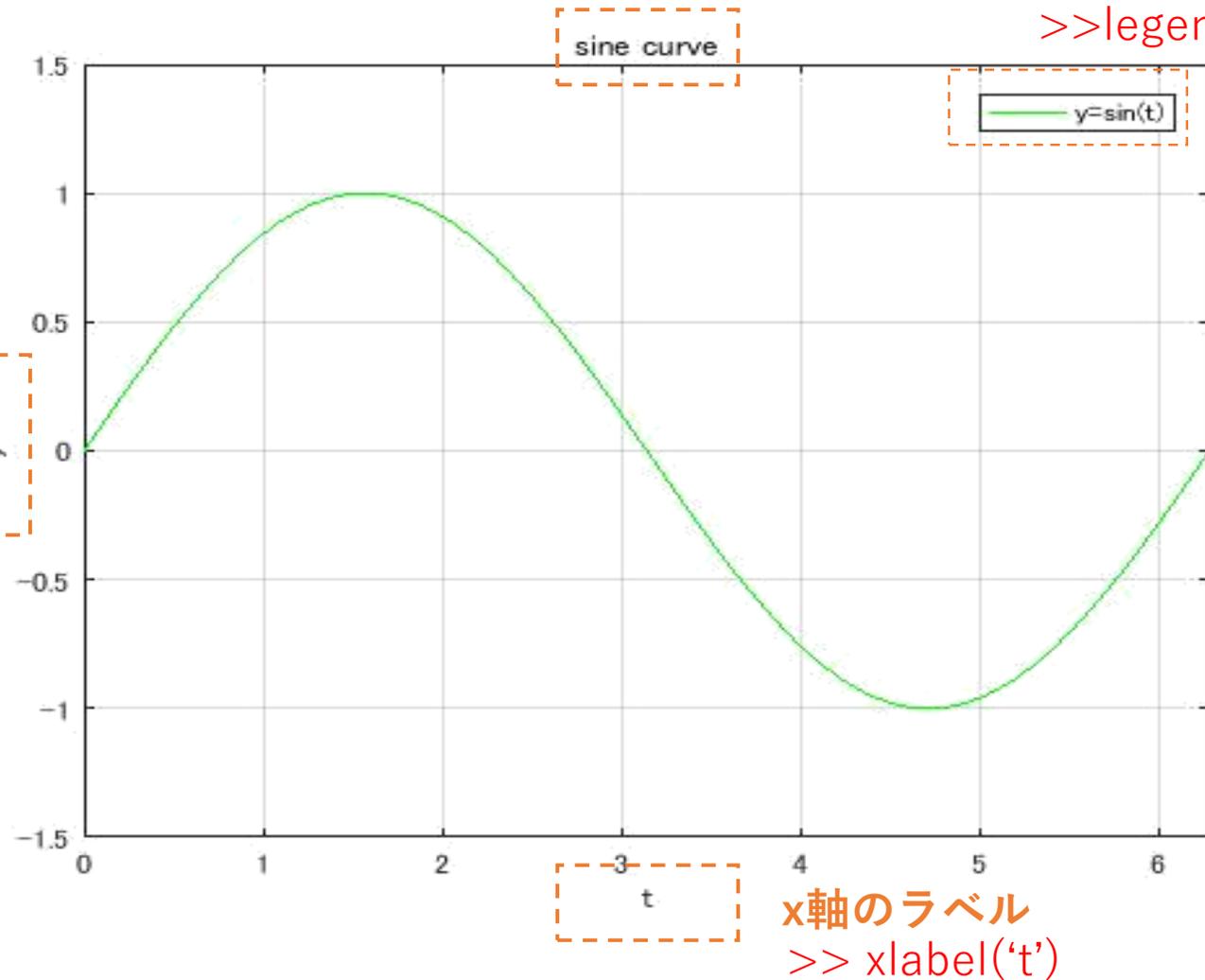
グラフ描画 拡張表現

上部のグラフタイトル
>> title('sine curve')

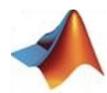
凡例
>> legend('y = sin(t)')

y軸のラベル
>> ylabel('y')

軸の範囲を
x軸 : $0 \sim 2\pi$
y軸 : $-1.5 \sim 1.5$ に変更
>> axis([0 2*pi -1.5 1.5])



x軸のラベル
>> xlabel('t')



