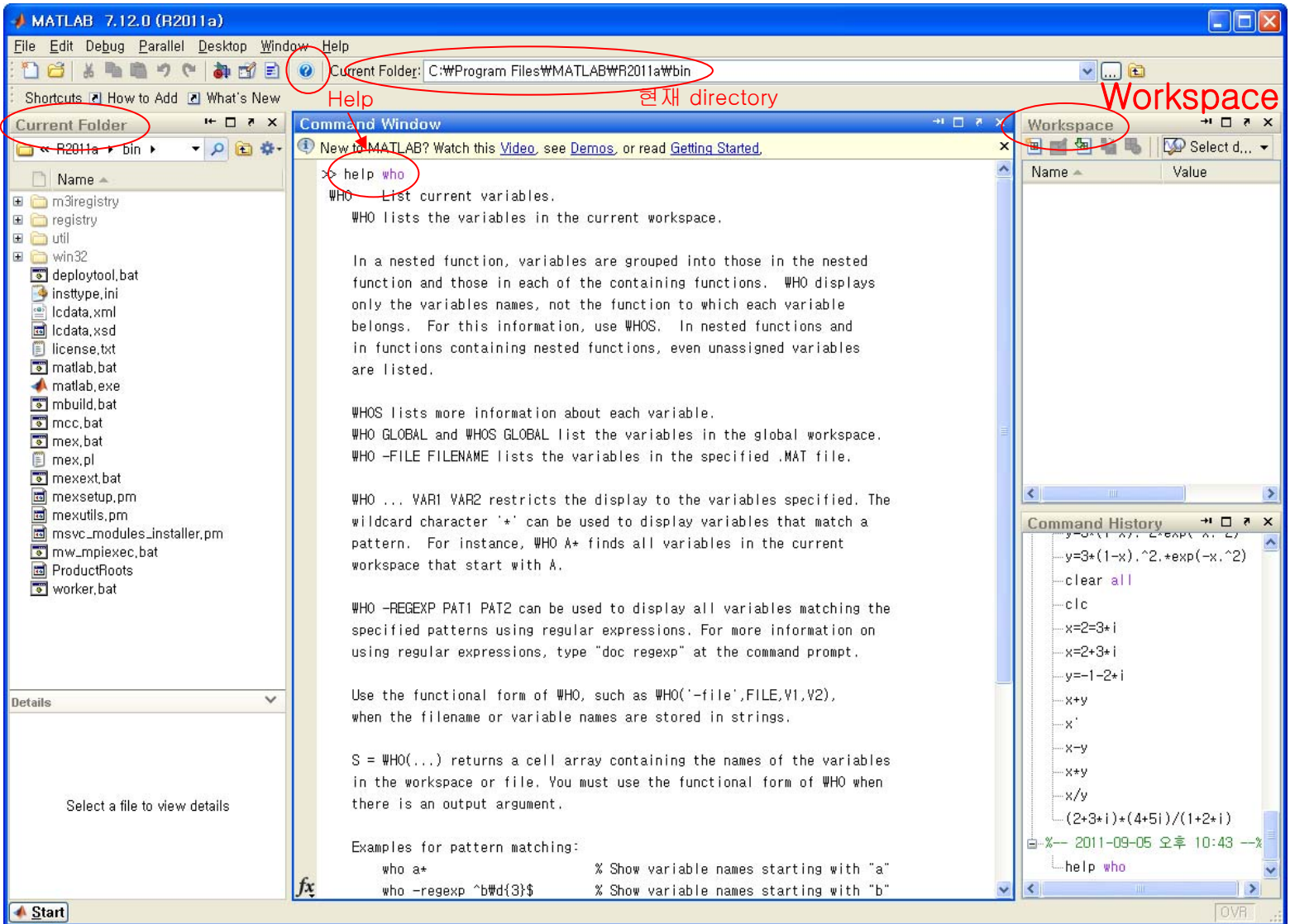


---

# MATLAB 기초 사용법

# 2.2. MATLAB의 작업환경



# 2.2. MATLAB의 작업환경

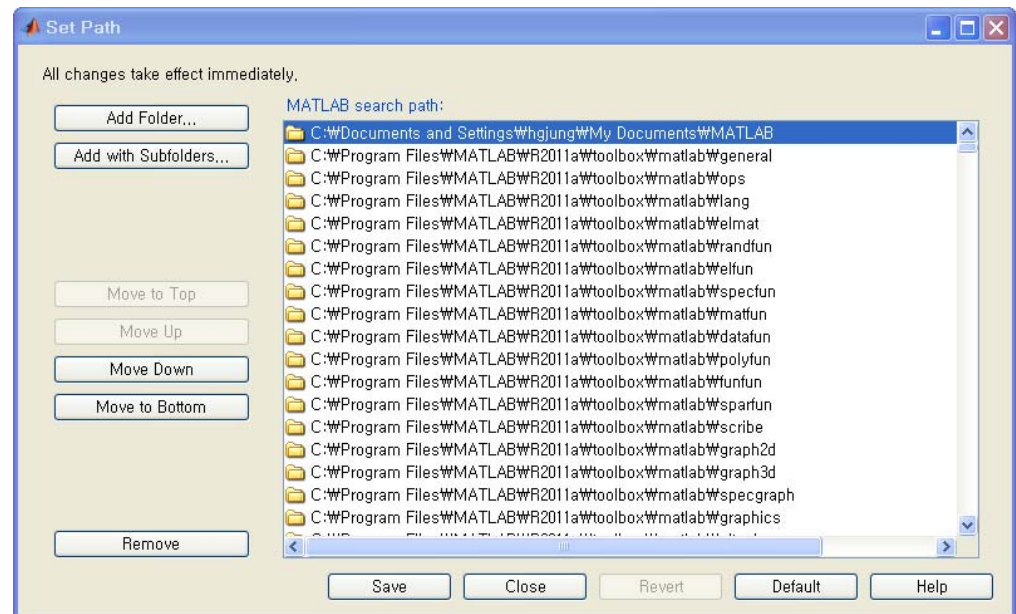
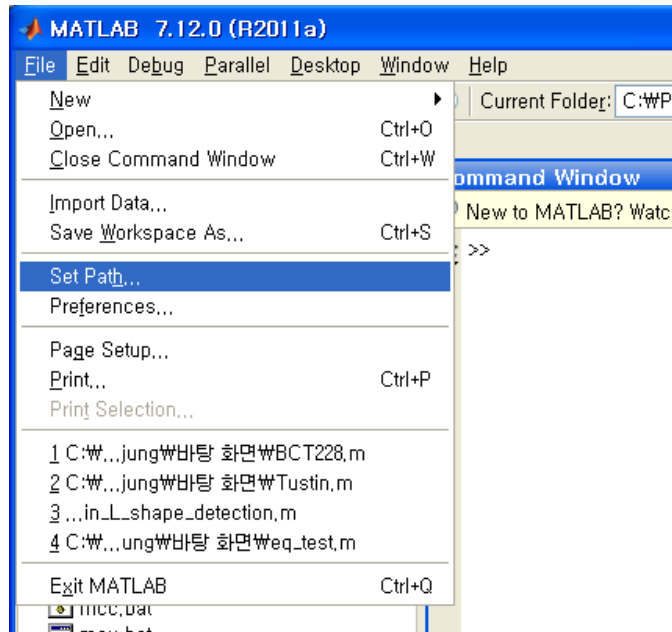
## 2.2.2 MATLAB의 작업폴더

