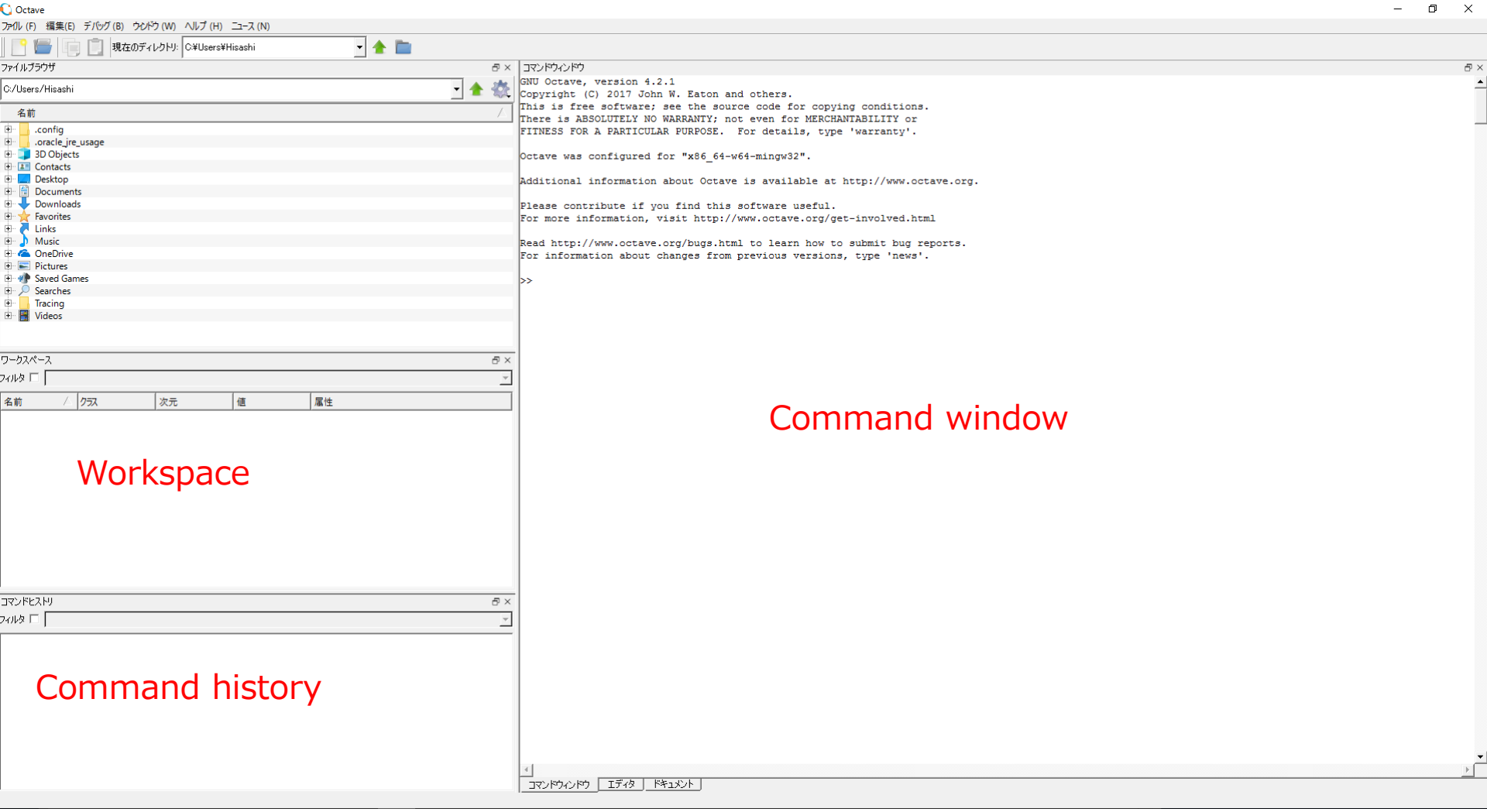
2. Fundamentals of Octave (&MATLAB)

- Octave GUI(Graphical User Interface)
- Command Window
- Scripts
- Variables
- Matrices
- Arithmetic operations & special values
- Mathematical functions
- Input/output with files
- Loops
- Conditional branch & flow control
- Plotting grpahs