例題13

t が $0 \leq t \leq 2\pi$, 刻み幅は 0.01 ($t = 0:0.01:2\pi$) のとき、

$$y_1 = \sin t$$

$$y_2 = \cos t$$

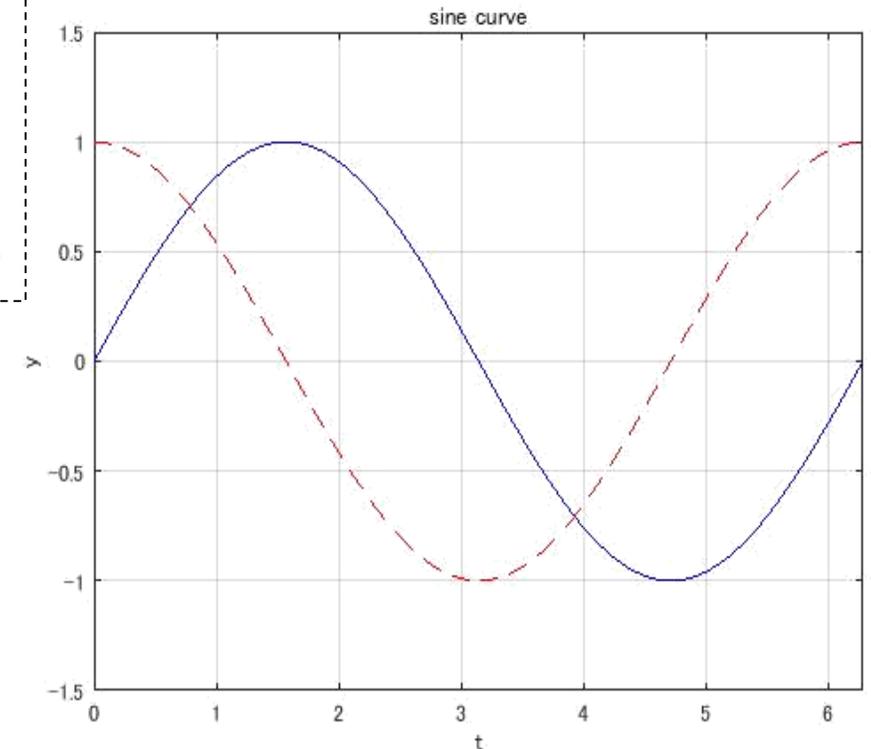
のグラフを**holdコマンド**を利用して同時に描画してみよう。

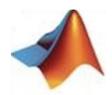
グラフの重ね描き

⇒ グラフを重ね描きする場合は、**holdコマンド (on/off)** を使用すると今のグラフに重ね描きすることができる。

⇒ 描画されたグラフは**clfコマンド**で消すことができる。

⇒ **closeコマンド**で、グラフが表示されているウィンドウを閉じることができる。





グラフ描写 拡張表現 (2つ以上のデータ)

例題13

t が $0 \leq t \leq 2\pi$, 刻み幅は 0.01 ($t = 0:0.01:2\pi$) のとき、

$$y_1 = \sin t$$

$$y_2 = \cos t$$

のグラフを**holdコマンド**を利用して同時に描画してみよう。

グラフの重ね描き

- ⇒ グラフを重ね描きする場合は、**holdコマンド(on/off)**を使用すると今のグラフに重ね描きすることができる。(offでやめる)
- ⇒ 描画されたグラフは**clfコマンド**で消すことができる。
- ⇒ **closeコマンド**で、グラフが表示されているウィンドウを閉じることができる。

プログラム例

```
t = 0 : 0.01 : 2*pi;  
y1 = sin(t);  
y2 = cos(t);
```

```
plot(t,y1,'b');  
hold on
```

```
plot(t,y2,'r--');
```