>> cd



>> dir

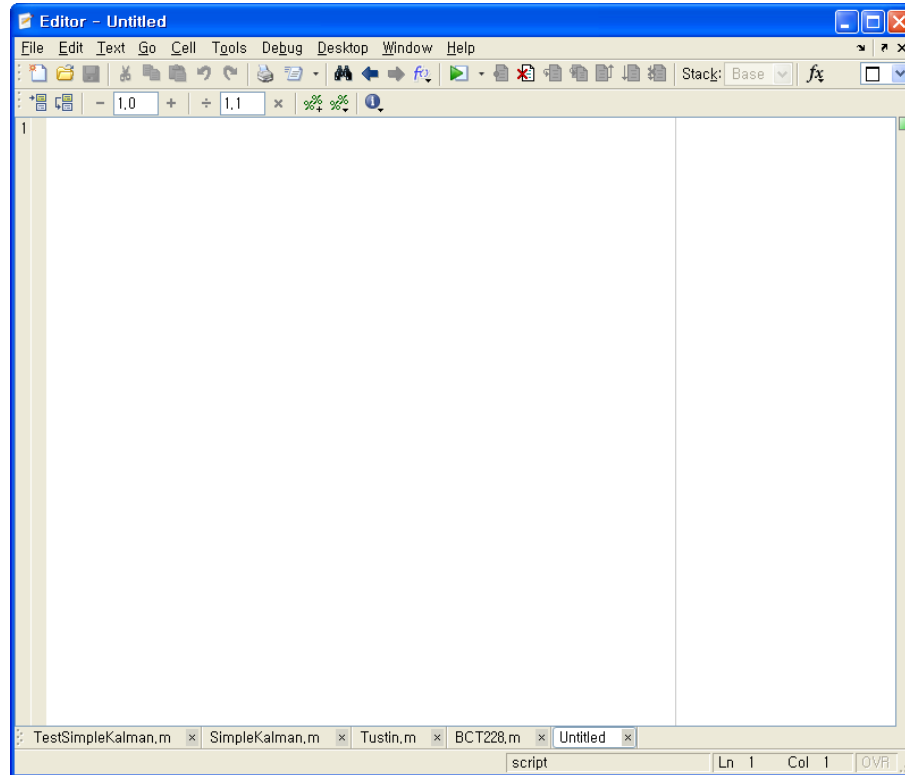
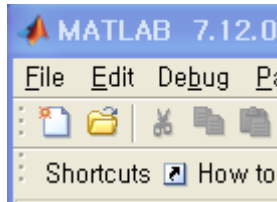
\* Path: MATLAB 프로그램이 파일을 찾는 경로



## 2.2. MATLAB의 작업환경

### 2.2.4. MATLAB의 작업방법

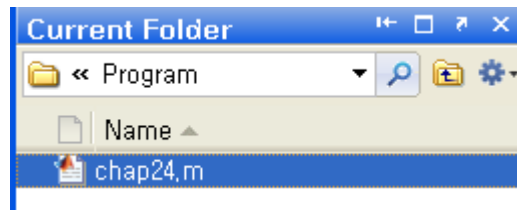
- 1) MATLAB에서 실행되는 파일인 m 파일을 만들어 실행하는 방법
- 2) 매번 MATLAB 명령어창의 “>>” 프롬프트에서 실행하는 방법



## 2.2. MATLAB의 작업환경

### 2.2.4. MATLAB의 작업방법

chap24.m

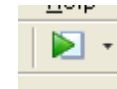


```
t=[1:20];
```

```
y=sin(t);
```

```
plot(t,y);
```

```
grid on;
```



## 2.2. MATLAB의 파일들

- 1) **“.m” 파일**: MATLAB의 함수나 수식을 사용하여 프로그램하는 매크로 파일
- 2) **“.mat”, “.dat”, “.txt” 파일**: MATLAB 작업 상에서 사용한 변수들의 값을 저장할 때 생기는 파일. 디스크에 저장된 데이터 파일들은 다시 MATLAB으로 불러와서 사용할 수 있다.

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Sta
>> A=[1 2 3 4; 5 6 7 8]

A =

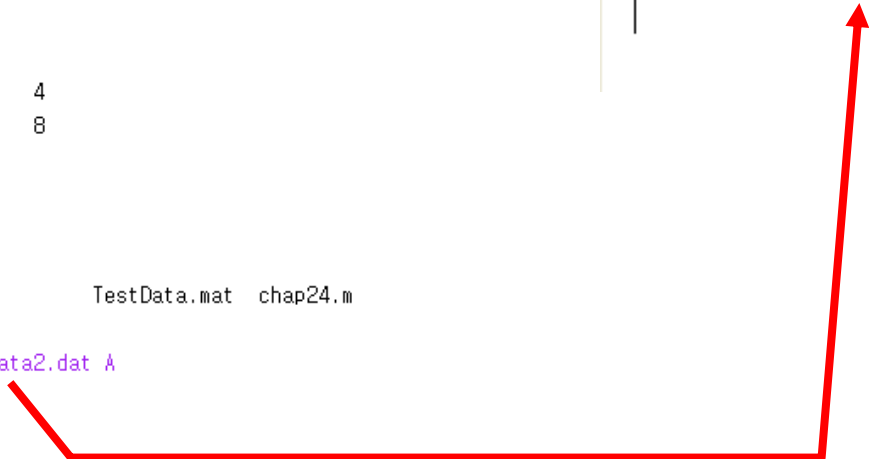
     1     2     3     4
     5     6     7     8

