MATLAB 기초

전북대학교 전자공학부
이태희

기본개념

- 함수도움말 : *help*
- 화면 정리 : *clc*
- 화면 및 변수 정리 : *clear all*
- 기본연산자 : + - * / ^
- 연산순서
  1. 괄호 : ()
  2. 곱셉과 나눗셈 : * / ^
  3. 덧셈과 뺄셈 : + -
  - ➢ 같은 수준에서는 왼쪽부터 연산
  - ➢ 오류

```
>> x=5;
>> 3x
??? 3x
     |
Error: Unexpected MATLAB expression.

>> 3*x

ans =

    15
```

수의 형태

Type	Examples
Integer	1362, −217897
Real	1.234, −10.76
Complex	$3.21 - 4.3i \quad (i = \sqrt{-1})$
Inf	Infinity (result of dividing by 0)
NaN	Not a Number, 0/0

- 수가 매우 작거나 클 경우 “e”를 사용하여 출력된다.

`-1.3412e+03` $= -1.3412 \times 10^{3} = -1341.2$
`-1.3412e-01` $= -1.3412 \times 10^{-1} = -0.13412$

Command	Example of Output
`>>format short`	`31.4162`(4–decimal places)
`>>format short e`	`3.1416e+01`
`>>format long e`	`3.141592653589793e+01`
`>>format short`	`31.4162`(4–decimal places)
`>>format bank`	`31.42`(2–decimal places)

- 기본 수 형식으로 돌아갈 때는 “*format*”

변수

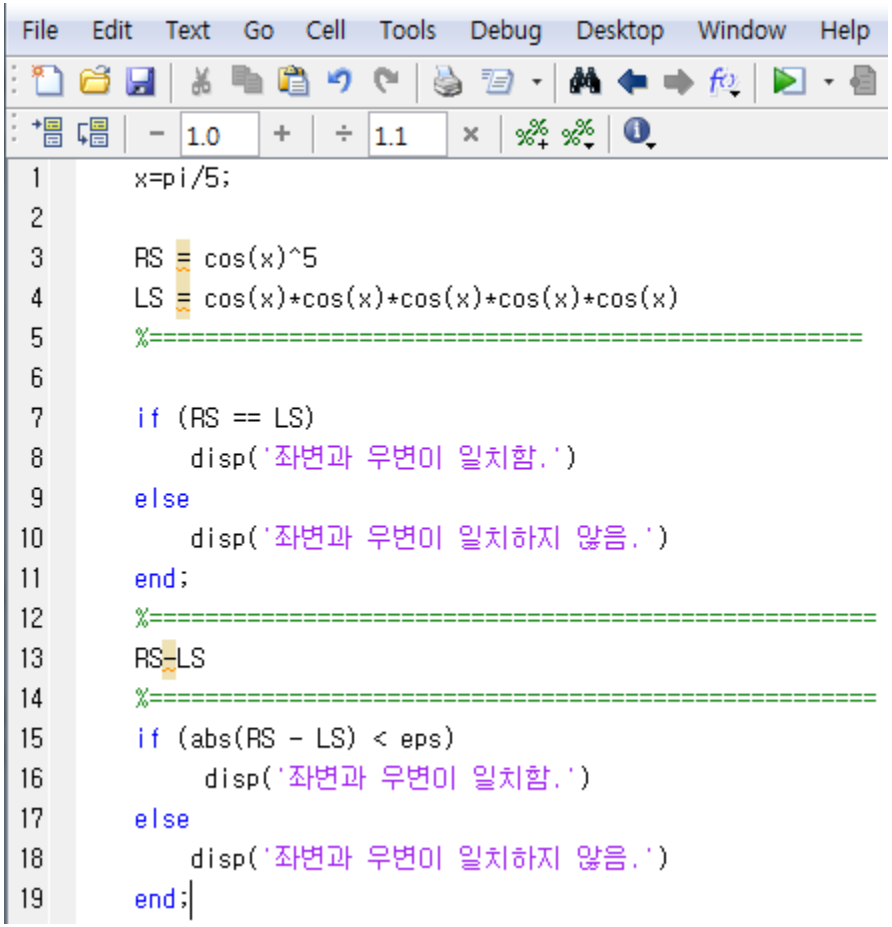
- 문자와 기호의 조합으로 설정가능
  예: `NetCost, Left2Pay, x3, X3, z25c5`
- 불가능한 경우
  - 숫자가 변수명의 제일 앞일경우
  - 특수문자가 들어갈 경우
  - 예: `Net-Cost, 2pay, %x, @sign`
- 특수한 변수들
  - *eps* : eps = 2.2204e-16 = 2^{-54}
  - *pi* : pi = 3.14159... = π.
  - *i,j* : 허수

```
>> 3-2^4
ans =
   -13
>> ans*5
ans =
   -65
```

```
>> x = 3-2^4
x =
   -13
>> y = x*5
y =
   -65
```

```
>> i,j, i=3
ans = 0 + 1.0000i
ans = 0 + 1.0000i
i   = 3
```

변수


File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
x=pi/5;
RS = cos(x)^5
LS = cos(x)*cos(x)*cos(x)*cos(x)*cos(x)
%==
if (RS == LS)
disp('좌변과 우변이 일치함.')
else
disp('좌변과 우변이 일치하지 않음.')
end;
%==
RS-LS
%==
if (abs(RS - LS) < eps)
disp('좌변과 우변이 일치함.')
else
disp('좌변과 우변이 일치하지 않음.')
end;

출력

- 결과를 출력하고 싶지 않을 때 : ;

```
>> x=-13; y = 5*x, z = x^2+y
y =
   -65
z =
   104
>>
```

- 연습

i) `-2^3+9`
ii) `2/3*3`
iii) `3*2/3`
iv) `3*4-5^2*2-3`
v) `(2/3^2*5)*(3-4^3)^2`
vi) `3*(3*4-2*5^2-3)`

함수

- 삼각함수 : *sin, cos, tan, asin, acos, atan*

```
>> x = 5*cos(pi/6), y = 5*sin(pi/6)
x =
    4.3301
y =
    2.5000
```

```
>> acos(x/5), asin(y/5)
ans = 0.5236
ans = 0.5236
```

- 기본 함수들 : *sqrt, exp, log, log10*

```
>> x = 9;
>> sqrt(x),exp(x),log(sqrt(x)),log10(x^2+6)
ans =
     3
ans =
   8.1031e+03
ans =
    1.0986
ans =
    1.9395

>> format long e, exp(log(9)), log(exp(9))
ans = 9.000000000000002e+00
ans = 9
>> format short
```

함수

- 사용자 지정함수 설정 : *inline*

```
>> f = inline('x^2 + x + 1', 'x')

f =

     Inline function:
     f(x) = x^2 + x + 1

>> f(4)

ans =

    21
```

벡터

- 벡터 변수값 선언 : *[값, 값 , 값]* 또는 *[값 값 값]*

```
>> v = [ 1 3, sqrt(5)]
v =
    1.0000    3.0000    2.2361
>> length(v)
ans =
     3
```

```
>> v2 = [3+ 4 5]
v2 =
     7     5
>> v3 = [3 +4 5]
v3 =
     3     4     5
```

- 벡터변수의 연산

```
>> v + v3
ans =
    4.0000    7.0000    7.2361
>> v4 = 3*v
v4 =
    3.0000    9.0000    6.7082
>> v5 = 2*v -3*v3
v5 =
   -7.0000   -6.0000  -10.5279
>> v + v2
??? Error using ==> +
Matrix dimensions must agree.
```

- 기존 변수를 이용한 벡터변수 선언

```
>> w  = [1 2 3],  z = [8 9]
>> cd = [2*z,-w], sort(cd)
w =
     1     2     3
z =
     8     9
cd =
    16    18    -1    -2    -3
ans =
    -3    -2    -1    16    18
```

- 벡터변수의 요소 수정

```
>> w(2) = -2, w(3)
w =
     1    -2     3
ans =
     3
```

벡터

- 규칙적으로 연속되는 값 : *시작값:간격:종료값*

```
>>  1:4
ans =
     1     2     3     4
>> 3:7
ans =
     3     4     5     6     7
>> 1:-1
ans =
     []
```

```
>> 0.32:0.1:0.6
ans =
    0.3200    0.4200    0.5200
>> -1.4:-0.3:-2
ans =
   -1.4000   -1.7000   -2.0000
```

- 벡터로의 이용

```
>>  r5 = [1:2:6, -1:-2:-7]
r5 =
     1     3     5    -1    -3    -5    -7
```

```
>> r5(3:6)
ans =
     5    -1    -3    -5
```

To get alternate entries:

```
>> r5(1:2:7)
ans =
     5    -1    -3    -5
```

벡터

- 열벡터 : *[값; 값; 값]*

```
>> c = [ 1; 3; sqrt(5)]
c =
    1.0000
    3.0000
    2.2361
>> c2 = [3
4
5]
c2 =
     3
     4
     5
>> c3 = 2*c - 3*c2
c3 =
   -7.0000
   -6.0000
  -10.5279
```

벡터

- 벡터의 전치 : ‘

```
>> w, w', c, c'
w =
     1    -2     3
ans =
     1
    -2
     3
c =
    1.0000
    3.0000
    2.2361
ans =
    1.0000    3.0000    2.2361
>> t = w + 2*c'
t =
    3.0000    4.0000    7.4721
>> T = 5*w'-2*c
T =
    3.0000
  -16.0000
   10.5279
```

벡터

- 복소수 벡터의 전치 : .’

```
>> x = [1+3i, 2-2i]
ans =
   1.0000 + 3.0000i   2.0000 - 2.0000i
>> x’
ans =
   1.0000 - 3.0000i
   2.0000 + 2.0000i

>> x.’
ans =
   1.0000 + 3.0000i
   2.0000 - 2.0000i
```

벡터의 연산

- 곱셉 : *

$$\underline{u} = [u_1, u_2, \ldots, u_n], \quad \underline{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \rightarrow \underline{u}\,\underline{v} = \sum_{i=1}^{n} u_i v_i$$

$$\underline{u} = [10, -11, 12] \quad \underline{v} = \begin{bmatrix} 20 \\ -21 \\ -22 \end{bmatrix} \rightarrow \underline{u}\,\underline{v} = 10 \times 20 + (-11) \times (-21) + 12 \times (-22) = 167$$

```
>> u = [ 10, -11, 12], v = [20; -21; -22]
>> prod = u*v
```

```
>> w = [2, 1, 3], z = [7; 6; 5]
w =
     2     1     3
z =
     7
     6
     5
```

벡터의 연산

```
>> u*w
??? Error using ==> *
Inner matrix dimensions must agree.

>> u*w'
ans =
    45
>> u*u'
ans =
   365
>> v'*z
ans =
   -96
```