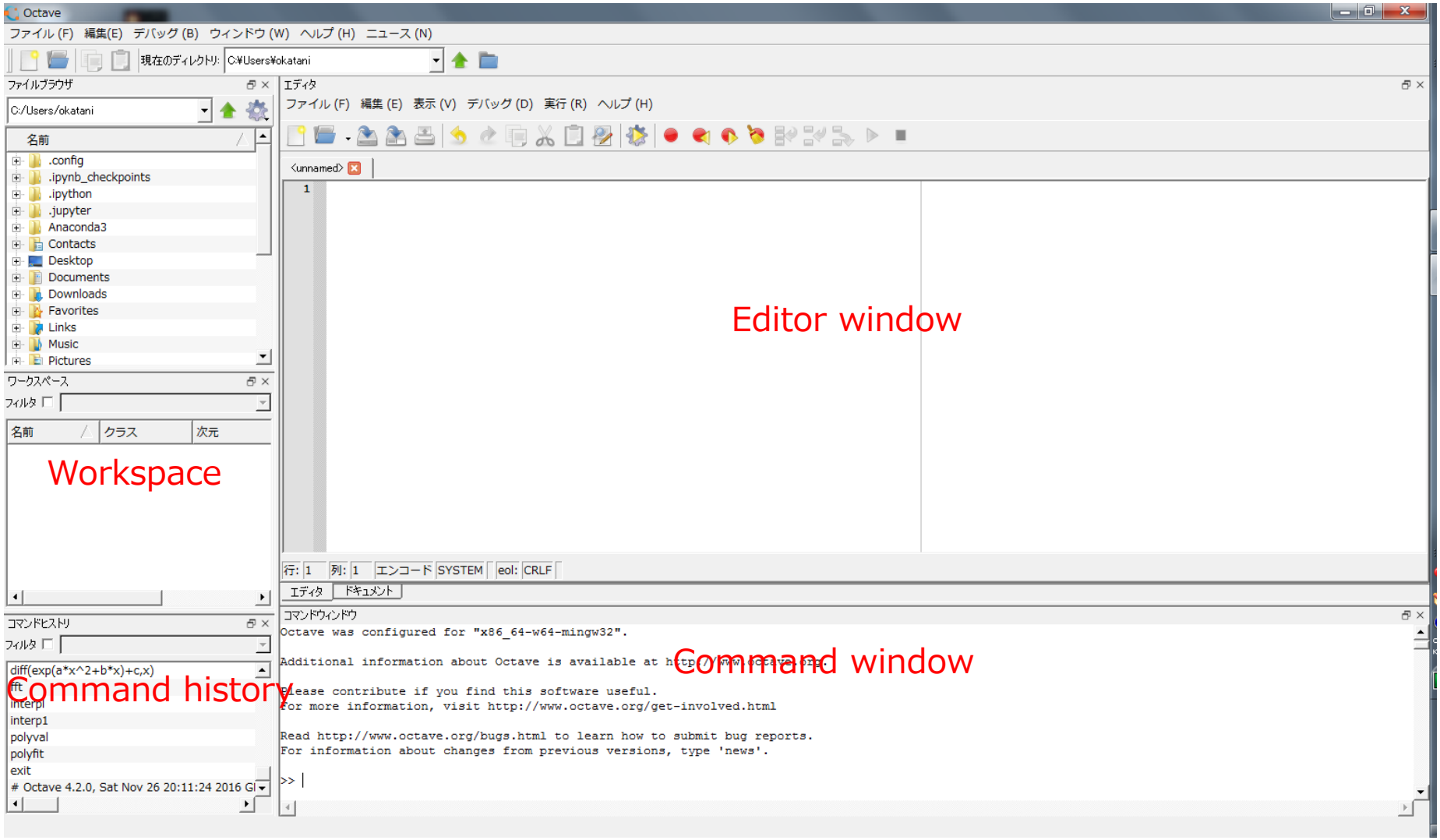
Octave GUI (起動直後)