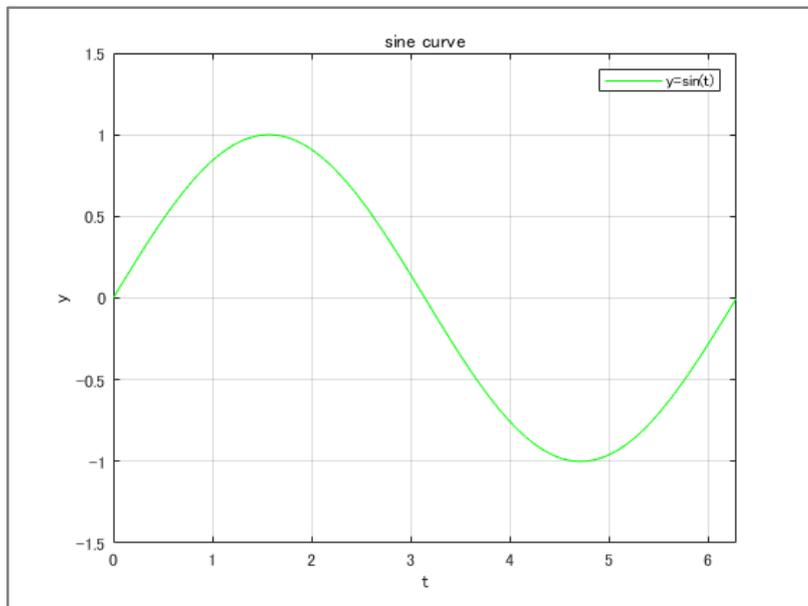
```
hold off
```

赤の鎖線でプロットする

```
title('sine curve');  
xlabel('t');  
ylabel('y');
```

```
grid on  
axis([0 2*pi -1.5 1.5])
```

グラフ描写 拡張表現 + α

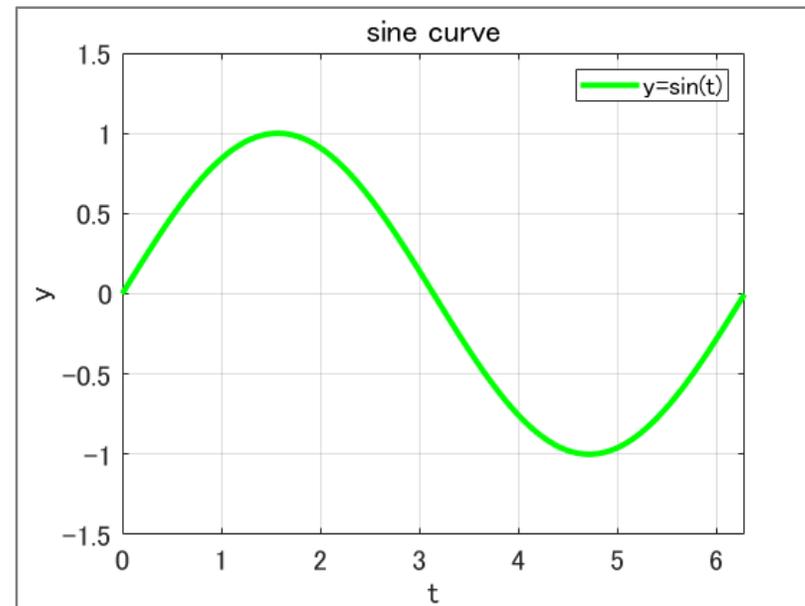


```
>> t = 0:0.01:2*pi;
>> y = sin(t);
>> plot(t,y,'g');
>> title('sine curve');
>> xlabel('t');
>> ylabel('y');
>> grid on
>> legend('y=sin(t)');
>> axis([0 2*pi -1.5 1.5]);
```



オプション

文字サイズや線の太さ
を変更して、より見や
すく体裁を整える。



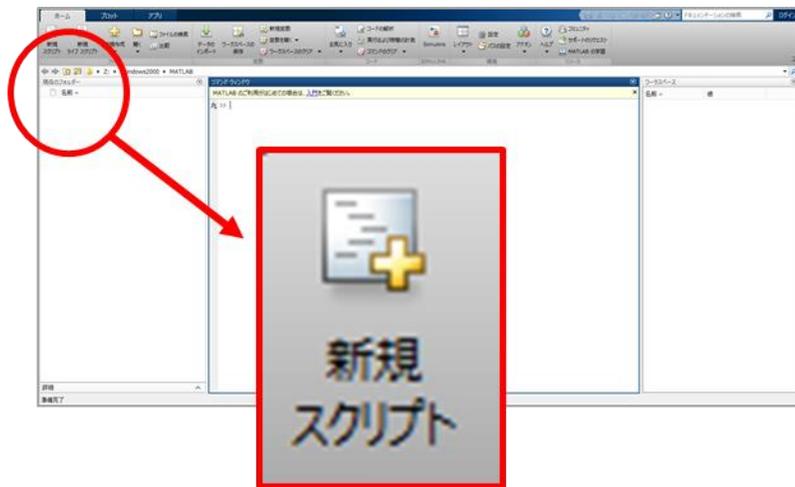
```
>> t = 0:0.01:2*pi;
>> y = sin(t);
>> plot(t,y,'g','LineWidth',3);
>> title('sine curve','FontSize',15);
>> xlabel('t','FontSize',15);
>> ylabel('y','FontSize',15);
>> grid on
>> legend('y=sin(t)','FontSize',15);
>> axis([0 2*pi -1.5 1.5]);
>> set(gca,'FontSize',15);
```