벡터의 연산

- 놈(norm) : *norm(값)*

$$\|\underline{u}\| = \sqrt{\sum_{i=1}^{n} |u_i|^2}$$

```
>> [ sqrt(u*u'), norm(u)]
ans =
   19.1050    19.1050
```

벡터의 연산

- 벡터의 Dot Product : .*

$$\underline{u} \cdot \underline{v} = [u_1 v_1, u_2 v_2, \ldots, u_n v_n]$$

```
>> u.*w
ans =
      20 -11 36
>> u.*v'
ans =
      200 231 -264
>> v.*z, u'.*v
ans =
      140 -126 -110
ans =
      200 231 -264
```

벡터의 연산

- 벡터의 Dot Division : ./

```
>> a = 1:5, b = 6:10, a./b
a =
     1     2     3     4     5
b =
     6     7     8     9    10
ans =
 0.1667  0.2857  0.3750  0.4444  0.5000
>> a./a
ans =
     1     1     1     1     1
>> c = -2:2, a./c
c =
    -2    -1     0     1     2
Warning: Divide by zero
ans =
 -0.5000  -2.0000  Inf    4.0000   2.5000
>> a.*b -24, ans./c
ans =
   -18   -10     0    12    26

Warning: Divide by zero
ans =
     9    10   NaN    12    13
```

벡터의 연산

- 벡터의 Dot Power : .^

```
>> u.^2
ans =
   100   121   144
>> u.*u
ans =
   100   121   144
>> u.^4
ans =
       10000       14641       20736
>> v.^2
ans =
   400
   441
   484
>> u.*w.^(-2)
ans =
    2.5000  -11.0000    1.3333
```

행렬

- 행렬의 선언 : *[값 값 값 ; 값 값 값]*

$$A = \begin{bmatrix} 5 & 7 & 9 \\ 1 & -3 & -7 \end{bmatrix}$$

```
>> A = [5 7 9
        1 -3 -7]
A =
     5     7     9
     1    -3    -7
>> B = [-1 2 5; 9 0 5]
B =
    -1     2     5
     9     0     5
>> C = [0, 1; 3, -2; 4, 2]
C =
     0     1
     3    -2
     4     2
>> D = [1:5; 6:10; 11:2:20]
D =
     1     2     3     4     5
     6     7     8     9    10
    11    13    15    17    19
```

행렬

- 행렬의 크기 : *size(행렬)*

```
>> size(A), size(x)
ans =
     2     3
ans =
     3     1
>> size(ans)
ans =
     1     2
```

```
>> [r c] = size(A'), S = size(A')
r =
     3
c =
     2
S =
     3     2
```

- 행렬의 전치 : '

```
>> D, D'
D =
     1     2     3     4     5
     6     7     8     9    10
    11    13    15    17    19
ans =
     1     6    11
     2     7    13
     3     8    15
     4     9    17
     5    10    19
```

```
>> size(D), size(D')
ans =
     3     5
ans =
     5     3
```

행렬

- 특수한 행렬
  - 일행렬 : *ones(n,m)*
  - 영행렬: *zeros(n,m)*
  - 단위행렬: *eye(n)*
  - 대각행렬: *diag(a)*

```
>> P = ones(2,3)
P =
     1     1     1
     1     1     1
>> Z = zeros(2,3), zeros(size(P'))
Z =
     0     0     0
     0     0     0
ans =
     0     0
     0     0
     0     0
```

```
>> I = eye(3), x = [8; -4; 1], I*x
I =
     1     0     0
     0     1     0
     0     0     1
x =
     8
    -4
     1
ans =
     8
    -4
     1
```

행렬

$$D = \begin{bmatrix} -3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

```
>> D = [-3 0 0; 0 4 0; 0 0 2]
D =
    -3     0     0
     0     4     0
     0     0     2
```

```
>>  d = [-3 4 2], D = diag(d)
d =
    -3     4     2
D =
    -3     0     0
     0     4     0
     0     0     2
```

```
>> F = [0 1 8 7; 3 -2 -4 2; 4 2 1 1]
F =
     0     1     8     7
     3    -2    -4     2
     4     2     1     1
>> diag(F)
ans =
     0
    -2
     1
```

행렬

- 행렬의 선언

```
>> C=[0 1; 3 -2; 4 2]; x=[8;-4;1];
>> G = [C x]
G =
     0     1     8
     3    -2    -4
     4     2     1
>> A, B, H = [A; B]
A =
     5     7     9
     1    -3    -7
B =
    -1     2     5
     9     0     5
ans =
     5     7     9
     1    -3    -7
    -1     2     5
     9     0     5
```

```
>> J = [1:4; 5:8; 9:12; 20 0 5 4]
J =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    20     0     5     4

>> K = [ diag(1:4) J; J' zeros(4,4)]
K =
   1   0   0   0   1   2   3   4
   0   2   0   0   5   6   7   8
   0   0   3   0   9  10  11  12
   0   0   0   4  20   0   5   4
   1   5   9  20   0   0   0   0
   2   6  10   0   0   0   0   0
   3   7  11   5   0   0   0   0
   4   8  12   4   0   0   0   0
```

- 그래픽 표현 : *spy(행렬)*

```
>> spy(K)
```

행렬

- 행렬의 선언

```
>> x = 0:0.1:0.5;
>> y = 4*sin(3*x); u = 3*sin(4*x);
>> [ x' y' u']
ans =
         0         0         0
    0.1000    1.1821    1.1683
    0.2000    2.2586    2.1521
    0.3000    3.1333    2.7961
    0.4000    3.7282    2.9987
    0.5000    3.9900    2.7279

>>  x = (0:0.1:0.5)';
>> [x 4*sin(3*x) 3*sin(4*x)]
```

행렬

- 행렬 요소의 편집

```
>> J
J =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    20     0     5     4
>> J(1,1)
ans =
     1
>> J(2,3)
ans =
     7
>> J(4,3)
ans =
     5
>> J(4,5)
??? Index exceeds matrix dimensions.
```

```
>> J(4,1) = J(1,1) + 6
J =
     1     2     3     4
     5     6     7     8
     9    10    11    12
     7     0     5     4
>> J(1,1) = J(1,1) - 3*J(1,2)
J =
    -5     2     3     4
     5     6     7     8
     9    10    11    12
     7     0     5     4
```

행렬

- 행렬 요소의 편집

```
>> J(:,3)          % 3rd column
ans =
     3
     7
    11
     5
>> J(:,2:3)        % columns 2 to 3
ans =
     2     3
     6     7
    10    11
     0     5
>> J(4,:)          % 4th row
ans =
     7     0     5     4
>> J(2:3,2:3)      % rows 2 to 3 & cols 2 to 3
ans =
     6     7
    10    11
```

행렬

- 행렬의 Dot Product : .*

```
>> A, B
A =
     5     7     9
     1    -3    -7
B =
    -1     2     5
     9     0     5
>> A.*B
ans =
    -5    14    45
     9     0   -35
>> A.*C
??? Error using ==> .*
Matrix dimensions must agree.
>> A.*C'
ans =
     0    21    36
     1     6   -14
```

행렬

- 행렬과 벡터의 연산

$$
\begin{aligned}
A\underline{x} &= \begin{bmatrix} 5 & 7 & 9 \\ 1 & -3 & -7 \end{bmatrix} \begin{bmatrix} 8 \\ -4 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 5\times 8 + 7\times(-4) + 9\times 1 \\ 1\times 8 + (-3)\times(-4) + (-7)\times 1 \end{bmatrix} \\
&= \begin{bmatrix} 21 \\ 13 \end{bmatrix}
\end{aligned}
$$

$(m\times \boxed{n}\text{ times }(\boxed{n}\times 1) \Rightarrow (m\times 1).$

```
>> A = [5 7 9; 1 -3 -7]
A =
     5     7     9
     1    -3    -7
>> x = [8; -4; 1]
x =
     8
    -4
     1
>> A*x
ans =
    21
    13
```

```
>> x*A
??? Error using ==> *
Inner matrix dimensions must agree.
```

행렬

- 행렬과 행렬의 연산

$$(m \times \boxed{n}) \text{ times } (\boxed{n} \times p) \Rightarrow (m \times p).$$

```
>> A = [5  7  9;  1 -3 -7]
A =
     5     7     9
     1    -3    -7
>> B = [0, 1; 3, -2; 4, 2]
B =
     0     1
     3    -2
     4     2
>> C = A*B
C =
    57     9
   -37    -7
```

```
>> D = B*A
D =
     1    -3    -7
    13    27    41
    22    22    22
>>  E = B'*A'
E =
    57   -37
     9    -7
```

행렬

- 역행렬 : *inv(행렬)*

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

A =

     1     2     3
     3     2     1
     4     4     4

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

B =

     1     2     3
     2     2     2
     1     2     4

>> C=inv(B)

C =

   -2.0000    1.0000    1.0000
    3.0000   -0.5000   -2.0000
   -1.0000         0    1.0000
```

```
>> A/B

ans =

     1     0     0
    -1     2     0
     0     2     0

>> A*C

ans =

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

Symbolic 기능

- 문자를 변수로 인식하여 계산하는 기능 : *syms*

```
>> syms x y
>> (x-y)*(x-y)^2

ans =

(x - y)^3

>> expand(ans)

ans =

x^3 - 3*x^2*y + 3*x*y^2 - y^3

>> factor(ans)

ans =

(x - y)^3
```

기본함수

- 반올림 : *round(값)*
- 0으로 부터 가장 가까운 정수로 올림 : *fix(값)*
- -∞로부터 가장 가까운 정수로 올림 : *floor(값)*
- ∞로부터 가장 가까운 정수로 올림 : *ceil(값)*
- 사인 : *sign(값)*

```
>> x=[-pi pi 3*pi -0.4 -0.6]

x =

   -3.1416    3.1416    9.4248   -0.4000   -0.6000

>> round(x)

ans =

    -3     3     9     0    -1

>> fix(x)

ans =

    -3     3     9     0     0
```

```
>> floor(x)

ans =

    -4     3     9    -1    -1

>> ceil(x)

ans =

    -3     4    10     0     0

>> sign(x)

ans =

    -1     1     1    -1    -1
```

기본함수

- 나머지 값 : *rem(x,y)*

```
>> rem(x,3)

ans =

   -0.1416    0.1416    0.4248   -0.4000   -0.6000
```

기본함수

- *sum(), trace()* 함수

```
>>  A = [1:3; 4:6; 7:9]
A =
     1     2     3
     4     5     6
     7     8     9
>> s = sum(A), ss = sum(sum(A))
s =
    12    15    18
ss =
    45

>> x = pi/4*(1:3)';
>> A = [sin(x), sin(2*x), sin(3*x)]/sqrt(2)
>> A =
    0.5000    0.7071    0.5000
    0.7071    0.0000   -0.7071
    0.5000   -0.7071    0.5000

>> s1 = sum(A.^2), s2 = sum(sum(A.^2))
s1 =
    1.0000    1.0000    1.0000
s2 =
    3.0000
```

```
>> A*A'

ans =

    1.0000         0         0
         0    1.0000    0.0000
         0    0.0000    1.0000

>> A'*A

ans =

    1.0000         0         0
         0    1.0000    0.0000
         0    0.0000    1.0000

>> trace(A)

ans =

    1.0000

>> trace(A*A')

ans =

    3.0000
```