>> save TestData A
>> dir

.          ..          TestData.mat  chap24.m

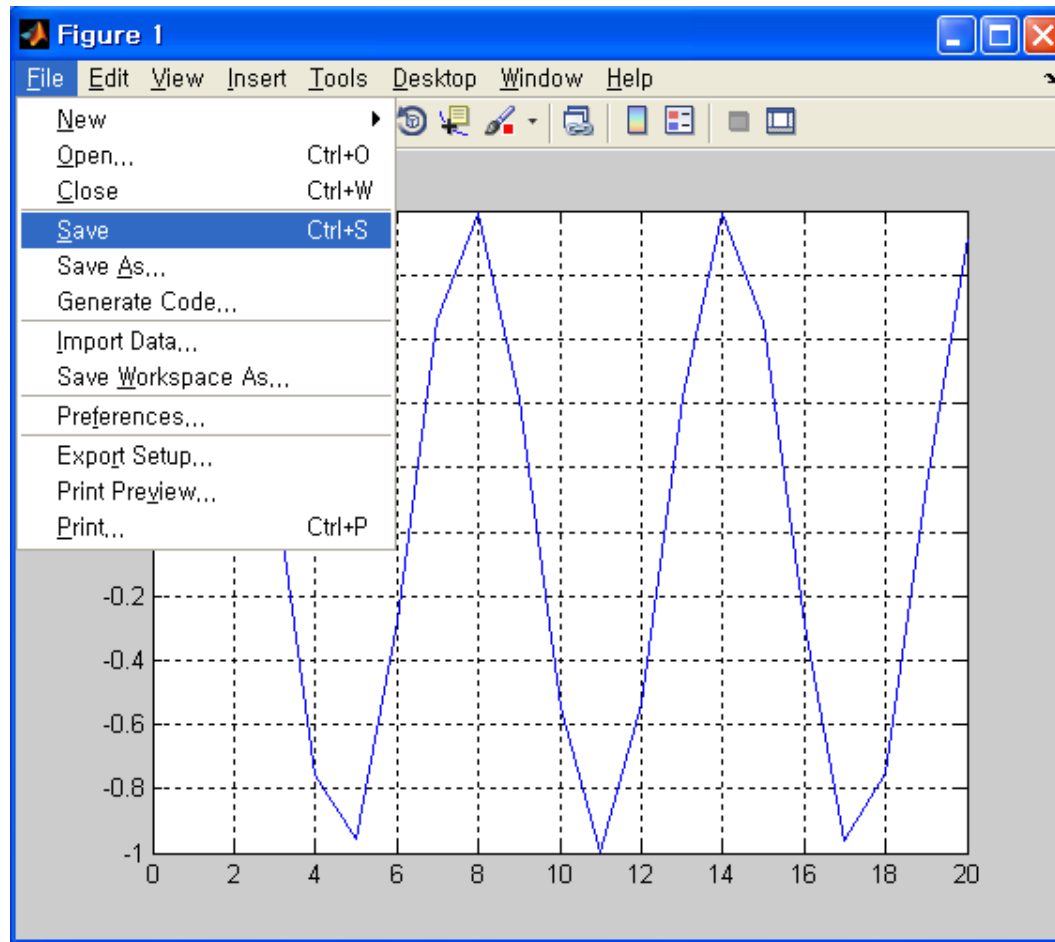
>> save -ascii TestData2.dat A
fx >>
```

```
1.0000000e+000  2.0000000e+000  3.0000000e+000  4.0000000e+000
5.0000000e+000  6.0000000e+000  7.0000000e+000  8.0000000e+000
```



## 2.2. MATLAB의 파일들

3) “.fig” 파일: MATLAB에서 그려진 그래프를 파일로 저장할 때 생기는 파일.



## 2.3. MATLAB에서 쓰이는 기호들

---

### 2.3.1. 일반기호

[1 2 3]' → 전치 (transpose)

'text' → 문자열

[ ] → 벡터, 행렬, 다항식

( ) → 벡터나 행렬, 다항식의 원소

% → 주석

; → 행의 끝을 나타내거나, 수식 끝에서 사용 시에는 화면에 출력을 방지한다.

: → 수열을 만든다.

. → 벡터나 행의 원소끼리 계산할 때 쓰인다.



## 2.3. MATLAB에서 쓰이는 기호들

---

### 2.3.1. 일반기호

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

```
A =
```

```
1 2
3 4
```

```
>> A'
```

```
ans =
```

```
1 3
2 4
```

```
>> t=[0:0.1:0.9]
```

```
t =
```

```
Columns 1 through 9
```

```
0 0.1000 0.2000 0.3000 0.4000 0.5000 0.6000 0.7000 0.8000
```

```
Column 10
```

```
0.9000
```

```
>> A(1,2)
```

```
ans =
```

```
2
```

## 2.3. MATLAB에서 쓰이는 기호들

---

### 2.3.2. 수식기호

+	.+ → 원소끼리 더하기
-	.- → 원소끼리 빼기
*	.* → 원소끼리 곱하기
/	./ → 원소끼리 나누기
^ → 거듭제곱	.^ → 각 원소의 거듭제곱

## 2.3. MATLAB에서 쓰이는 기호들

---

### 2.3.2. 수식기호

[예제 2.1] 벡터나 행렬의 계산과 원소끼리의 계산

```
>> a=[1 2 3]
a =
     1     2     3
>> b=[4 5 6]
b =
     4     5     6
>> c=a+b'
c =
    32
>> c=a.+b
c =
     4    10    18
```

## 2.3. MATLAB에서 쓰이는 기호들

---

### 2.3.3. 관계/논리 연산자

<

>

<=

>=

== 같다

~= 같지 않다

& and

| or

~ not

## 2.3. MATLAB에서 쓰이는 기호들

---

### 2.3.4. MATLAB의 예약변수

i 또는 j 허수

inf 무한대

pi  $\pi$

NaN Not-a-Number

```
>> sqrt(-1)
```

```
ans =
```

```
0 + 1.0000i
```

```
>> pi
```

```
ans =
```

```
3.1416
```

```
>> pi>inf
```

```
ans =
```

```
0
```

```
>> pi<inf
```

```
ans =
```

```
1
```

```
>> 0/0
```

```
ans =
```

```
NaN
```

```
>> inf+1
```

```
ans =
```

```
Inf
```

```
>> inf-1
```

```
ans =
```

```
Inf
```

## 2.4. 함수의 표현

---

### MATLAB의 수학함수

abs(x)

acos(x)

asin(x)

atan(x)

Atan2(y,x)

conj(x)

cos(x)

exp(x)

imag(x)

log(x)

log10(x)

real(x)

sqrt(x)

tan(x)

## 2.4. 함수의 표현

### MATLAB의 수학함수

[예제 2.3] 공액보소수와 전치행렬의 차이점