- ⇒線の太さ
- ⇒タイトルの文字サイズ
- ⇒ラベルの文字サイズ
- ⇒ラベルの文字サイズ
- ⇒凡例の文字サイズ
- ⇒軸の文字サイズ

スクリプトファイル作成①

長く複雑な（複数の命令を実行する）プログラムのまとまりを保存する方法。プログラムの内容がファイルとして保存できるため、後から編集・確認できるようになる。

実行手順 ① MATLAB付属のテキストエディタを表示



方法1：画面左上の”新規スクリプト”
クリック

実行例

```
>>edit
```

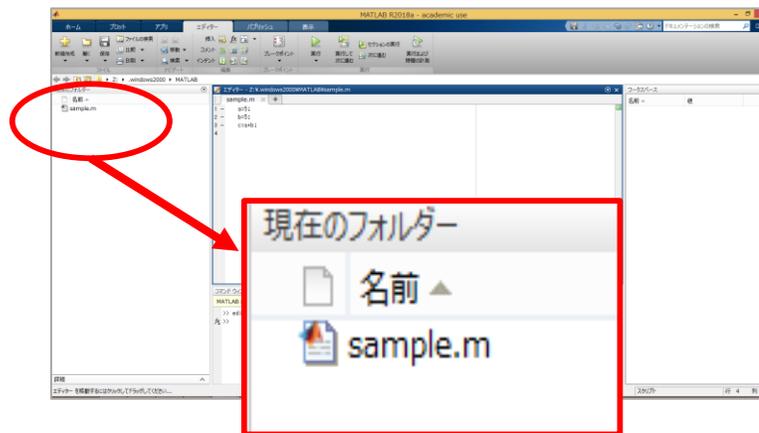
方法2：コマンドウィンドウ上に
”edit”と入力

🚀 スクリプトファイル作成②

② 出力されたテキストエディタにプログラムを書いて保存

注意⚠ 保存時の拡張子は “.m ”

③ プログラムの呼び出し



方法1：画面左のカレントディレクトリ上の
該当ファイルをクリック

実行例 (sample.mの場合)

```
>>sample
```

方法2：コマンドウィンドウ上に
ファイル名入力

スクリプトファイル作成③

Mファイル使用時に便利な関数

>>clear all; ワークスペースからすべてのオブジェクトを消去
⇒残っていた変数による予期せぬエラーを回避させる

>>clc; コマンドウィンドウのクリア
⇒入力・出力の全てをクリアにし、画面を整理させる

>>pack; ワークスペースのメモリ整理
⇒限られたメモリ量を有効に使用する

スクリプトファイル作成 例題

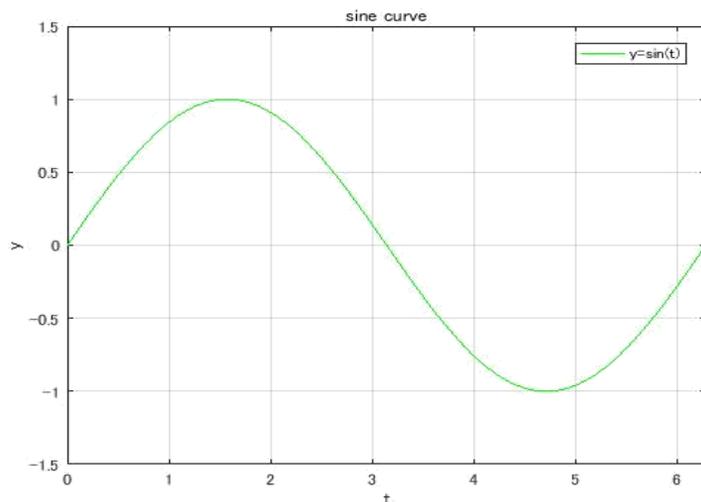
例題14

例題12のプログラムをスクリプト化してみよう。

スクリプト例

```
example14.m × +
1 %例題14の解答(スクリプト化)
2 t = 0 : 0.01 : 2*pi;
3 y = sin(t);
4
5 plot(t,y,'g');
6
7 title('sine curve');
8 xlabel('t');
9 ylabel('y');
10
11 grid on
12 legend('y=sin(t)')
13 axis([0 2*pi -1.5 1.5])
```