기본함수

- *min(), max(), abs()* 함수

```
>> x = [1.3 -2.4 0 2.3], max(x), max(abs(x))
x =
    1.3000   -2.4000         0    2.3000
ans =
    2.3000
ans =
    2.4000
>> [m, j] = max(x)
m =
    2.3000
j =
     4
```

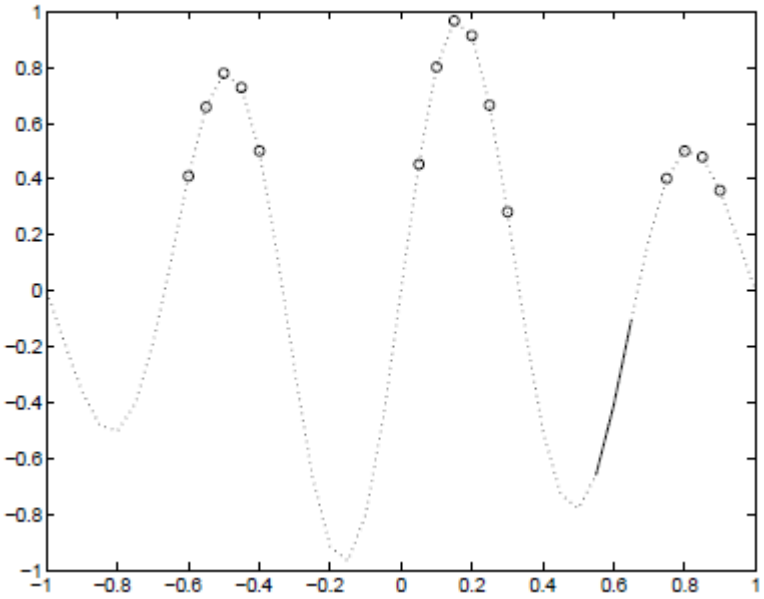
- (0과 1사이의)랜덤함수 : *rand()*

```
>> y =  rand, Y = rand(2,3)
y =
    0.9191
Y =
    0.6262    0.1575    0.2520
    0.7446    0.7764    0.6121
```

기본함수

- find(조건)

```
>> x = -1:.05:1;
>> y = sin(3*pi*x).*exp(-x.^2); plot(x,y,':')
>> k = find(y > 0.2)
 k =
  Columns 1 through 12
    9   10   11   12   13   22   23   24   25   26   27   36
  Columns 13 through 15
  37   38   39
>> hold on, plot(x(k),y(k),'o')
>> km = find( x>0.5 & y<0)
km =
    32    33    34
>> plot(x(km),y(km),'-')
```



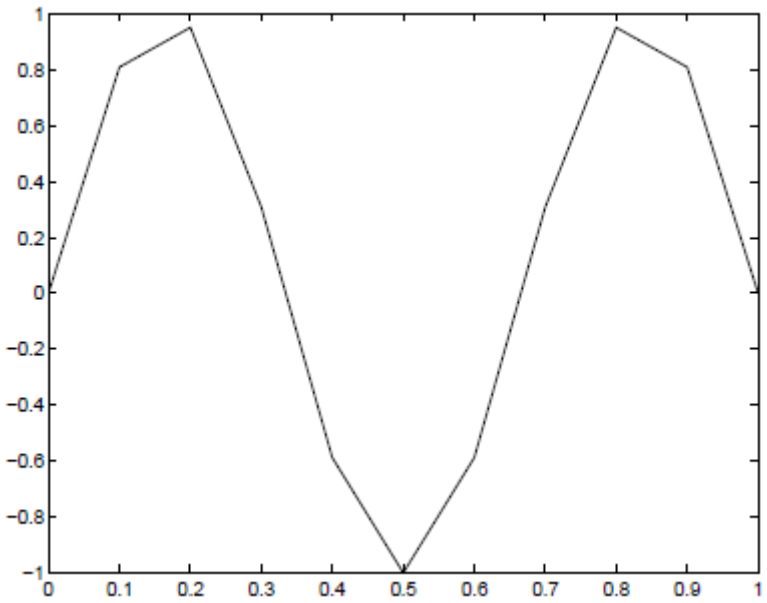
기본함수

```
>> A = [ -2 3 4 4; 0 5 -1 6; 6 8 0 1]
A =
    -2     3     4     4
     0     5    -1     6
     6     8     0     1
>> k = find(A==0)
k =
     2
     9
>> n  = find(A <= 0)
n =
     1
     2
     8
     9
>> A(n)
ans =
    -2
     0
    -1
     0
>> m = find( A' == 0)
m =
     5
    11
```

그림 생성

- *plot(x축 데이터,y축 데이터,'선종류')*

```
>> N = 10; h = 1/N; x = 0:h:1;
>> y = sin(3*pi*x);
>> plot(x,y)
```




```
>> N = 100; h = 1/N; x = 0:h:1;
>> y = sin(3*pi*x);  plot(x,y)
```

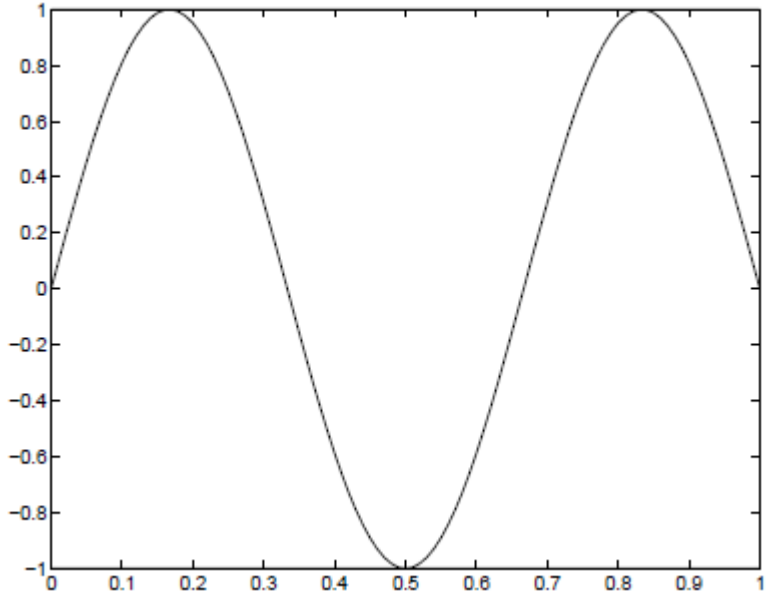


그림 생성

- 그림 제목 : *title('그림제목')*
- x축, y축 라벨 : *xlabel('x축 라벨')* *ylabel('y축 라벨')*
- 눈금 표시 : *grid / grid off*
- 선 종류

Colours		Line Styles	
y	yellow	.	point
m	magenta	o	circle
c	cyan	x	x-mark
r	red	+	plus
g	green	-	solid
b	blue	*	star
w	white	:	dotted
k	black	-.	dashdot
		--	dashed

- 여러그림 : *plot(x데이터1,y데이터1,'선종류',x데이터2,y데이터2,선종류)*
- 각 그림의 설명 : *legend('그림설명')*

그림 생성

```
>> N=100;
>> h=1/N;
>> x=0:h:1;
>> y=sin(3*pi*x);
>> z=cos(3*pi*x);
>> plot(x,y,'b-',x,z,'g--')
>> legend('Sin curve','Cos curve')
>> title('Multi-plot')
>> xlabel('x axis')
>> ylabel('y axis')
>> grid
>>
```

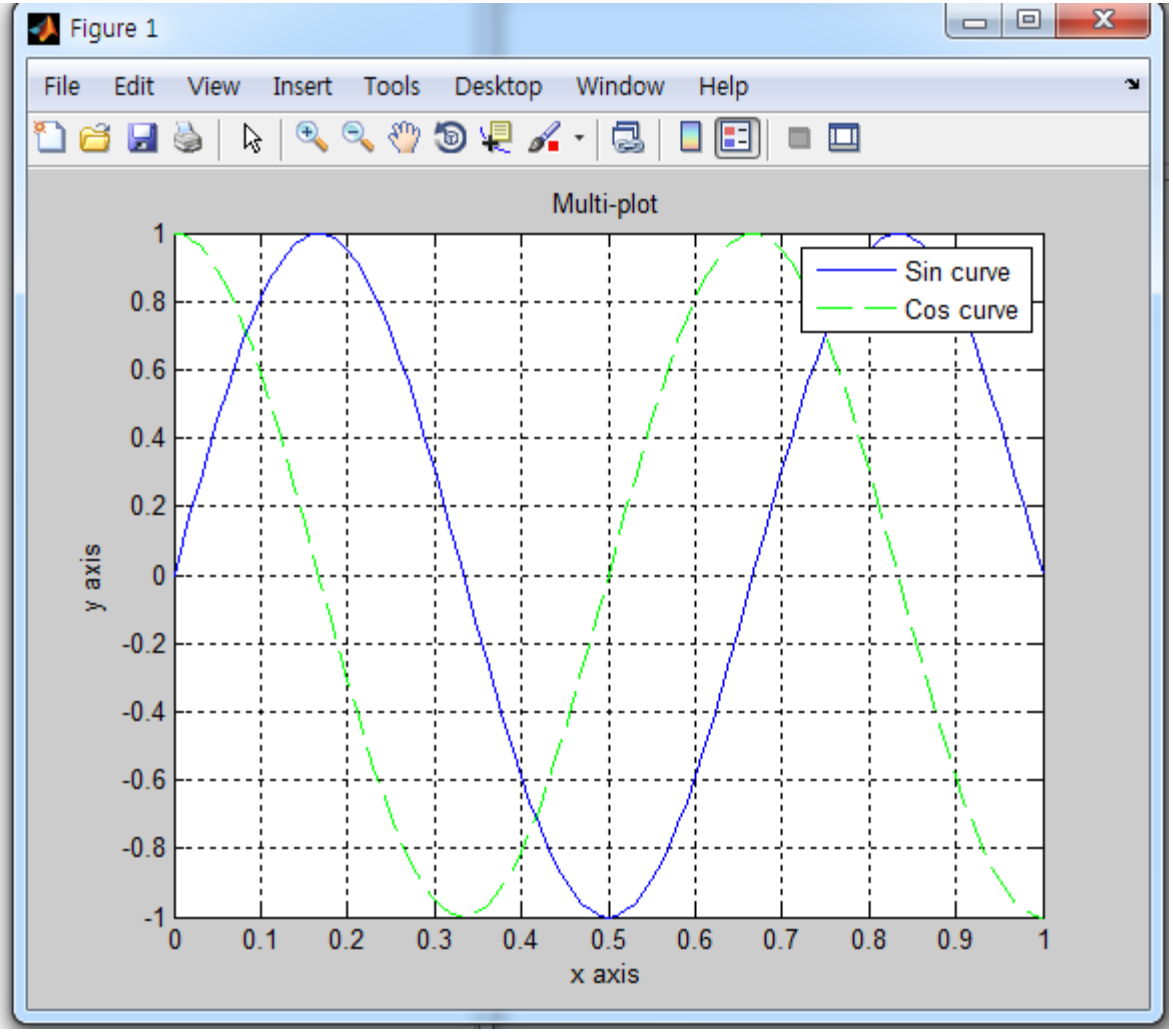


그림 생성

- 여러그림 : *hold on / hold off*

```
>> plot(x,y,'b-')
>> y=cos(3*pi*x);
>> hold on
>> plot(x,y,'g--')
>>
```