```
>> A=[1+2*j 3-4*j; 5+6*j 7-8*j]
```

```
A =
```

```
1.0000 + 2.0000i    3.0000 - 4.0000i  
5.0000 + 6.0000i    7.0000 - 8.0000i
```

```
>> conj(A)
```

```
ans =
```

```
1.0000 - 2.0000i    3.0000 + 4.0000i  
5.0000 - 6.0000i    7.0000 + 8.0000i
```

```
>> A'
```

```
ans =
```

```
1.0000 - 2.0000i    5.0000 - 6.0000i  
3.0000 + 4.0000i    7.0000 + 8.0000i
```

In mathematics, the conjugate transpose, Hermitian transpose, Hermitian conjugate, or adjoint matrix of  $\mathbf{A}$  is  $\mathbf{A}^*$  obtained from  $\mathbf{A}$  by taking the transpose and then taking the complex conjugate.

[http://en.wikipedia.org/wiki/Conjugate\\_transpose](http://en.wikipedia.org/wiki/Conjugate_transpose)

## 2.4. 함수의 표현

---

[예제 2.4] MATLAB을 사용한 수식계산의 예

```
>> x=2;
```

```
>> y=3+(1-x)^2+exp(-x^2)
```

```
y =
```

```
0.0549
```

```
>> x=[0:0.1:0.5];
```

```
>> y=3+(1-x)^2+exp(-x^2)
```

```
??? Error using ==> mpower
```

```
Inputs must be a scalar and a square matrix.
```

```
To compute elementwise POWER, use POWER (.^) instead.
```

```
>> y=3+(1-x).^2+exp(-x.^2)
```

```
y =
```

```
3.0000    2.4058    1.8447    1.3435    0.9203    0.5841
```



## 2.4. 함수의 표현

다음의 수식을 MATLAB상에서 표현해보고,  $x=1+2i$ 일 경우와  $x=[1\ 2\ 3]$ 일 경우에 각각의 값을 계산해라.

$$y = 0.5e^{-3x^2} + \sqrt{\sin \frac{x}{2}} + \cos \frac{\pi}{2}$$

```
>> x=1+2+j
x =
    1.0000 + 2.0000i
>> y=0.5*exp(-3*x^2)+sqrt(sin(x/2))+cos(pi/2)
y =
    3.4199e+003 + 2.1745e+003i
>> real(y)
ans =
    3.4199e+003
>> imag(y)
ans =
    2.1745e+003
```

## 2.5. 벡터와 행렬의 표현 및 계산

### 2.5.1. 벡터의 입력

>> 변수 = 지정 값

숫자 사이 공백은 “,”와 같은 역할을 한다. 열 벡터 만드는 법.

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

```
>> b=[1 2 3]'
b =
     1
     2
     3
>> b=[1;2;3]
b =
     1
     2
     3
```

## 2.5. 벡터와 행렬의 표현 및 계산

---

### 2.5.2. 등간격 벡터 만들기

>> 벡터 = [초기 값: 증가율: 마지막 값]

```
>> t=[0:1:10]
```

```
t =
```

```
0 1 2 3 4 5 6 7 8 9 10
```

```
>> t=[0:0.1:0.5]
```

```
t =
```

```
0 0.1000 0.2000 0.3000 0.4000 0.5000
```

## 2.5. 벡터와 행렬의 표현 및 계산

---

### 2.5.2. 등간격 벡터 만들기

>> 벡터 = linspace(초기 값, 마지막 값, 원소의 개수)

```
>> t=linspace(0,1,11)
```

```
t =
```

```
Columns 1 through 8
```

```
0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000    0.7000
```

```
Columns 9 through 11
```

```
0.8000    0.9000    1.0000
```

## 2.5. 벡터와 행렬의 표현 및 계산

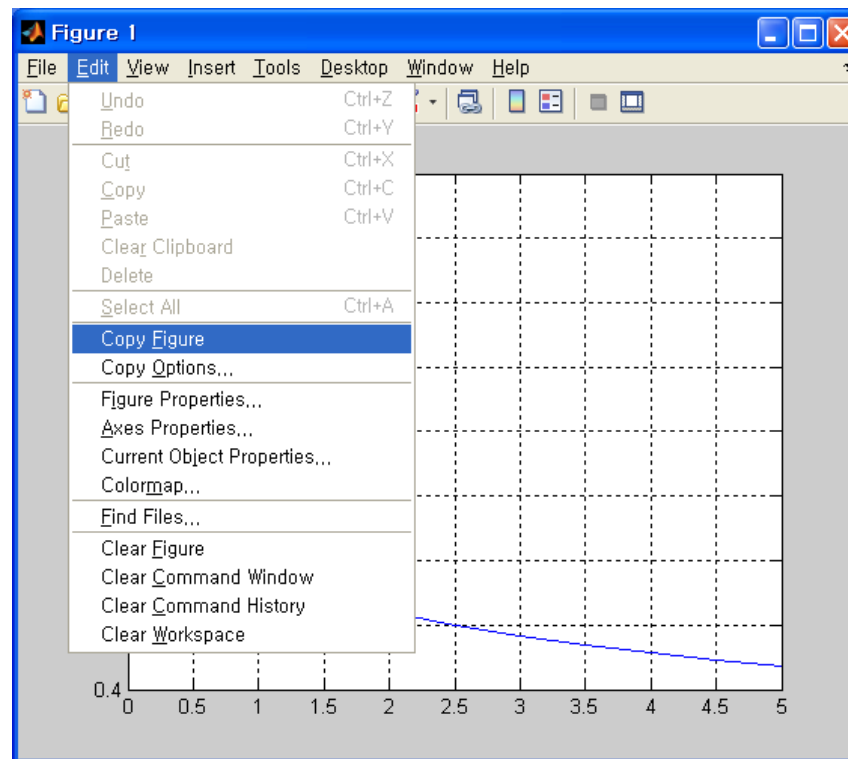
### 2.5.2. 등간격 벡터 만들기

[예제 2.7] 함수의 계산

시간  $t$ 가 0초부터 5초까지 0.1초씩 증가할 때, 다음의 함수  $y$ 를 구하고 그래프를 그려보자.

$$y = e^{-2t} + e^{-\frac{1}{3}\sqrt{t}}$$

```
>> t=[0:0.5:5];  
>> y=exp(-2*t)+exp(-1/3*sqrt(t));  
>> plot(t,y);  
>> grid on;
```



## 2.5. 벡터와 행렬의 표현 및 계산

---

### 2.5.3. 벡터의 출력

벡터의 원소 값을 출력할 때는 괄호 “( )”를 사용한다.