Octave GUI（新規スクリプト作成後）

・ファイル→新規→新規のスクリプト

エディタ画面をドラック&ドロップで以下のような状態になる。



コマンドウィンドウの使用例

- 例: コマンドウィンドウの”>>”の横に“1+2” と入力後に「エンター」を押す。

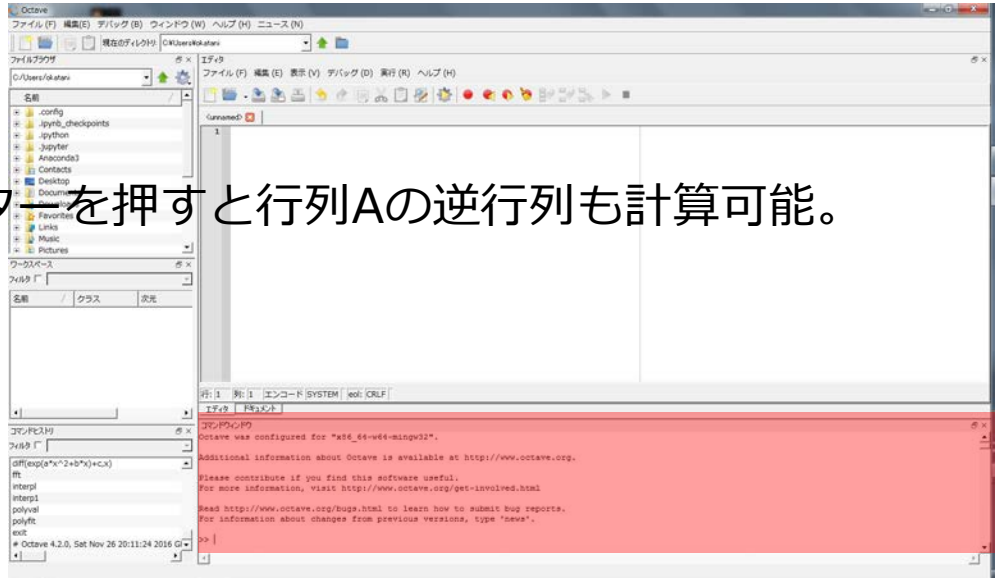
```
>> 1+2
ans =  3
>>
```

- 例:以下の様に2×2の行列を作成することも可能。

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

- 例:さらに”inv(A)” と入力してエンター を押すと行列Aの逆行列も計算可能。

```
>> inv(A)
ans =
  -2.00000   1.00000
   1.50000  -0.50000
```