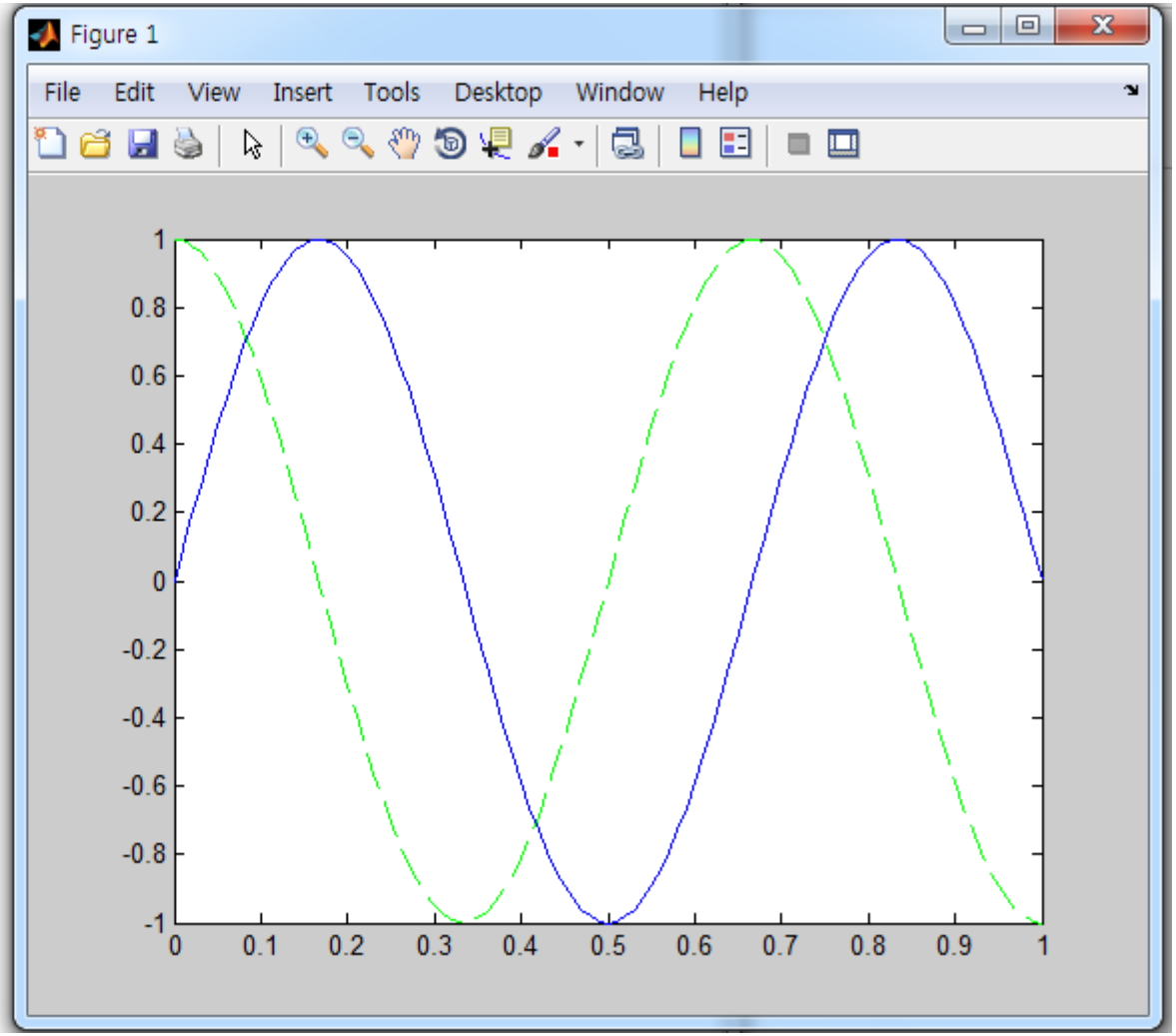


그림 생성

- 그림의 파일로의 출력 : *print –djpg 파일명*
- 여러 개의 그림을 한 그림에 그리기 : *subplot(행, 열, 번호)*

```
>> subplot(221), plot(x,y)
>>   xlabel('x'),ylabel('sin 3 pi x')
>> subplot(222), plot(x,cos(3*pi*x))
>>   xlabel('x'),ylabel('cos 3 pi x')
>> subplot(223), plot(x,sin(6*pi*x))
>>   xlabel('x'),ylabel('sin 6 pi x')
>> subplot(224), plot(x,cos(6*pi*x))
>>   xlabel('x'),ylabel('cos 6 pi x')
```

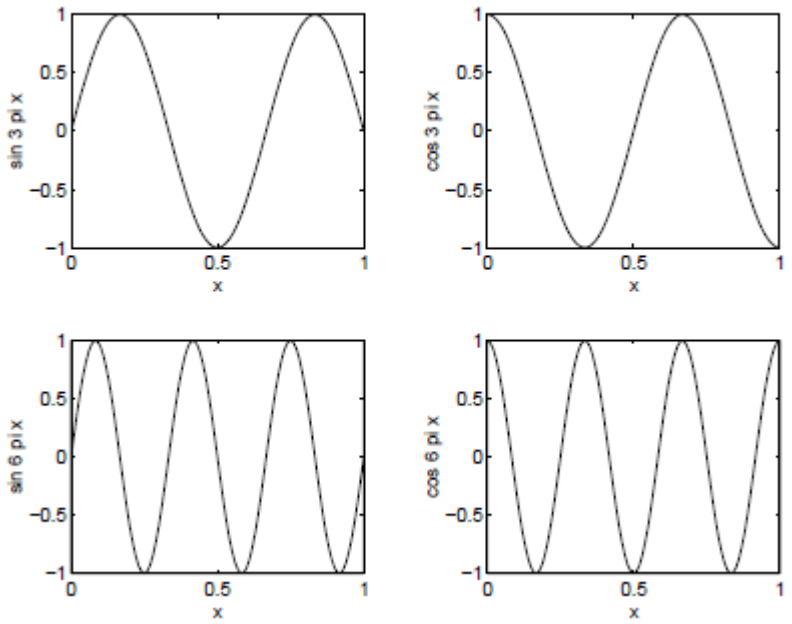


그림 생성

- 축 범위 설정 : *axis([x축 최소값, x축 최대값, y축 최소값,y축 최대값])*

```
>> clf
>> N=100;
>> h=1/N;
>> x=0:h:1;
>> y=sin(3*pi*x);
>> plot(x,y)
>> axis([-0.5,1.5,-1.2,1.2])
>> grid
>>
```