実行結果



実行例

現在のフォルダ内

- 名前 ▲
- example1.m
- example2.m
- example3.m
- example4.m
- example5.m
- example6.m
- example7.m
- example8.m
- example9.m
- example10.m
- example11.m
- example12.m
- example14.m
- text_exercise1.m
- text_exercise2.m
- text_exercise3.m
- text_exercise4.m
- text_exercise5.m

コマンドウィンドウ

MATLABのご利用がはじめての場合は、[入門](#)をご覧ください。

>> example14

>> example14

現在のフォルダ内にスクリプトがある状態で実行する！

ヘルプドキュメント (docコマンド)

① docコマンド

The screenshot shows the MATLAB help search interface. At the top, a search bar contains the text 'doc'. Below the search bar, the text 'doc の検索結果' is displayed. The search results are listed as follows:

- doc** - ヘルプ ブラウザーのリファレンス ページ
この MATLAB 関数 は、ヘルプ ブラウザーを開きます。
ドキュメンテーション > MATLAB > 環境と設定 > ヘルプとサポート
- helpwin** - すべての関数のヘルプ コメントへのアクセスを提供
この MATLAB 関数 は関数群に関するトピックを MATLAB ヘルプ ブラウザーに表示します。
ドキュメンテーション > MATLAB > 環境と設定 > ヘルプとサポート
- ヘルプとドキュメンテーション**
すべての関数には対応するドキュメンテーションが例と共に提供されており、関数の入出力や呼び出し構文が説明されています。
ドキュメンテーション > MATLAB > MATLAB 入門
- 関数のヘルプの参照方法**
MATLAB の各関数には対応するドキュメンテーションが例と共に提供されており、関数の入出力や呼び出し構文が説明されています。
ドキュメンテーション > MATLAB > 環境と設定 > ヘルプとサポート
- publish** - MATLAB ファイルの表示を指定した形式で生成します。
この MATLAB 関数 は、指定した MATLAB コード ファイルの表示と、共有可能な HTML 形式の出力を生成します。

Annotations on the screenshot include:

- A red box around the search bar containing 'doc'.
- A blue box around the search bar containing the text '② 調べたい関数名等を検索欄に入力'.
- A blue box around the first search result containing the text '③ ヘルプドキュメントを開く'.

ヘルプドキュメント (docコマンド)

51

ドキュメンテーション

[すべて](#) [例](#) [関数](#)

R2019b のドキュメンテーションを検索

[閉じる](#)[ドキュメンテーションのホーム](#)[MATLAB](#)[環境と設定](#)[ヘルプとサポート](#)**doc**[項目一覧](#)[構文](#)[説明](#)[例](#)[入力引数](#)[ヒント](#)[参考](#)[評価版](#) [製品の更新](#)

最新のリリースでは、このページがまだ翻訳されていません。 [このページの最新版は英語でご覧になれます。](#)

doc

R2019a

[ヘルプ ブラウザーのリファレンス ページ](#)[ページ内をすべて折りたたむ](#)

構文

```
doc  
doc name
```

説明

`doc` は、ヘルプ ブラウザーを開きます。ヘルプ ブラウザーが既に開かれているが表示されていない場合、`doc` により前面に移動され新しいタブが開かれます。

`doc name` は、関数、クラス、ブロックなど、`name` で指定された機能のドキュメンテーションを表示します。 [例](#)

- `name` に対応する MathWorks® のリファレンス ページがある場合、`doc` はヘルプ ブラウザーにそのページを表示します。`doc` コマンドではサードパーティまたはカスタムの HTML ドキュメンテーションは表示されません。
- `name` に対応するリファレンス ページがない場合、`doc` は `name.m` または `name.mlx` という名前のファイル内でヘルプ テキストを検索します。ヘルプ テキストがある場合、`doc` はそれをヘルプ ブラウザーに表示します。
- `name` に対応するリファレンス ページがなく、かつ関連付けられたヘルプ テキストがない場合、`doc` は `name` のドキュメンテーションを検索し、検索結果をヘルプ ブラウザーに表示します。

例

[すべて折りたたむ](#)[関数リファレンス ページの表示](#)

自由に試せる例題 (docコマンド)