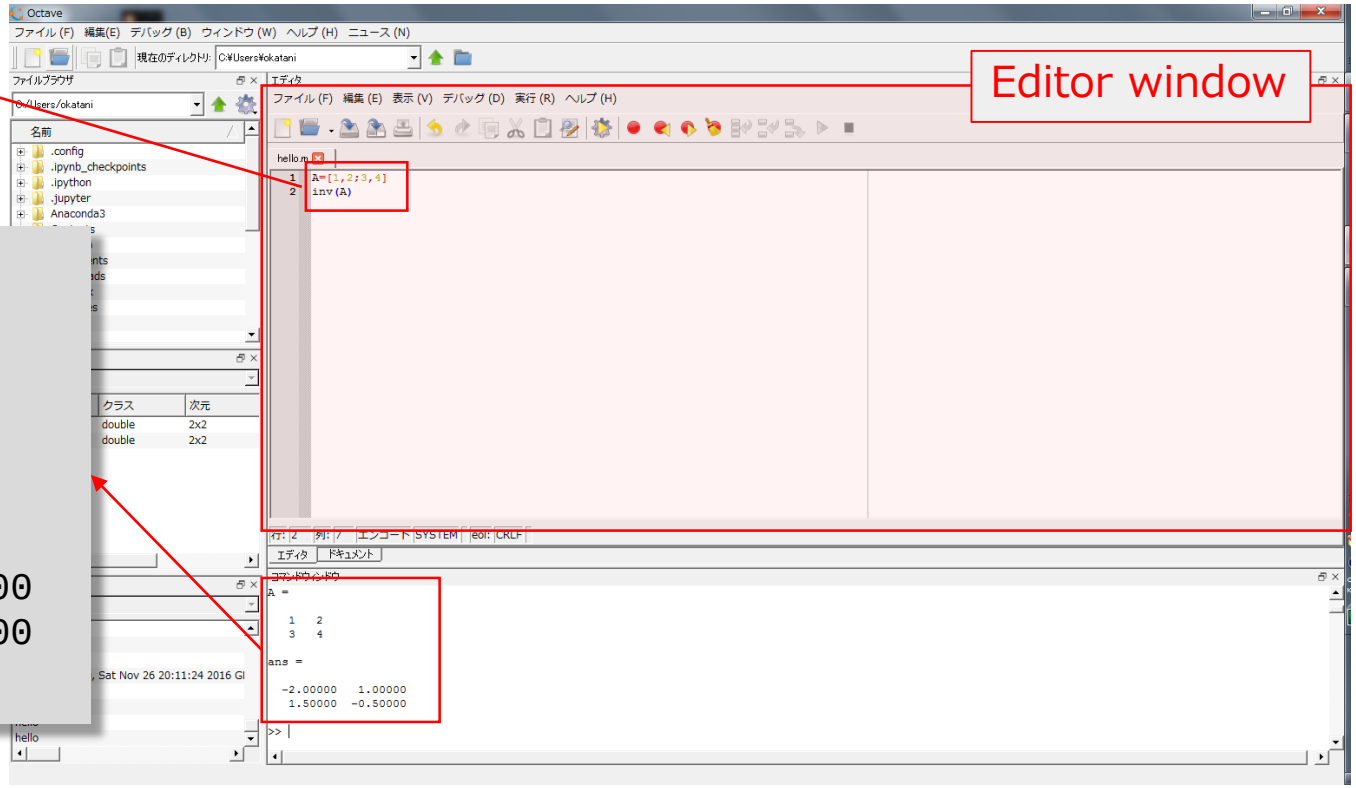


スクリプトファイルの作成

- 以下の内容をエディタウィンドウに入力し、ファイル→ファイルの保存を選択。ファイル名に “hello”と入力して“保存”をクリック。
  - “hello.m”というファイル名でこのスクリプトが保存される。
- コマンドウィンドウで “hello” と入力し「Enter」を押すとスクリプトが実行される。
  - エディタウィンドウ上の”実行”→”ファイルを保存して実行”でも同様

```
A=[1,2;3,4]
inv(A)
```

```
>> hello
A=[1,2;3,4]
A =
   1   2
   3   4
>> inv(A)
ans =
  -2.00000   1.00000
   1.50000  -0.50000
```



変数の使用例

- 先の例に示した**A**の様に変数の作成や使用が可能。
  - 変数の名は既に存在するファイル名や変数と異なる必要があります。
  - ただし、長さに制限はありません。(MATLABでは19文字以下の制限有り)

```
>> the_1st_variable=[1;2];
>> the_1st_variable
the_1st_variable =

   1
   2
```

- 英数字と’_’(アンダースコア)のみが変数に利用可能。
- 結果を表示したくない場合は最後に’;’(セミコロン)を入力

- 今まで作成した全ての変数がワークスペース上で利用可能。
- 次のように‘clear’コマンドを使用することで変数情報を削除可能。

```
>> clear A
```

行列の使い方

- Octave/Matlabにおいて最も基本的なデータの記述方式
- ‘,’(カンマ)と”;”(セミコロン)を用いることで様々なサイズの行列が作成可能。
  （’,’(カンマ)は列区切りを示し、’;’(セミコロン)は行区切り示す。）

2x3 matrix

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

3x2 matrix

```
>> B=[1,2;2,3;3,4]
B =
   1   2
   2   3
   3   4
```

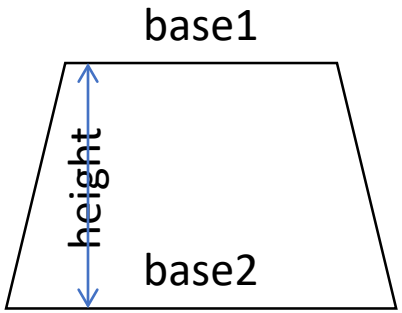
- 組み込み関数の”size”を用いることで行列サイズを調べることが可能。

```
>> size(A)
ans =
   2   3
>> size(B)
ans =
   3   2
```

算術演算と特殊な値

- 基本的演算子：+, -, *, /

```
>> base1=3.0;base2=5.0;height=3.0;
>> area=(base1+base2)*height/2
area =  12
```



- 累乗：^

```
>> 2^40
ans =    1.0995e+12
```

- 円周率：π

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

- 虚数単位：i or j

```
>> i
ans =  0 + 1i
>> j
ans =  0 + 1i
>> exp(-pi*i)
ans = -1.0000e+00 - 1.2246e-16i
```

$$e^{i\pi} = -1$$

(Euler's formula)