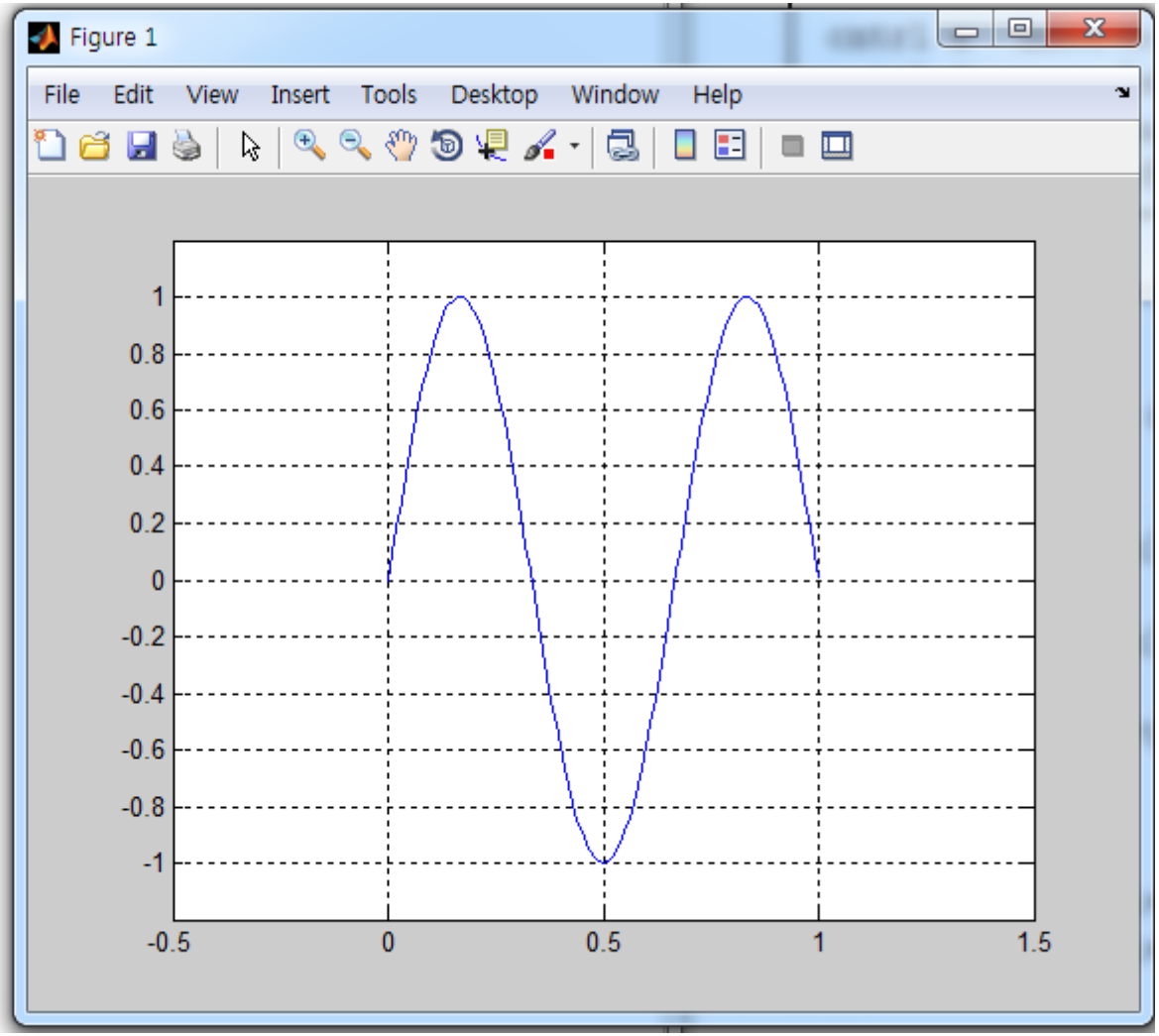
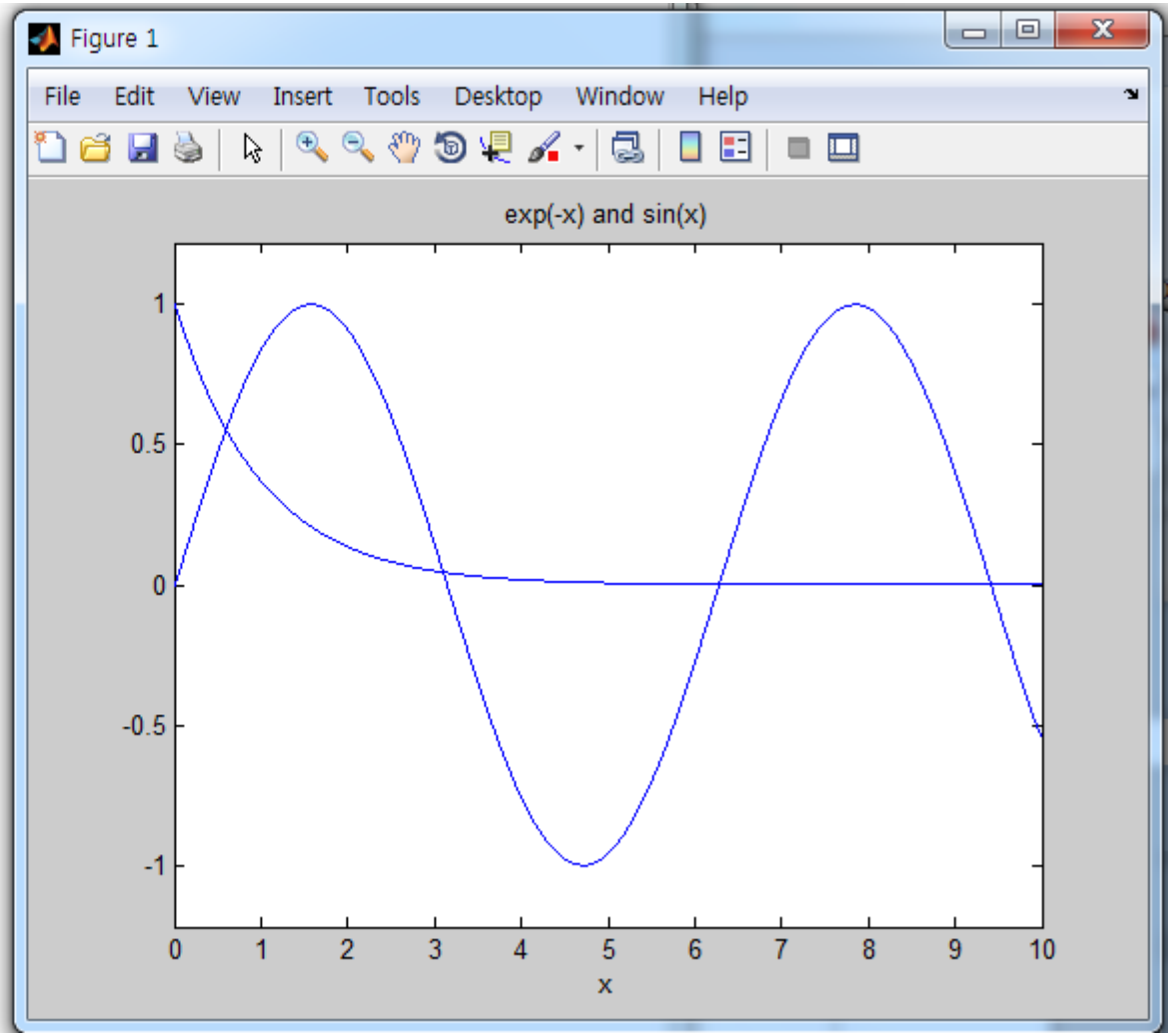


그림 생성

- *ezplot('y',[x 최소값, x최대값])*

```
>> ezplot('exp(-x)',[0,10])
>> hold on
>> ezplot('sin(x)',[0,10])
>> hold off
>> title('exp(-x) and sin(x)')
>> xlabel('x')
>>
```



그림생성

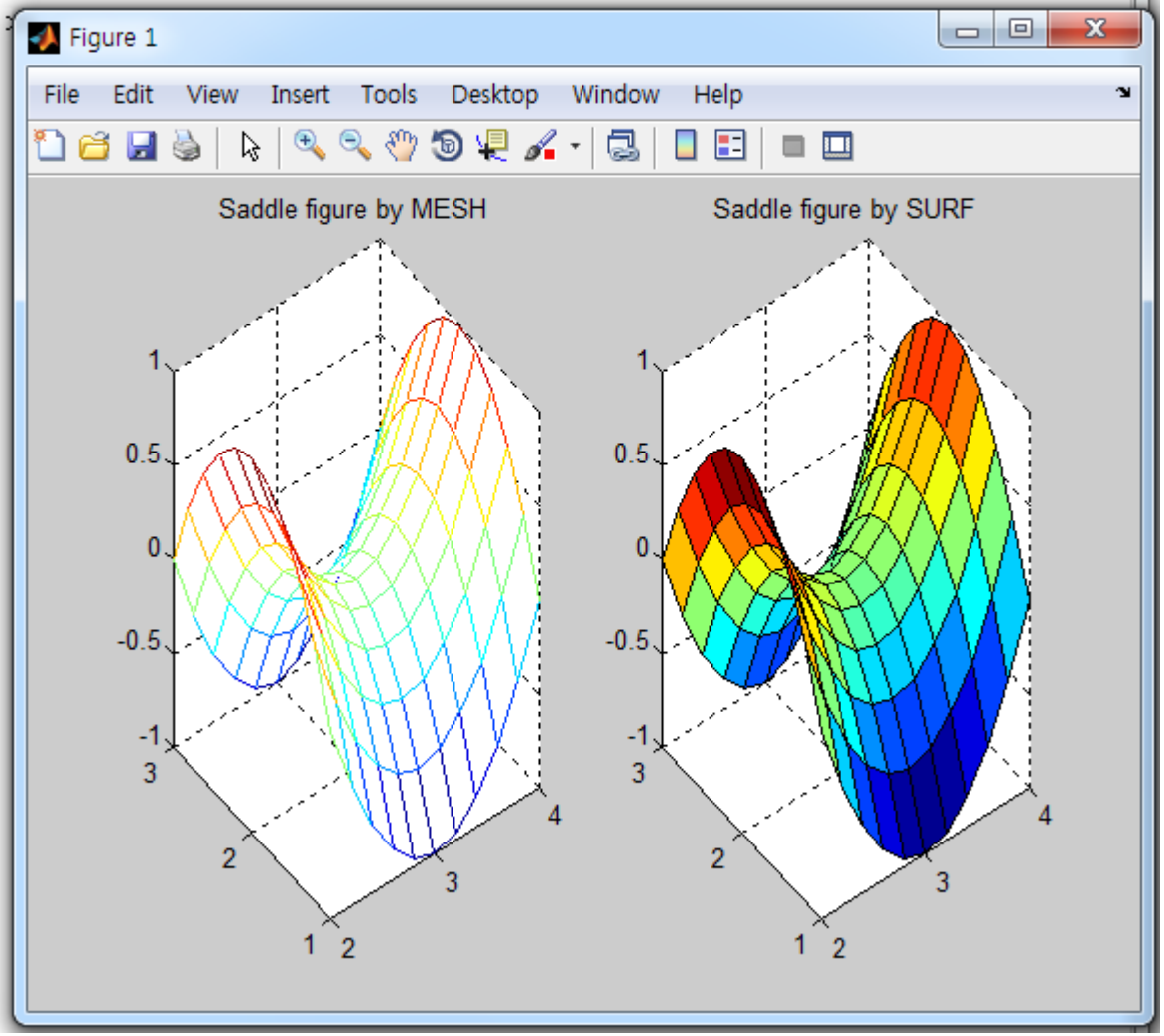
- 평면그림
  다음의 x,y 구간 동안

  $2 \le x \le 4 \ and \ 1 \le y \le 3$

  함수 f의 그래프를 그려라

  $f(x,y) = (x-3)^2 - (y-2)^2$

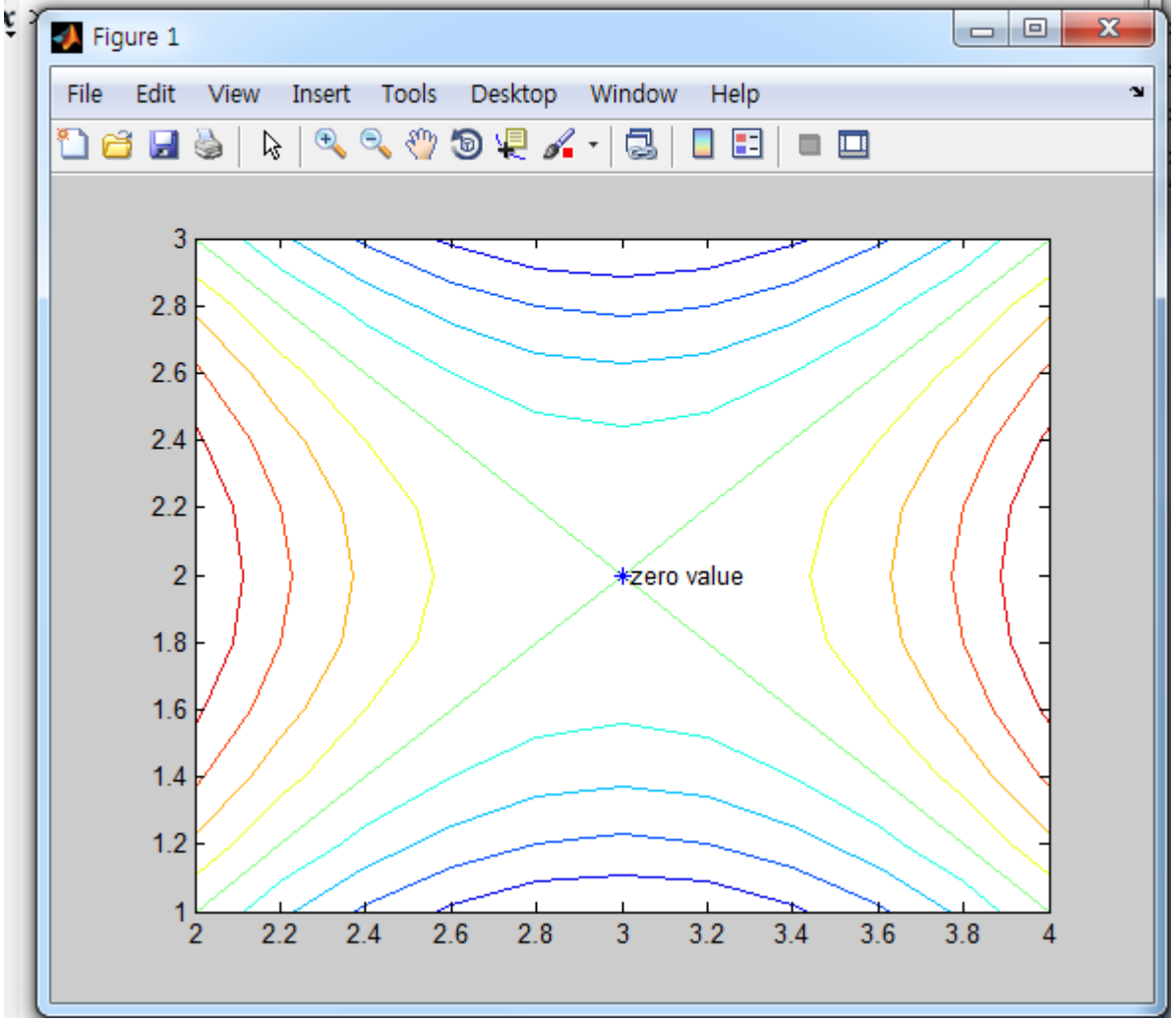
```
>> [x,y]=meshgrid(2:0.2:4, 1:0.2:3);
>> z=(x-3).^2-(y-2).^2;
>> subplot(121),mesh(x,y,z), title('Saddle figure by MESH')
>> subplot(122),surf(x,y,z), title('Saddle figure by SURF')
```