```
>> a=[1 2 3 4]
```

```
a =
```

```
    1    2    3    4
```

```
>> a(2:3)
```

```
ans =
```

```
    2    3
```

```
>> a(2:3)=[10 20]
```

```
a =
```

```
    1   10   20    4
```

## 2.5. 벡터와 행렬의 표현 및 계산

### 2.5.6. 벡터나 행렬을 만드는 function들

ones(m,n)

```
>> eye(3)
```

```
ans =
```

```
1 0 0
0 1 0
0 0 1
```

zeros(m,n)

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

```
v =
```

```
1 2 3
```

eye(m)

```
>> diag(v)
```

```
ans =
```

```
1 0 0
0 2 0
0 0 3
```

diag(v)

inv(A)

eig(A)

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

```
A =
```

```
1 2
3 4
```

```
>> det(A)
```

```
ans =
```

```
-2
```

```
>> inv(A)
```

```
ans =
```

```
-2.0000 1.0000
1.5000 -0.5000
```

```
>> eig(A)
```

```
ans =
```

```
-0.3723
5.3723
```

## 2.5. 벡터와 행렬의 표현 및 계산

### 2.5.7. 행렬의 출력

```
A(r,c)          >> A=[1 2 3; 4 5 6; 7 8 9]          >> A(6)
A(r,:)          A =                                ans =
A(:,c)          1     2     3                                8
A(r1:r2,c1:c2) 4     5     6
A(r,c1:c2)      7     8     9          >> A(5)
A(r1:r2,c)      >> A(1:2,2:3)          ans =
A(:,c1:c2)      ans =                                5
A(r1:r2,:)      2     3
A(i)            5     6
```

MATLAB에서는 열 벡터가 기본형

행렬 원소를 일련의 숫자(index)로 지정 가능



## 2.5. 벡터와 행렬의 표현 및 계산

### 2.5.8. 행렬의 대수 연산

행렬의 거듭제곱과 행렬 원소의 거듭제곱

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

```
A =
```

```
1 2
3 4
```

```
>> A^2
```

```
ans =
```

```
7 10
15 22
```

```
>> A.^2
```

```
ans =
```

```
1 4
9 16
```

```
>> A=[1 2 3; 4 5 6]
```

```
A =
```

```
1 2 3
4 5 6
```

```
>> B=[1 2 3;11 12 13]
```

```
B =
```

```
1 2 3
11 12 13
```

```
>> A+B
```

```
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

```
>> A+B'
```

```
ans =
```

```
14 74
32 182
```

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

```
ans =
```

```
1 4 9
44 60 78
```

## 2.5. 벡터와 행렬의 표현 및 계산

### 2.5.8. 행렬의 대수 연산

행렬의 나눗셈에는 오른쪽 나누기와 왼쪽 나누기의 두 가지가 있다.

$$Ax=b \rightarrow x=A \setminus b : \text{왼쪽 나누기} \rightarrow x=A^{-1}b$$

$$xA=b \rightarrow x=b/A : \text{오른쪽 나누기} \rightarrow x=bA^{-1}$$

$$b/A=(A' \setminus b)'$$

```
>> A=[3 4 2;6 0 -1; -5 -2 1]
```

```
A =
```

```
     3     4     2
     6     0    -1
    -5    -2     1
```

```
>> b=[19 4 -9]'
```

```
b =
```

```
    19
     4
    -9
```

```
>> x=A\b
```

```
x =
```

```
    1.0000
    3.0000
    2.0000
```

```
>> x=inv(A)*b
```

```
x =
```

```
    1.0000
    3.0000
    2.0000
```

## 2.5. 벡터와 행렬의 표현 및 계산

### 2.5.8. 행렬의 대수 연산

[연습문제 2.11] 다음 연립방정식을 푸시오.

$$\begin{cases} 5x - 3y - 3z = -1 \\ 3x + 2y - 2z = -10 \\ 2x - y + 2z = 8 \end{cases}$$

```
>> A=[5 -3 -3;3 3 -2;2 -1 2]
```

```
A =
```

```
    5    -3    -3
     3     3    -2
     2    -1     2
```

```
>> B=[-1;-10;8]
```

```
B =
```

```
    -1
   -10
     8
```

```
>> X=A\B
```

```
X =
```

```
    0.3377
   -1.8442
    2.7403
```

```
>> A*X
```

```
ans =
```

```
    -1
   -10
     8
```

## 2.6. MATLAB의 기본 명령어들

---

### 2.6.1. 벡터와 행렬의 크기를 알려주는 명령어

“length” : 행과 열의 크기 중에서 큰 값이 출력 → 벡터의 길이 구할 때

“size” : 행과 열의 크기가 출력

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

```
A =
```

```
    1    2    3
```

```
>> length(A)
```

```
ans =
```

```
    3
```

```
>> size(A)
```

```
ans =
```

```
    1    3
```

## 2.6. MATLAB의 기본 명령어들

---

### 2.6.2. 변수상태 점검명령어, who와 whos

```
>> who
```

Your variables are:

```
A    B    X    a    ans  b    t    v    x    y
```

```
>> whos
```

Name	Size	Bytes	Class	Attributes
A	1x3	24	double	
B	3x1	24	double	
X	3x1	24	double	
a	1x4	32	double	
ans	1x2	16	double	
b	3x1	24	double	
t	1x11	88	double	
v	1x3	24	double	
x	3x1	24	double	
y	1x11	88	double	

## 2.6. MATLAB의 기본 명령어들

---

### 2.6.3. 지움 명령어, clear

```
>> who
```

```
Your variables are:
```

```
A    B    X    a    ans  b    t    v    x    y
```

```
>> clear t
```

```
>> who
```

```
Your variables are:
```

```
A    B    X    a    ans  b    v    x    y
```

## 2.6. MATLAB의 기본 명령어들

---

### 2.6.4-5. 저장명령어 `save`, 데이터 입력명령어 `load`

>> `save filename` → 모든 변수를 저장

>> `save filename A` → 변수 A만 지정 파일에 저장

```
>> A=1
```

```
A =
```

```
1
```

```
>> B=2
```

```
B =
```

```
2
```

```
>> save testdata A B
```

```
>> clear all
```

```
>> load testdata
```

```
>> who
```

```
Your variables are:
```

```
A B
```