数値演算で用いる関数

- 三角関数
  - sin, sinh, asin, cos, cosh, acos, tan, tanh, atan, atan2
- 指数関数, 対数関数, その他
  - exp, log, log10, sqrt
- 行列要素への様々な操作
  - sum, max, min, sort, mod
- 絶対値や複素数に対する関数
  - abs, conj, imag, real

```
>> sin(pi/2)
ans =  1
>> sin(pi)
ans =    1.2246e-16
>> log(e)
ans =  1
```

```
>> A
A =
   1   2   3
   2   3   4
>> sum(A)
ans =
   3   5   7
>> sum(sum(A))
ans =  15
```

```
>> a=2.0-3.0j
a =  2 - 3i
>> imag(a)
ans = -3
>> real(a)
ans =  2
>> abs(-a)
ans =  3.6056
>> conj(a)
ans =  2 + 3i
```

ファイルへの入出力

- “save”コマンドで特定のファイルへ様々な変数の値を書き込むことが可能。

```
>> save(‘A.txt’, ‘A’)
```

- “load”コマンドで書き込まれた値を読み込むことも可能。

```
>> load(‘A.txt’)
>> A
A =
   1   2   3
   2   3   4
```

```
>> B=load(‘A.txt’)
>> B
B =
   1   2   3
   2   3   4
```

- さらに、ワークスペース内の全ての内容を特定のファイルへ”save”および”load”することも可能。

```
>> save(‘workspace1’)
```

```
>> load(‘workspace1’)
```

繰り返し

- 繰り返しコマンド（ `for index=start:step:end … end` ）

スクリプト

(*# ~~~：コメントアウト、プログラムに影響を与えない文)

```
# loop1.m
for i=1:10
  x = 2^i;
  printf('%d: %f¥n', i, x)
end
```

改行の意味
環境によっては”＼n”と
表記される

スクリプトファイル”loop1.m”を保存後、
コマンドウィンドウで”loop1”とタイプ

結果

```
>> loop1
1: 2.000000
2: 4.000000
3: 8.000000
4: 16.000000
5: 32.000000
6: 64.000000
7: 128.000000
8: 256.000000
9: 512.000000
10: 1024.000000
```

```
# loop2.m
# calculate position of a vehicle
# with a constant acceleration
a = 1.0; # acceleration
for t=0.0:0.5:3 # time
  y=.5*a*t^2; # position
  printf('%f: %f¥n', t, y)
end
```

スクリプトファイル”loop2.m”を保存後、
コマンドウィンドウで”loop2”とタイプ

```
>> loop2
0.000000: 0.000000
0.500000: 0.125000
1.000000: 0.500000
1.500000: 1.125000
2.000000: 2.000000
2.500000: 3.125000
3.000000: 4.500000
```

条件付き分岐

- if-elseif-else-end structure

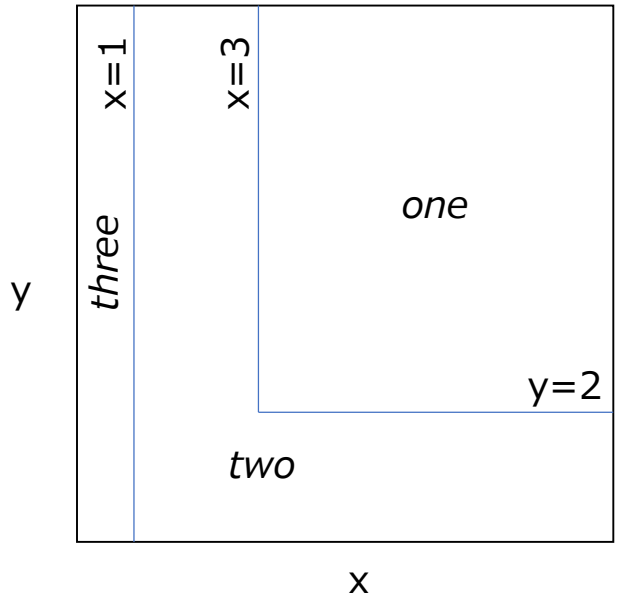
スクリプト

```
#ifelse.m
if x > 3.0 && y > 2.0
  disp('one')
elseif x > 1.0
  disp('two')
else
  disp('three')
end
```

Logical AND
(if both are true)
論理積

```
#ifelse2.m
if x < 3.0 || y < 2.0
  if x < 1.0
    disp('three')
  else
    disp('two')
  end
else
  disp('one')
end
```