그림생성

- 등고선 : contour(x,y,z)

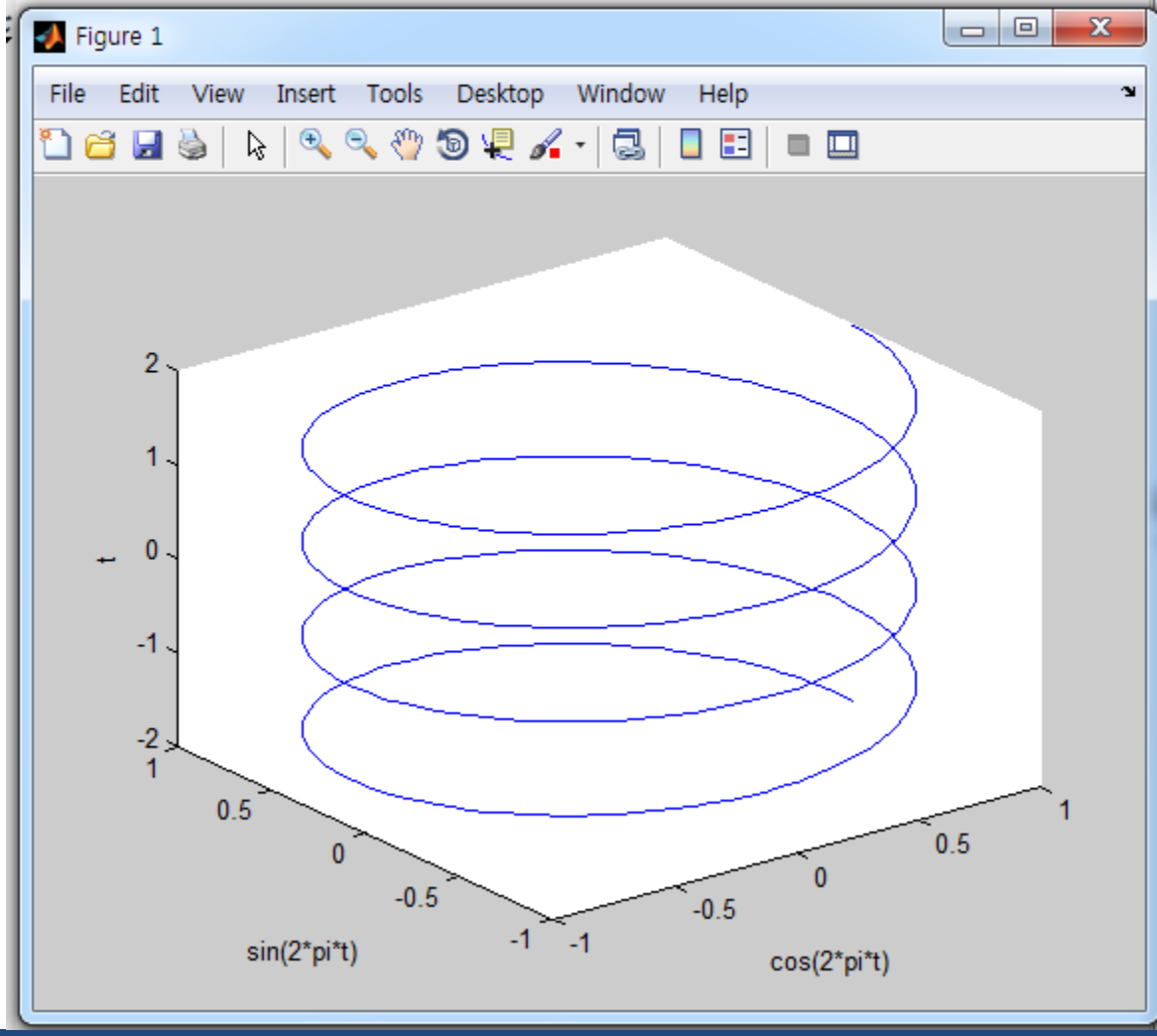
```
>> contour(x,y,z)
>> hold on
>> plot(x(61),y(61),'*')
>> text(x(61),y(61),' zero value')
```



그림생성

- 3차원 그래프 : plot3(x,y,z)

```
>> t=-2:0.01:2;
>> plot3(cos(2*pi*t),sin(2*pi*t),t)
>> xlabel('cos(2*pi*t)')
>> ylabel('sin(2*pi*t)')
>> zlabel('t')
```



반복문

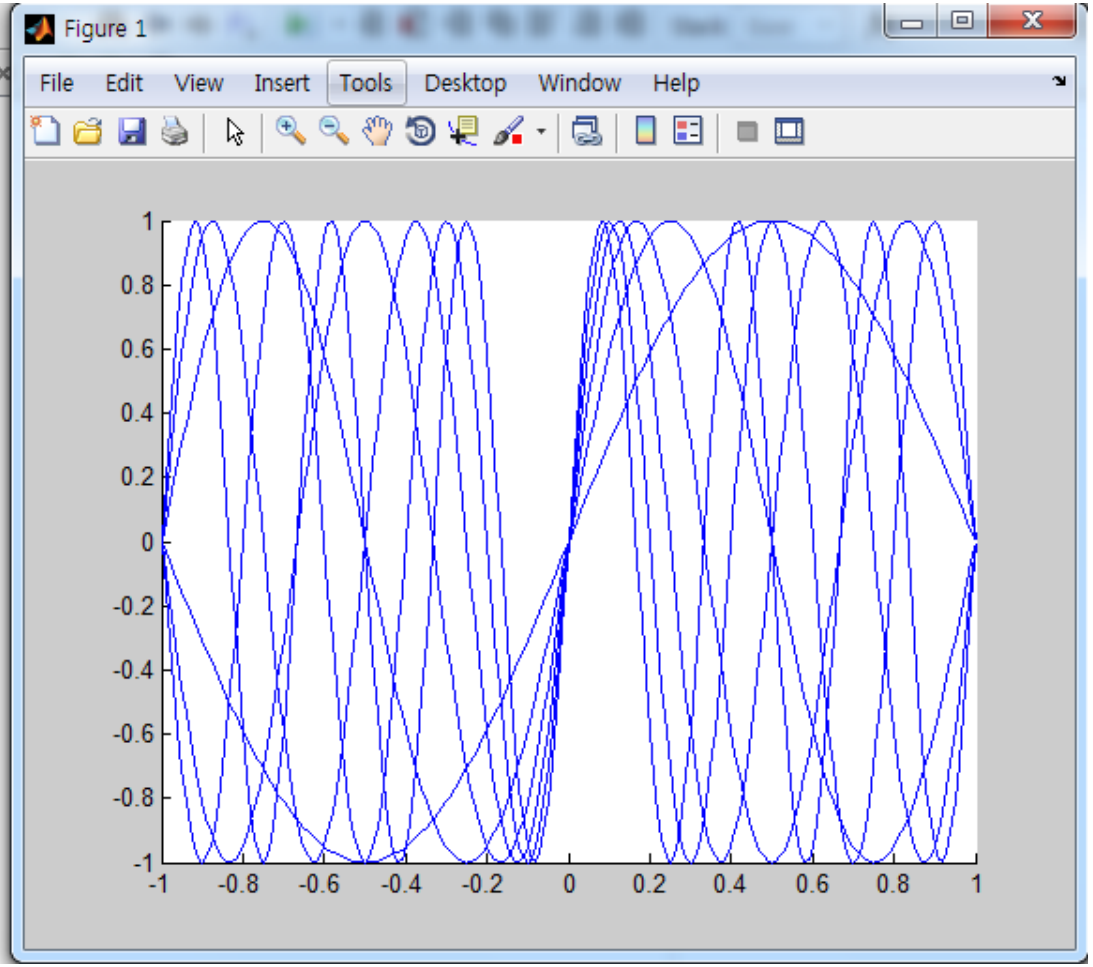
- for 문 : *for 변수= 변수범위*
  end
- 예제. $10! = 10 \times 9 \times \cdots \times 1$를 구하라.

```
>> f=1;
>> for n=2:10
f=f*n;
end
>> f

f =

     3628800
```