## 2.6. MATLAB의 기본 명령어들

---

### 2.6.8. 화면출력 명령어, disp

```
>> disp('Hello, World!')
Hello, World!
>> n=123

n =

    123

>> disp(n)
    123

>> disp(['n=' num2str(n)])
n=123
>> disp(['n=' num2str(n, '%04d')])
n=0123
```



# 2.7. 그래프 그리기

---

## 2.7.1. 그래프를 그리는 명령어, plot

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

```
>> t=[0:0.1:2*pi];
```

```
>> y=sin(t);
```

```
>> plot(t,y); grid on;
```

```
>> whos t y
```

Name	Size	Bytes	Class	Attributes
t	1x63	504	double	
y	1x63	504	double	

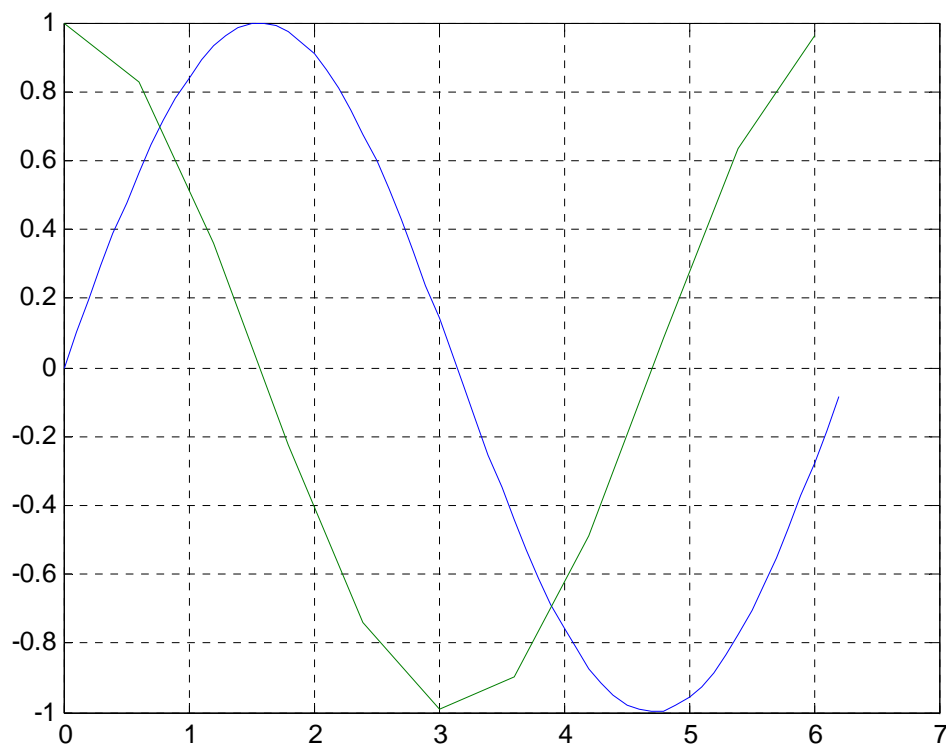
## 2.7. 그래프 그리기

### 2.7.1. 그래프를 그리는 명령어, plot

여러 그래프를 한꺼번에 그릴 수 있다. 이때, 벡터 쌍의 크기는 모두 같아야 한다.

```
>> plot(t1,y1,t2,y2,t3,y3,t4,y4)
```

```
>> t1=[0:0.1:2*pi];  
>> y1=sin(t1);  
>> t2=[0:0.6:2*pi];  
>> y2=cos(t2);  
>> plot(t1,y1,t2,y2); grid on;
```



## 2.7. 그래프 그리기

---

### 2.7.2. 로그그래프를 그리는 명령어, **semilogx**, **semilogy**, **loglog**

figure : 새로운 그래프 창을 띄운다.

plot(a,b)

loglog(a,b)

semilogx(a,b) : a벡터를 상용로그인 x축으로, b벡터를 y축으로 그래프를 그린다.

semilogy(a,b) : a벡터를 x축으로, b벡터를 상용로그인 y축으로 그래프를 그린다.

subplot(m,n,p) : 그래프를 m행, n열로 나눈 다음, p번째 그래프를 활성화한다.

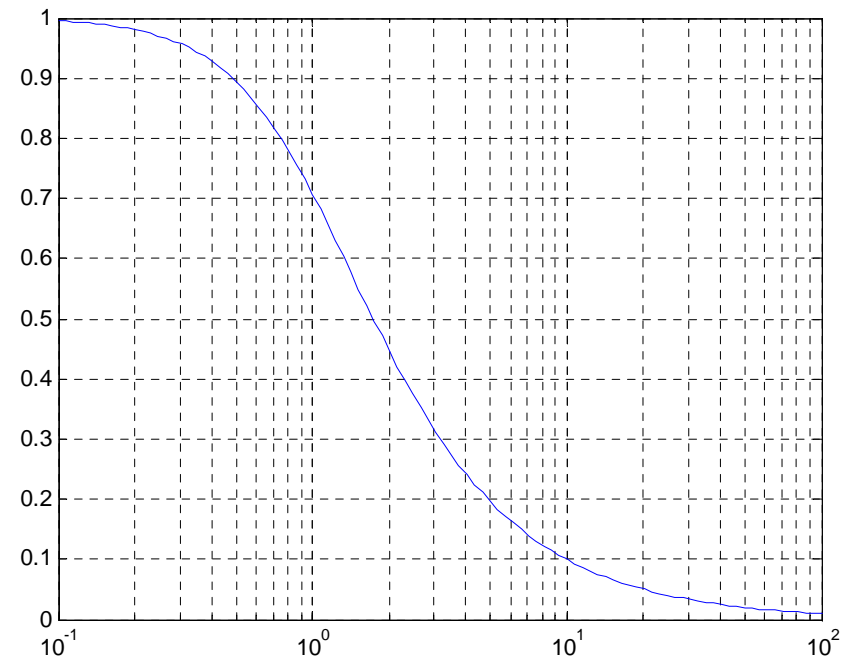
## 2.7. 그래프 그리기

### 2.7.2. 로그그래프를 그리는 명령어, `semilogx`, `semilogy`, `loglog`

다음 식의 주파수 특성을 그려라.