Logical OR
(if either is true)
論理和



結果

```
>> x=4;y=5;
>> ifelse
one
>> x=2;y=5;
>> ifelse
two
>> x=y=0;
>> ifelse
three
```

比較演算子

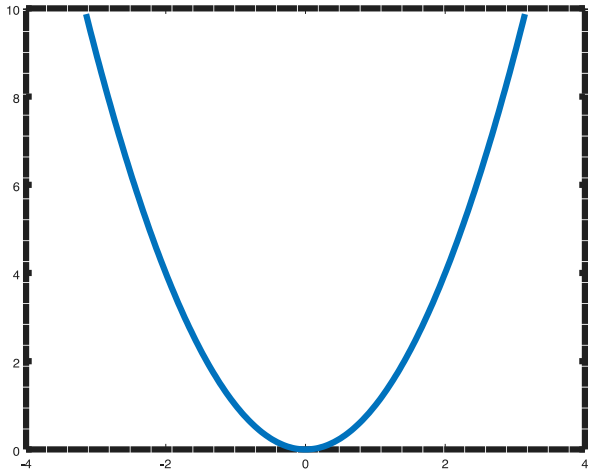
```
x <= y   less
than or equal

x == y   equal

x >= y
greater than
or equal

x != y
not equal
```

グラフ描画

- plot(x,y)： x,y共にベクトル(ないしは行列)であり、
  その要素数は同一で、yの要素はxの要素の対でプロットされる。

```
>> x=-pi:pi/100:pi;
>> y=x.^2;
>> plot(x,y)
```

'.^' はベクトルの各要素の
平方を表す

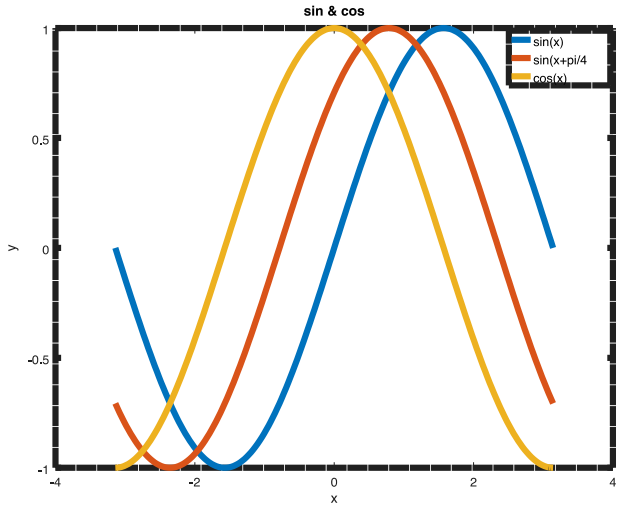


- 一つのグラフに複数のグラフを描画

```
>> plot(x,sin(x),x,sin(x+.25*pi),x,cos(x))
```

- 軸ラベルやタイトル、凡例を表示

```
>> xlabel('x'), ylabel('y'), title('sin & cos')
>> legend('sin(x)','sin(x+pi/4)','cos(x)')
```



- フォントサイズの変更(ただし、プロット前に記載)

```
>> set(0,"defaultaxesfontsize",20)
>> set(0,"defaulttextfontsize",20)
```

演習 2.1 (宿題)

- ある3桁の自然数について、各桁の３乗の和が元の数と等しくなる数字を全て求めよ。
- 例：153 ($1^3+5^3+3^3 = 153$)
- 下記のスクリプトを修正して条件を満たす数を求めよ

```
for i = 100:999
  i1 = mod(i, 10);
  i2 = mod(floor(i/10), 10);
  i3 = floor(i/100);
  disp([i3 i2 i1])
end
```

Hint: このスクリプトは全ての３桁の数に対し、各桁の数を抜き出すスクリプトです。

- (例えば)下記の空欄を埋めて、３乗和が元の数と等しくなる数字を求めよ

```
for i3 = 1:9
  for i2 = 0:9
    for i1 = 0:9
      [          ]

    end
  end
end
```

提出先：
東北大学ｲﾝﾀｰﾈｯﾄｽｸｰﾙ(ISTU)上で提出
もしくは
Email: hisashi.kino.a1＠tohoku.ac.jp
shimada@m.tohoku.ac.jp

〆切：
2017年6月22日の午前8:50