반복문

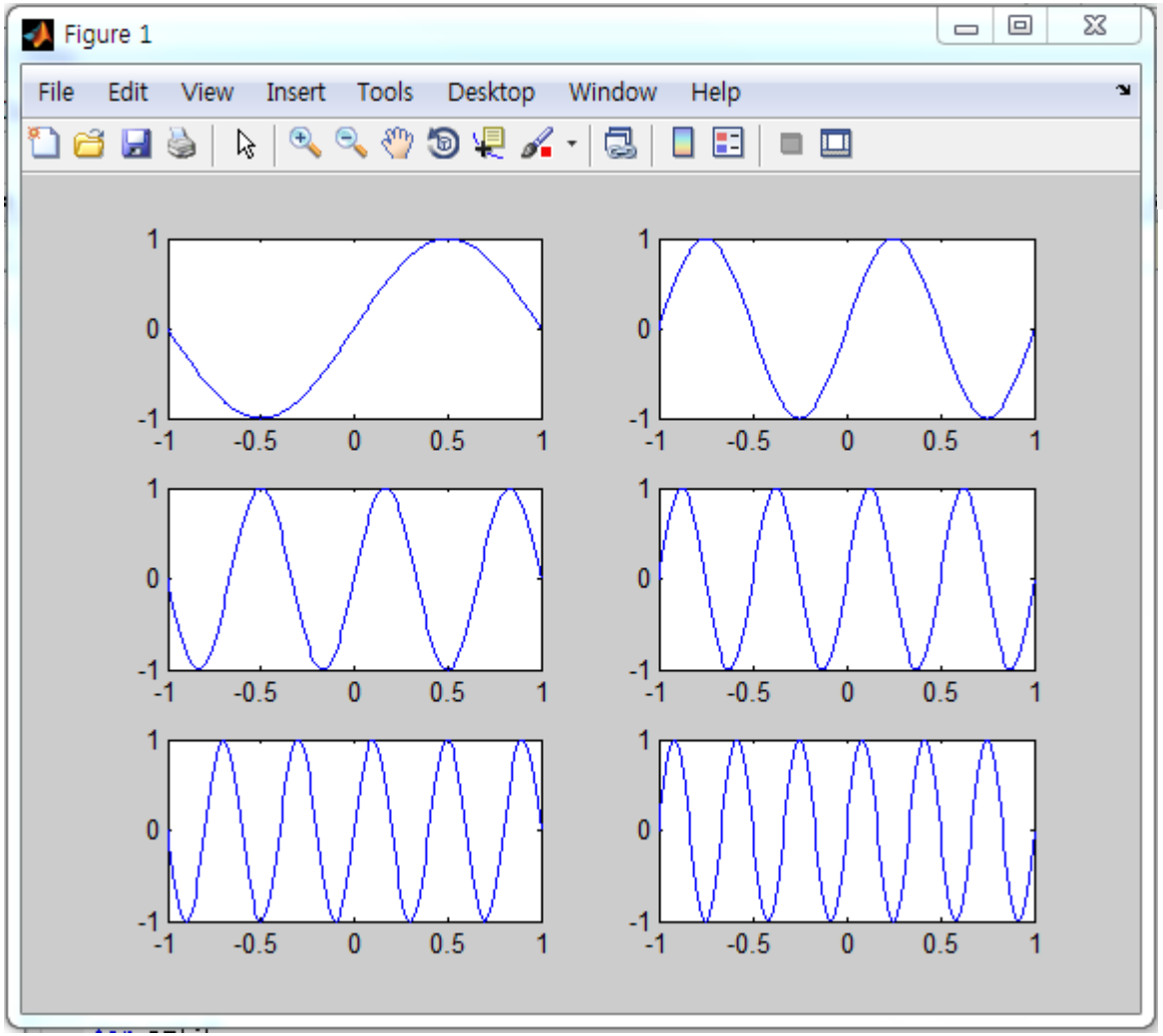
- 예제. $-1 \le x \le 1$ 구간에서의 $\sin(n\pi x)$의 그래프를 그려라. (n=1,2,…8)

```
% 한 그림에 그리기
x=-1:0.01:1;
figure;
hold on
for n=1:6
    plot(x,sin(n*pi*x))
end
hold off
```



반복문

- 연습. 다음 그래프를 그리시오.




```
% 여러 그림에 나누어 그리기
x=-1:0.01:1;
for n=1:6
    subplot(3,2,n), plot(x,sin(n*pi*x))
end
```

논리연산

- MATLAB에서는 참이면 1, 거짓이면 0의 값을 반영
- 연산기호

`x == 2`	is x equal to 2?
`x ~= 2`	is x **not** equal to 2?
`x > 2`	is x greater than 2?
`x < 2`	is x less than 2?
`x >= 2`	is x greater than or equal to 2?
`x <= 2`	is x less than or equal to 2?

논리연산

- 예

```
>> x = pi
x =
    3.1416
>> x ~= 3,   x ~= pi
ans =
     1
ans =
     0
```

```
x =
   -2.0000    3.1416    5.0000
   -1.0000         0    1.0000
>> x == 0
ans =
     0     0     0
     0     1     0
>> x > 1,   x >=-1
ans =
     0     1     1
     0     0     0
ans =
     0     1     1
     1     1     1
```

```
>> y = x>=-1, x > y
y =
     0     1     1
     1     1     1
ans =
     0     1     1
     0     0     0
```

논리연산

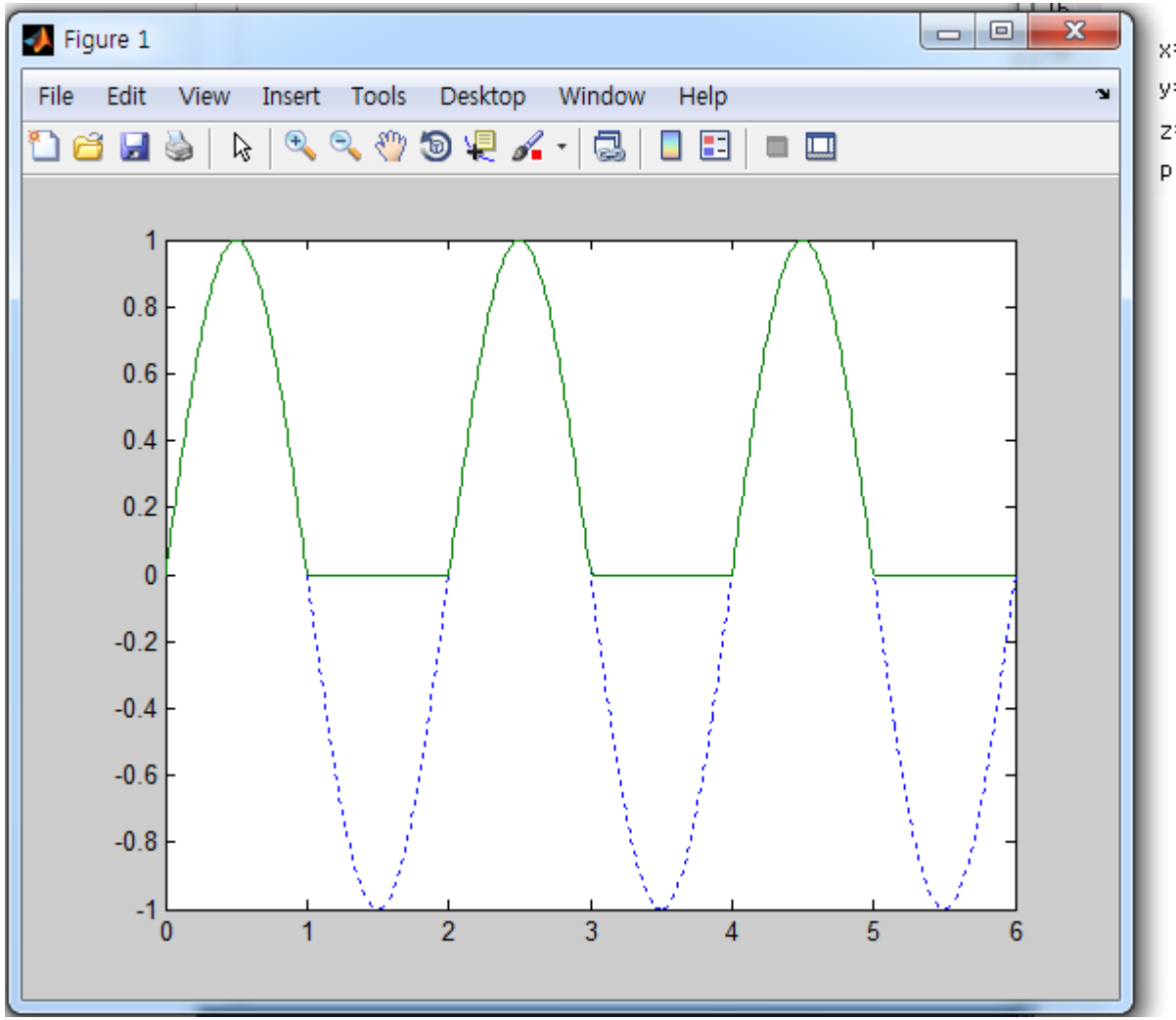
- AND : &, OR : |

```
x =
   -2.0000    3.1416    5.0000
   -5.0000   -3.0000   -1.0000
>> x > 3 & x < 4
ans =
     0     1     0
     0     0     0
>> x > 3 | x == -3
ans =
     0     1     1
     0     1     0

>> x > 3 | x == -3 | x <= -5
ans =
     0     1     1
     1     1     0
```

```
>> x, L =  x >= 0
x =
   -2.0000    3.1416    5.0000
   -5.0000   -3.0000   -1.0000
L =
     0     1     1
     0     1     1
>> pos = x.*L
pos =
         0    3.1416    5.0000
         0         0         0
```

논리연산

- 연습. 다음 그래프를 그리시오.




```
x=0:0.01:6;
y=sin(pi*x);
z=(y>=0).*y;
plot(x,y,':',x,z,'-')
```

논리반복문

- While 반복문
- 예제. 다음수식의 값이 100보다 작게 되는 최대의 n의 값은 얼마인가?

$$1^2 + 2^2 + \cdots + n^2$$

```
>> S = 1; n = 1;
>> while S+ (n+1)^2 < 100
    n = n+1;    S = S + n^2;
   end
>> [n, S]
ans =
     6    91
```

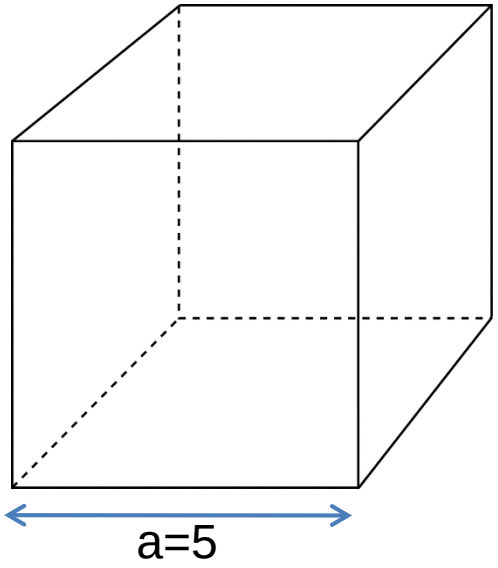
논리반복문

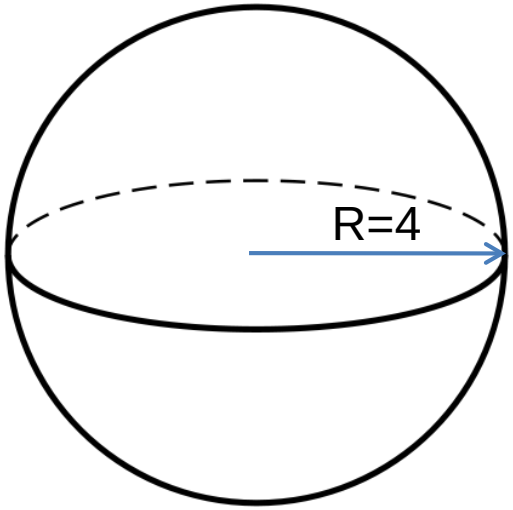
- if…else 문

```
>> a = 4;
b = 3;
if a<b
    ('b is bigger than a')
elseif a>b
    ('a is bigger than b')
else ('a and b are equal')
end

ans =

a is bigger than b
```