$$H = \frac{1}{(1+w^2)^{0.5}}$$

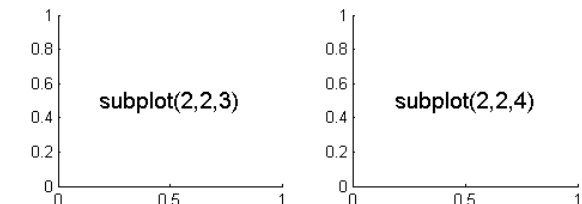
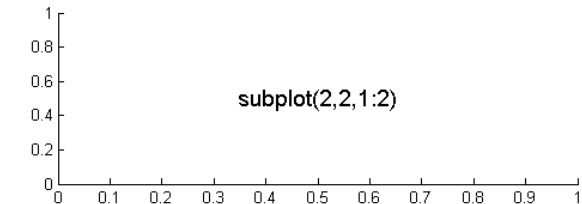
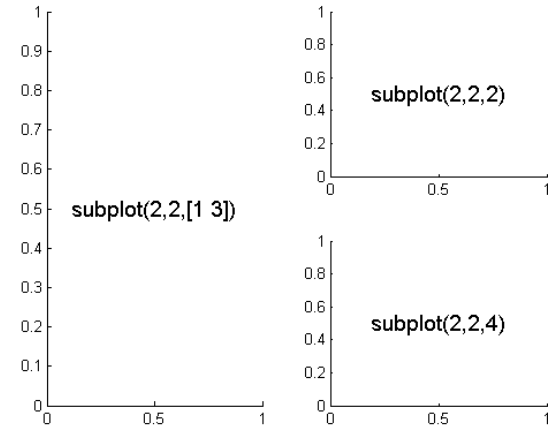
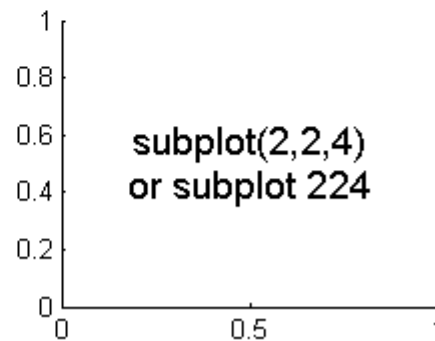
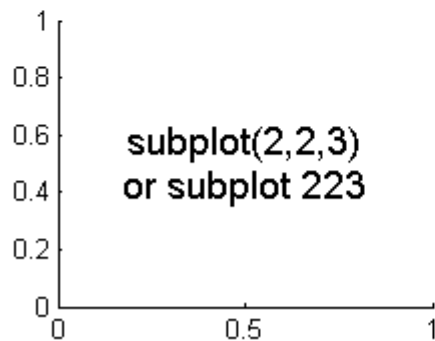
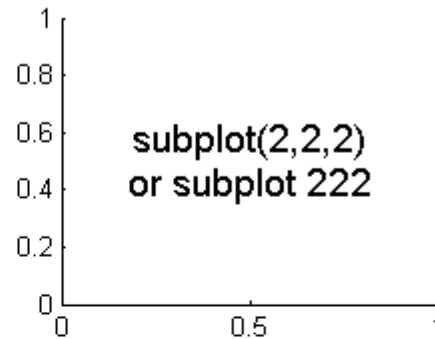
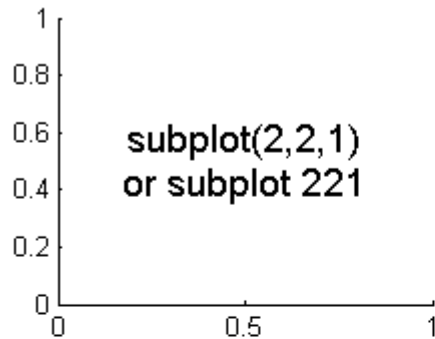
```
>> w=logspace(-1,2,100);  
>> H=1./(1+w.^2).^0.5;  
>> semilogx(w,H); grid on;
```



# 2.7. 그래프 그리기

## 2.7.3. 그래프 나누기

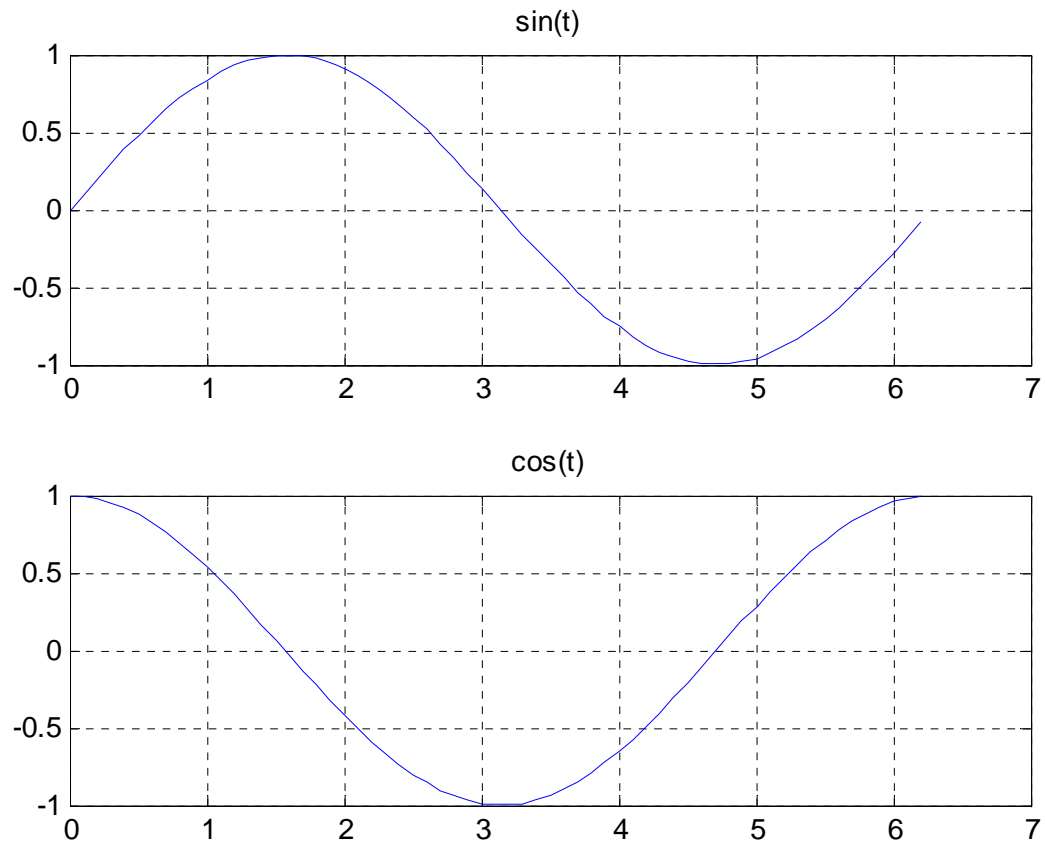
하나의 그래프 창에 여러 개의 그래프를 함께 그려서 비교하기: subplot