① docコマンド

ヘルプ 例 x +

ドキュメンテーション すべて **例** 関数 ブロック アプリ

R2019b のドキュメンテーションを検索

評価版

例

MATLAB

基本的な行列演算

スペクトル解析のための FFT

一般的な 2 次元プロットプロットの作成

スプレッドシートデータへの読み取り

この例では、MATLAB® 言語で行列を操作するための基本的な手法と関数を示します。

この例では、スペクトル解析に対する関数 FFT の使用を示します。FFT の一般的な利用法は、時間領域のノイズの含まれる信号内に埋もれた信

この例では、MATLAB® でさまざまな 2 次元プロットを作成する方法を示します。

数値とテキストが混在するスプレッドシートから table ポートする。

ライブ スクリプトを開く

ライブ スクリプトを開く

ライブ スクリプトを開く

ライブ スクリプト

② 「例」をクリック

- カテゴリ
- MATLAB
 - Simulink
 - 5G Toolbox
 - Aerospace Blockset
 - Aerospace Toolbox
 - Antenna Toolbox
 - Audio Toolbox
 - Automated Driving Toolbox
 - AUTOSAR Blockset
 - Bioinformatics Toolbox
 - Communications Toolbox
 - Computer Vision Toolbox
 - Control System Toolbox
 - Curve Fitting Toolbox

③ カテゴリの中から選択

自由に試せる例題 (docコマンド)

ヘルプ

Deep Learning Toolbox — 例

ドキュメンテーション

すべて 例 関数 アプリ

R2019b のドキュメンテーション

目次 問題

« ドキュメンテーションのホーム

« 例

カテゴリ

- Deep Learning Toolbox**
- Deep Learning Toolbox 入門 8
- イメージを使用した深層学習 14
- 時系列、シーケンス、およびテキストを使用した深層学習 17
- 深層学習の調整および可視化 14
- 並列およびクラウドでの深層学習 8
- 深層学習の応用 41
- 深層学習のインポート、エクスポート、およびカスタマイズ 17
- 深層学習データの前処理 9
- 深層学習のコード生成 21
- 関数近似、クラスタリング、および制御 24

Deep Learning Toolbox 入門

coffee mug 0.83

Top 5

この例では、事前学習済みの深層畳み込みニューラル ネットワーク GoogLeNet を使用して、Web カメラのイメージをリアルタイムで分類

[スクリプトを開く](#)

新しいイメージを分類するための深層学習ネットワークの学習

この例では、転移学習を使用して、畳み込みニューラル ネットワークの再学習を行い、新しい一連のイメージを分類する方法を説明します

[ライブ スクリプトを開く](#)

Forecast with Updates

RMSE = 122.807

この例では、長短期記憶 (LSTM) ネットワークを使用して時系列データを予測する方法を説明します。

[ライブ スクリプトを開く](#)

bell pepper

Training Progress (08-Apr-2019 04:53:01)

この例では、長短期記憶 (LSTM) ネットワークを使用して時系列データを予測する方法を説明します。

[ライブ スクリプトを開く](#)

タイプ

ソースコードの確認 (editコマンド)

```
>> edit strcat  
fx >>
```

```
エディター - C:\Program Files\MATLAB\R2019b\toolbox\matlab\strfun\strcat.m  
strcat.m x +  
1 function t = strcat(varargin)  
2 %STRCAT Concatenate text.  
3 % COMBINEDSTR = STRCAT(S1, S2, ..., SN) horizontally concatenates text  
4 % in arrays S1, S2, ..., SN. Inputs can be combinations of character  
5 % vectors, character vectors in scalar cells, character arrays with the  
6 % same number of rows, same-sized cell arrays of character vectors,  
7 % scalar strings, or same-sized string arrays. If any input is a string  
8 % array, COMBINEDSTR is a string array. Otherwise, if any input is a cell  
9 % array, COMBINEDSTR is a cell array. Otherwise, COMBINEDSTR is a  
10 % character array.  
11 %  
12 %  
13 %  
14 %  
15 %  
16 %  
17 %  
18 %  
19 %  
20 %  
21 %  
22 %  
23 %  
24 %  
25 %  
26 %  
27 %  
28 %  
29 %  
30 %  
31 %  
32 %  
33 %  
34 %  
35 %  
36 %  
37 %  
38 %  
39 %  
40 %  
41 %  
42 %  
43 %  
44 %  
45 %  
46 %  
47 %  
48 %  
49 %  
50 %  
51 %  
52 %  
53 %  
54 %  
55 %  
56 %  
57 %  
58 %  
59 %  
60 %  
61 %  
62 %  
63 %  
64 %  
65 %  
66 %  
67 %  
68 %  
69 %  
70 %  
71 %  
72 %  
73 %  
74 %  
75 %  
76 %  
77 %  
78 %  
79 %  
80 %  
81 %  
82 %  
83 %  
84 %  
85 %  
86 %  
87 %  
88 %  
89 %  
90 %  
91 %  
92 %  
93 %  
94 %  
95 %  
96 %  
97 %  
98 %  
99 %  
100 %  
101 %  
102 %  
103 %  
104 %  
105 %  
106 %  
107 - end  
108 - s = s(1:pos-1);  
109 - notempty(1, i) = ~isempty(s);  
110 - scell{1, i} = s;  
111 - end  
112 - if n > 1  
113 - t = char(scell{notempty});  
114 - else  
115 - t = s;  
116 - end  
117
```