논리반복문

- 예제. 다음의 정육면체와 구의 부피를 비교하시오.






```
>> a=5;
R=4;
cube=a^3;
sphere=4/3*pi*R^3;
if(cube<sphere)
    ('sphere is bigger than cube')
else
    ('cube is bigger than sphere')
end

ans =

sphere is bigger than cube
```

$V = a^3$

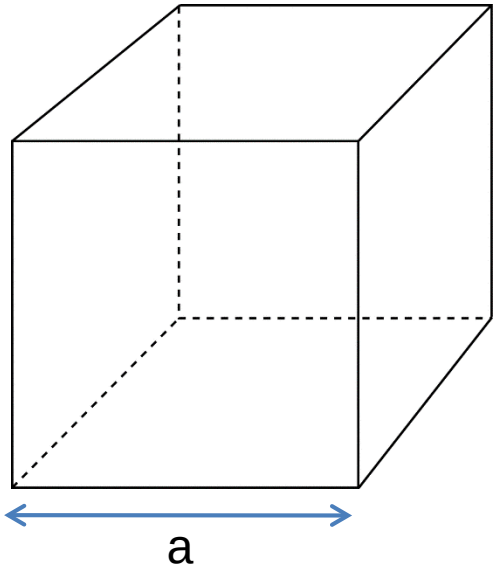
$R = a/2$

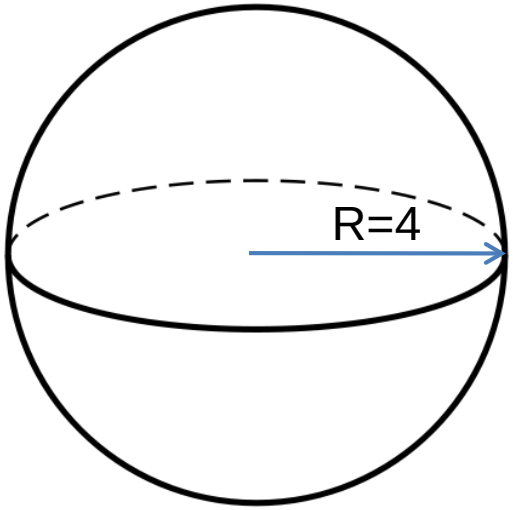
$a = 2R$

$V = 4/3\,\pi R^3$

논리반복문

- 연습. 다음의 정육면체의 부피가 구의 부피와 $0.0001m^3$ 오차 이내로 같아지는 a값을 구하시오.





$$V = a^3$$

$$R = a/2$$
$$a = 2R$$

$$V = \frac{4}{3}\pi R^3$$

```
>> R=4; sphere=4/3*pi*R^3;
a=1;
cube=1;
st=1;
while abs(sphere-cube)>0.0001
    cube=a^3;
    if sphere-cube>0
        a=a+st;
    else
        a=a-st; st=st/10; a=a+st; cube=a^3;
    end
end
sphere, cube, a

sphere =

  268.0826


cube =

  268.0825


a =

    6.4480
```

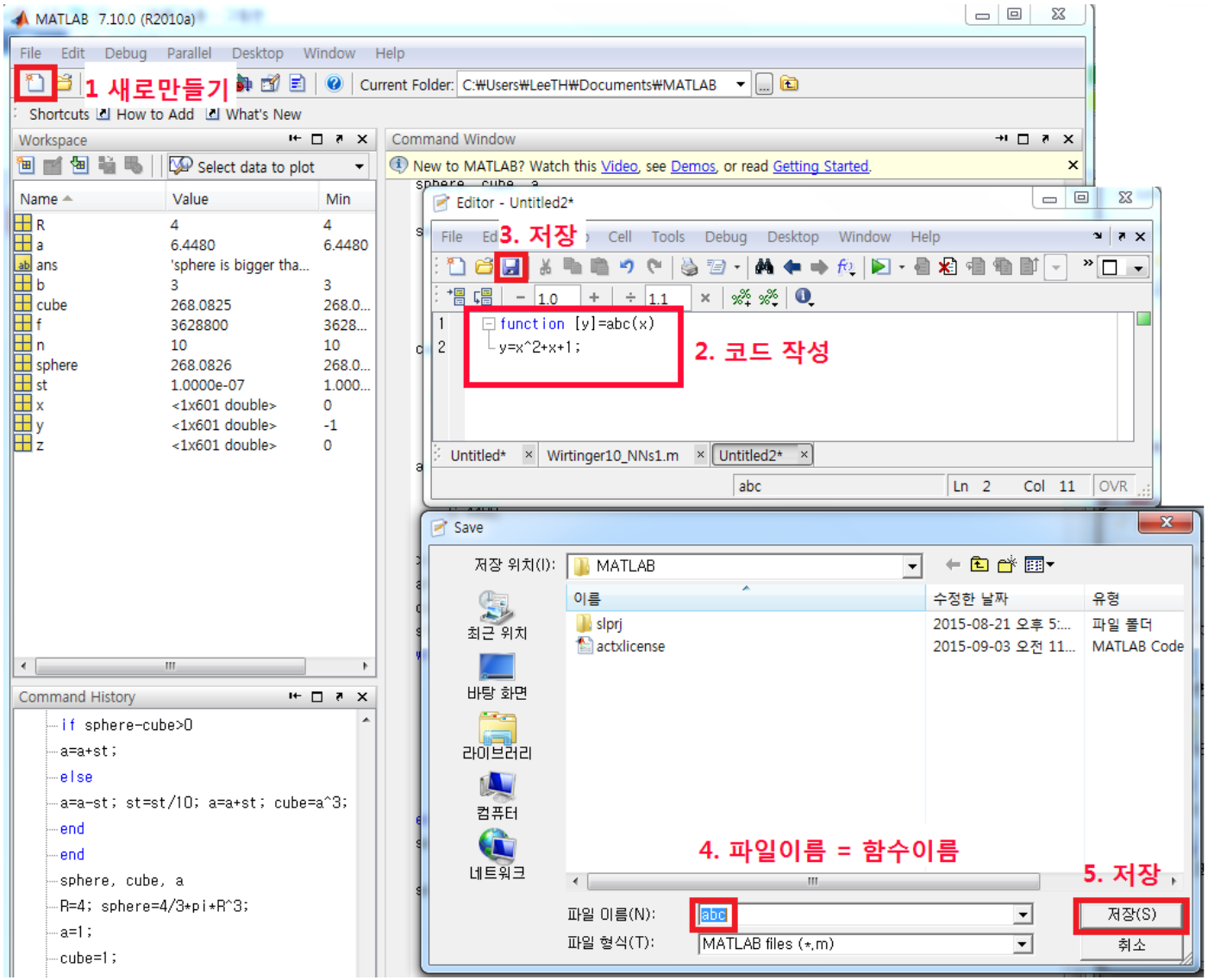
m파일 함수

- *inline('출력값','입력값')*

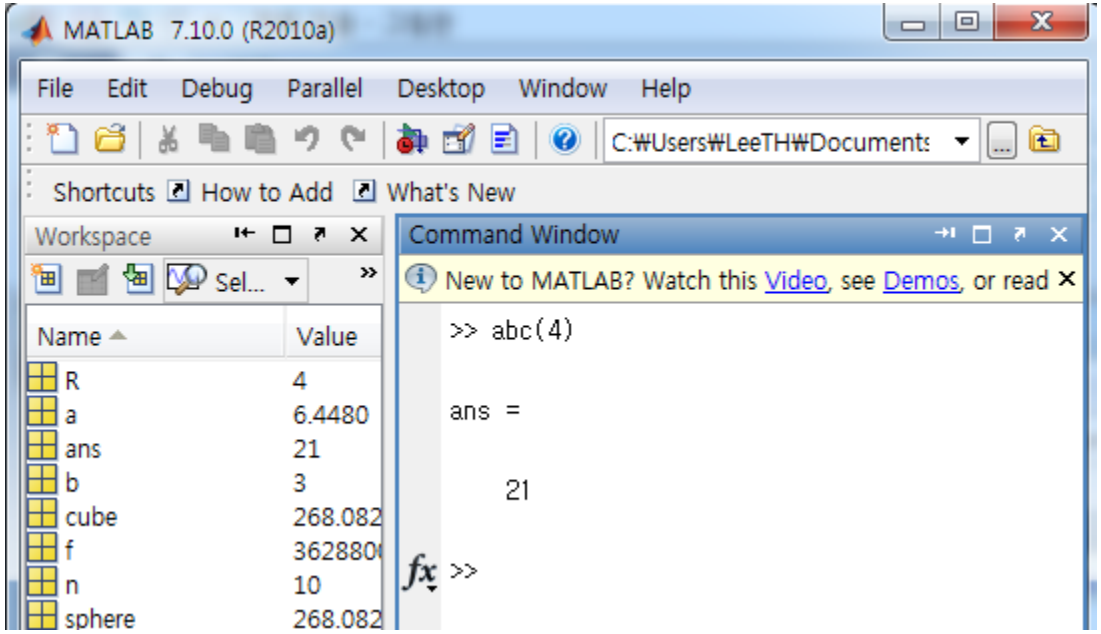
```
>> f = inline('x^2 + x + 1', 'x')

f =

     Inline function:
     f(x) = x^2 + x + 1

>> f(4)

ans =

    21
```

m파일 함수

- *function [출력]=함수명(입력,입력,입력)*



m파일 함수


MATLAB 7.10.0 (R2010a)
File Edit Debug Parallel Desktop Window Help
C:₩Users₩LeeTH₩Documents
Shortcuts How to Add What's New
Workspace
Sel...
Name Value
R 4
a 6.4480
ans 21
b 3
cube 268.082
f 362880
n 10
sphere 268.082
Command Window
New to MATLAB? Watch this Video, see Demos, or read
>> abc(4)
ans =
21
>>

m파일 함수

- 연습. 입력 a, b, c에 따라 다음의 값 $A = \sqrt{s(s-a)(s-b)(s-c)}$를 출력하는 함수 AREA를 만들어라. 여기서 $s = (a+b+c)/2$이다.