# 2.7. 그래프 그리기

## 2.7.3. 그래프 나누기

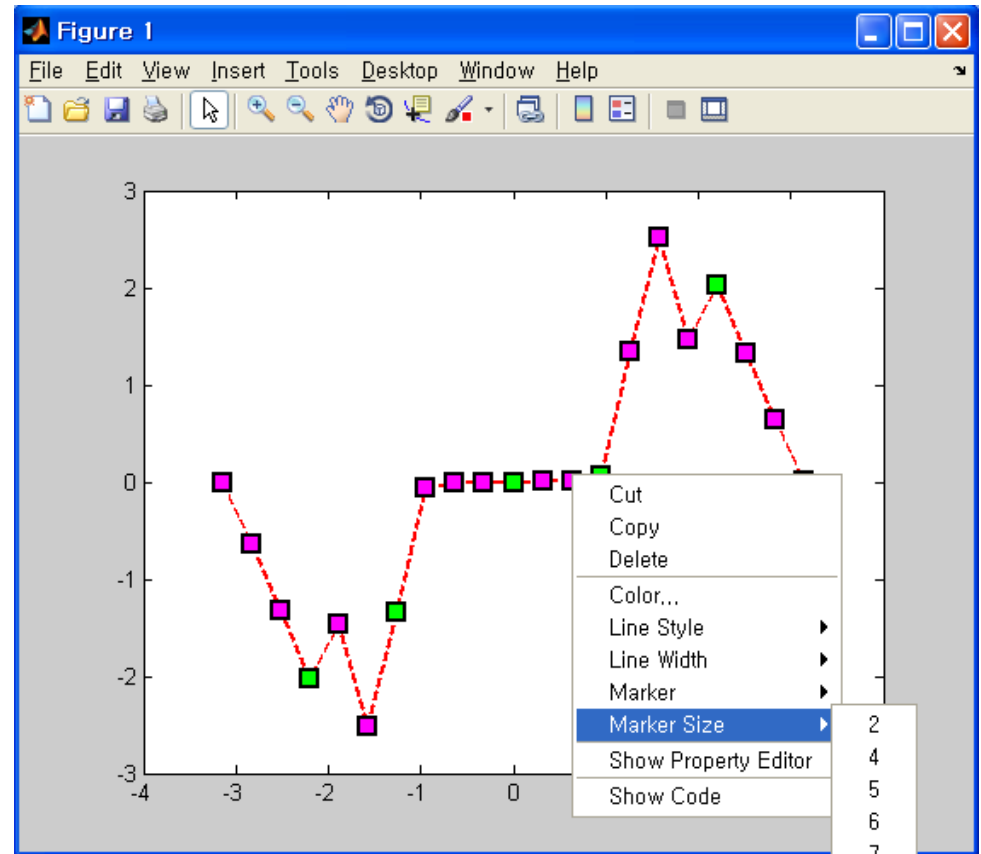
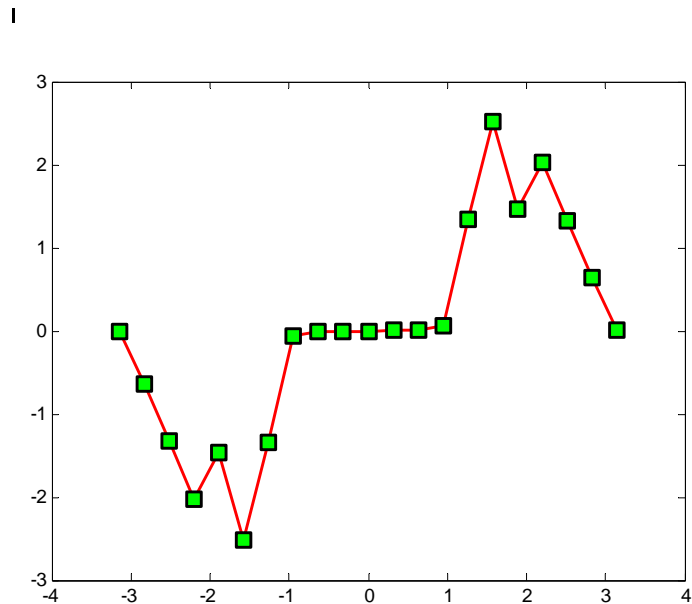
```
>> t=[0:0.1:2*pi];  
>> y1=sin(t);  
>> y2=cos(t);  
>> subplot(2,1,1);  
>> plot(t,y1); grid on;  
>> title('sin(t)');  
>> subplot(2,1,2);  
>> plot(t,y2); grid on;  
>> title('cos(t)');
```



# 2.7. 그래프 그리기

## 2.7.4. 그래프 조작하기

```
>> x = -pi:pi/10:pi;  
y = tan(sin(x)) - sin(tan(x));  
plot(x,y,'--rs','LineWidth',2,...  
      'MarkerEdgeColor','k',...  
      'MarkerFaceColor','g',...  
      'MarkerSize',10)
```



# 2.7. 그래프 그리기

## 2.7.4. 그래프 조작하기

▼ [Marker Specifiers Table](#)

Specifier	Marker Type
'+'	Plus sign
'o'	Circle
'*'	Asterisk
'.'	Point
'x'	Cross
'square' or 's'	Square
'diamond' or 'd'	Diamond
'^'	Upward-pointing triangle
'v'	Downward-pointing triangle
'>'	Right-pointing triangle
'<'	Left-pointing triangle
'pentagram' or 'p'	Five-pointed star (pentagram)
'hexagram' or 'h'	Six-pointed star (hexagram)
'none'	No marker (default)

▼ [Line Style Specifiers Table](#)

Specifier	Line Style
'-'	Solid line (default)
'--'	Dashed line
'.'	Dotted line
'-.'	Dash-dot line
'none'	No line

### ColorSpec

RGB Value	Short Name	Long Name
[1 1 0]	y	yellow
[1 0 1]	m	magenta
[0 1 1]	c	cyan
[1 0 0]	r	red
[0 1 0]	g	green
[0 0 1]	b	blue
[1 1 1]	w	white
[0 0 0]	k	black

### Lineseries Properties



# 2.7. 그래프 그리기

## 2.7.4. 그래프 조작하기

```
xlabel('text');  
ylabel('text');  
title('text');  
legend('line1','line2');
```

```
>> t=[0:0.1:2*pi];  
>> y1=sin(t);  
>> y2=cos(t);  
>> plot(t,y1,t,y2); grid on;  
>> xlabel('t');  
>> title('sin(t) and cos(t)');  
>> legend('sin(t)','cos(t)');
```