PID制御

サイト内検索

PID制御とは

PID制御は、古くから産業界で幅広く使用されているフィードバック制御の一種で、長年の経験の蓄積からも、実用化されているフィードバック制御の代表として、モーター速度制御や温度制御など応用先は様々です。PIDという文字は、比例 (P: Proportional)、積分 (I: Integral)、微分 (D: Differential) の頭文字に由来します。

図1は、一般的なフィードバック制御系のブロック線図を表しています。PID制御の仕組みは、図2に示すように、制御対象の出力信号に対して、比例演算、積分演算、そして、微分演算の3つの演算を行います。言い換えると、PID制御は、比例制御、積分制御、そして、微分制御が可能な制御系となります。制御理論の立場では、PID制御を制御系の中で、つまり、伝達関数を用いた周波数領域の世界の中

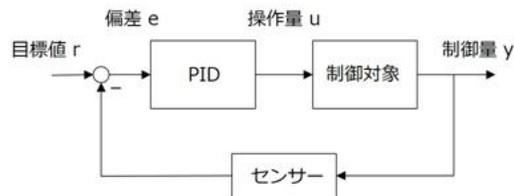


図1: フィードバック制御系

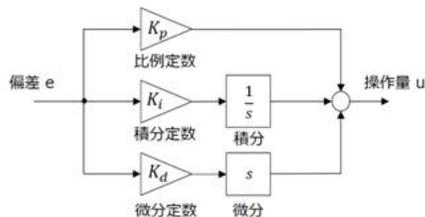


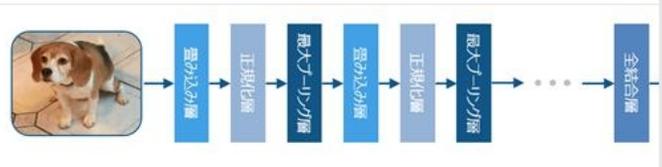
図2: PID制御器

畳み込みニューラルネットワーク (CNN)

サイト内検索

畳み込みニューラルネットワークとは

畳み込みニューラルネットワーク (Convolutional Neural Network: CNN) は、ニューラルネットワークで、特に画像認識の分野で優れた性能を発揮している。畳み込み層や「プーリング層」などの幾つかの特徴的な機能を持った層を積み重ねていく。



畳み込みニューラルネットワーク

この畳み込みニューラルネットワークは、主に一般物体認識と呼ばれる画像認識に使われているものですが、この圧倒的な性能の他に、転移学習による効果などもこの技術が普及した大きなきっかけになったと言えるでしょう。次節では、この技術についてみていきます。

畳み込みニューラルネットワークの仕組み

畳み込み層

一般的なニューラルネットワークでは層状にニューロンを配置し、前後の層が普通ですが、この畳み込みニューラルネットワークではこのニューロンと同じ手法を使うことで、画像の畳み込みに相当するような処理をニューロンで行っています。この層は「畳み込み層」と呼ばれ、この畳み込みニューラル

画像処理

サイト内検索

画像処理とは

画像処理(英語: Image Processing)とは、カメラ等で取り込んだデジタル画像から意図した情報を取り出すために処理加工を行うことをいいます。情報を取り出す画像解析とその解析をやすくするために前処理を合わせて指すのが一般的です。

画像処理の歴史は1960年代から人工衛星が取得した衛星画像処理の品質改善のために研究が始まり、CTやMRI等の医用画像処理に徐々に広がっていきました。その後、PCの価格低下と処理の高速化、デジタルカメラの普及により、一般用途での活用が急速に広がりました。また高解像度なデータを高速画像処理する必要性から、画像処理にFPGA等のハードウェアを使用するケースも増えてきています。

画像処理の利用と応用例

画像処理はノイズを取り除いてきれいな画像を出力する以外にも、最近では、コンピュータ上で特定の物体を見つける画像認識の活用が増えています。人間の視覚による直感的な判断をコンピュータが行えるようになることで、これまで人による目視検査に頼っていた生産ラインでの外観検査等をコンピュータによる画像検査に置き換える動きが進んでいます。

その応用は医用画像、顔認識、文字認識(OCR)といったより高度な対象に広がってきており、ロボット産業の市場拡大の傾向から、ロボットの目(画像のセンサ)として動体検出や3次元画像処理を取り扱うコンピュータビジョンという研究分野として発展しています。(画像認識とは)

画像処理での色空間

カラー JPEG, BMP形式等のデジタル画像データでは、各画素(ピクセル)をRGBの色空間で取り扱われます。画像処理の色調整にRGB色空間を利用すると、黄色等の中間色を作る場合にはRGB 3色の配合と最終的な色の関係理解をした上で調整する必要があります。そこで人間の直感に近い処理を可能にするためにHSV色空間を用いることがあります。HSVは色相(Hue)、彩度(Saturation)、明度(Value)の軸で色を表し、〇〇色をこのぐらいの明るさでという調整が容易となります。

Hue: 色相



まとめ

MATLABは

解析・モデリング・シミュレーションを
扱うことに適した数値計算処理ソフト

特徴

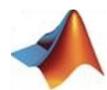
- ✓ データ型や大きさの宣言が不要で記述が簡単
- ✓ 豊富なオプションを持つ描画機能
- ✓ 他アプリケーションとの連携

MATLAB Campus-Wide Licenseとは

～いつでも、どこでも、誰でも、制約のない利用環境～

- 対象: 学校に所属している全ての教職員、学生
- 対象マシン: 学校管理のマシン、教職員/学生個人のマシン **台数無制限**
- 利用場所: 演習室、研究室、実験室、自宅あらゆる場所で利用可能
- 付加サービス: ブラウザやタブレット端末からの利用など、オンラインサービスを提供





MATLAB Campus-Wide License利用手順

59

利用手順

利用を希望する方は、以下の手順に従ってください。

1. MathWorksアカウントを[こちら](#)から作成してください。 **「math works account」** で検索

* 登録上の注意

1. 記載するメールアドレスは、**MeijiMailのアドレス (*@meiji.ac.jp) のみ**とします。
 2. 登録すると「**Verify Email Address**」というメールがMathWorks社から届きますので、アドレスの認証を行ってください。
2. 各自、MathWorks社サイトより製品をダウンロードし、インストールしてください。
インストールの詳細は、[MathWork社のHP](#)をご参照ください。
3. 利用開始から1年経過する毎に、ライセンス更新作業をしてください。
アプリケーションを起動し、メニューの[ヘルプ]→[ライセンス]から「現在のライセンスを更新する」を選択することで完了します。
なお、Matlabの継続利用とは別に、MathWorksアカウントの作成から1年経過する毎に、MathWorks社から再認証の確認メールが届きます。
そのメールにある手順に沿って再認証手続きをしてください。



MATLABについてもっと知りたいときは

オンライン自習システムへのアクセス（無償2時間コース）

<https://matlabacademy.mathworks.com/jp>



MATLAB入門（日本語）
2時間コース無料
（アカウント作成のみ必要）

MATLAB入門では、MATLAB
環境と操作方に慣れ、
MATLABを使うために必要な
基本スキルが身に付きます

MATLABのツールストリップ(画面上部)から「MATLAB の学習」をクリック
または「[matlab academy](#)」で検索してURLを見つけログイン
「[MATLAB入門でMATLABを学ぶ](#)」をクリック

MathWorks.com を検索

MATLAB 入門 30分

13.1 フロー制御

タスク1

```

index 特定の条件が満たされる時に、コードのセクションを実行したいとします。そのためには、if ステートメントを使用します。各 if ステートメントには1つの if 条件、1つの do 条件、および 1 つの do 条件を指定しなければなりません。条件が満たされた場合にのみ、if キーワードと end キーワードの間のコードが実行されます。

x = rand;
if x > 0.5
    y = 3; % x > 0.5 の時のみ実行
end
  
```

タスク2

WEBブラウザベースのクラウド環境上で、MATLABのプログラミング演習が可能

参考文献

- ・ 「MATLAB講習会 導入編 2019年度テキスト、補足資料」
『<https://www.meiji.ac.jp/isys/doc/internal/matlab01201808.pdf>』
『<https://www.meiji.ac.jp/isys/doc/internal/matlab01201808appendix.pdf>』
- ・ 池原雅章・奥田正浩・長井隆行（2006）
『だれでもわかるMATLAB 即戦力ツールブック』 培風館
- ・ 上阪吉則（2011）
『MATLABプログラミング入門』 牧野書店

明治大学生田図書館で借りることができます！

