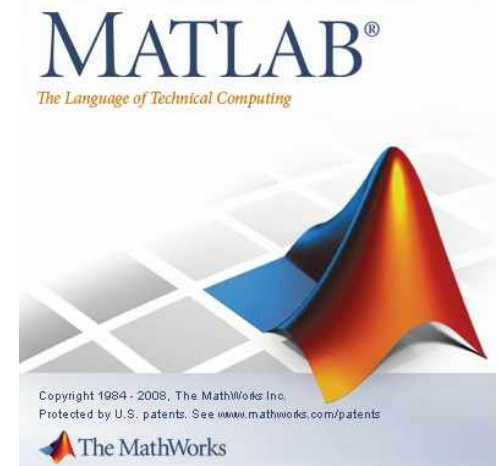
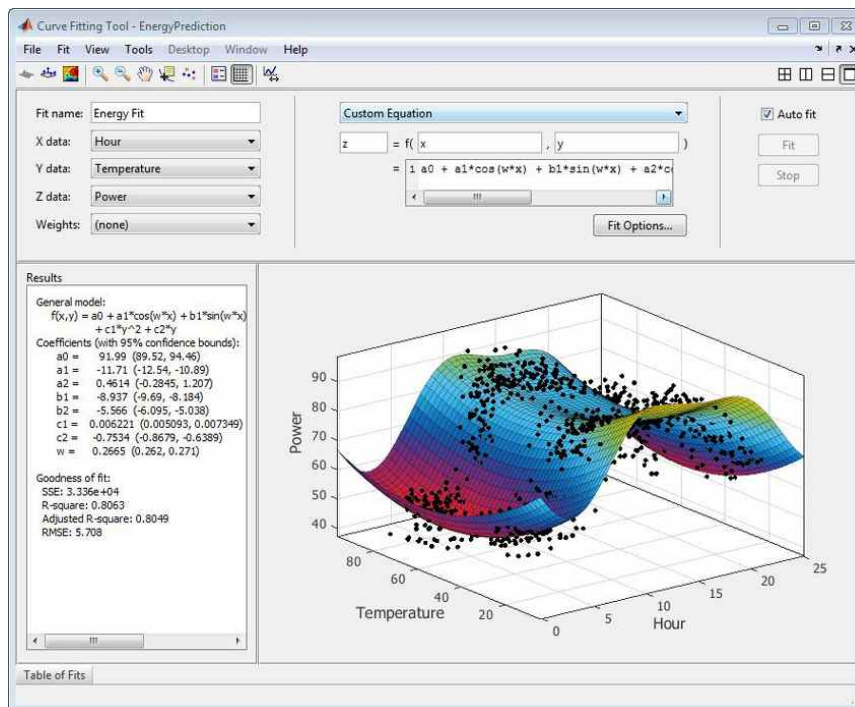


MATLAB 기초



MATLAB 30일 무료 평가판

0.1270 0.5969 0.9
0.9134 0.9575 0.9634
0.6324 0.9649 0.8003

$$f(x) = (2 + \sin(x)) \cdot \exp(-x/2)$$

$$f'(x) = (2 + \sin(x)) \cdot \exp(-x/2) - \frac{1}{2} \cdot \exp(-x/2) \cdot (2 + \sin(x))$$

$$f''(x) = (2 + \sin(x)) \cdot \exp(-x/2) - \exp(-x/2) \cdot (2 + \sin(x)) - \frac{1}{2} \cdot \exp(-x/2) \cdot (2 + \sin(x))$$

[공유하기](#)

코딩 빠르게 하기

MATLAB 30일 무료 평가판 테스트 사용



연구과정 중, 짧은 시간에 데이터를 아이디어로 아이디어는 다시 프로토타입화가 필요하다.

MATLAB®은 Windows®, Mac® 뿐만 아니라 Linux®에서 사용가능, 데이터를 빠르게 이해 하고 한번에 모든 아이디어 코딩을 바로 만드는데 필요한 도구를 제공합니다.

- ✓ 데이터 분석 및 시각화
- ✓ 알고리즘 및 애플리케이션 개발
- ✓ 수치 계산 수행
- ✓ 결과 보고 및 애플리케이션 배포

전문가

매스웍스 제품 전문가에 의해 평가판 소프트웨어가 제공되고, 사용하는 것을 도와줍니다.

[무료평가판 시작하기](#)

MATLAB

수백만의 엔지니어 및 연구원들이 생각을 공유하고 여러분야에 협업을 위해 공통적인 언어로 MATLAB을 사용합니다.

[» 자세히 알아보기](#)

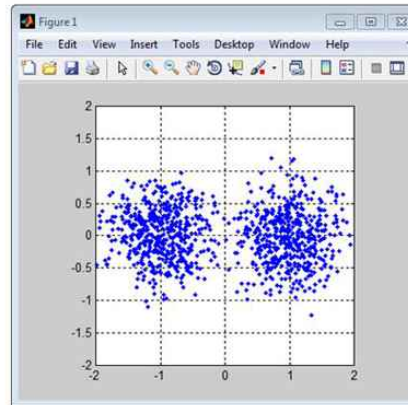
2

MATLAB Fundamentals

CHAPTER OBJECTIVES

The primary objective of this chapter is to provide an introduction and overview of how MATLAB's command mode is used to implement interactive computations. Specific objectives and topics covered are

- Learning how real and complex numbers are assigned to variables.
- Learning how vectors and matrices are assigned values using simple assignment, the colon operator, and the `linspace` and `logspace` functions.
- Understanding the priority rules for constructing mathematical expressions.
- Gaining a general understanding of built-in functions and how you can learn more about them with MATLAB's Help facilities.
- Learning how to use vectors to create a simple line plot based on an equation.



```
%% Generate vector of N bits
N = 1024;
B = rand(N,1) > 0.5;

%% Convert to transmitted signal
Tx = 1-2*B;

%% Add noise to create received signal
sigma = 0.3;
noise = sigma*(randn(N,1)+1i*randn(N,1));
Rx = Tx + noise;

%% Display constellation plot
plot(Rx, '.');
axis([-2 2 -2 2]);
axis square; grid on;
```

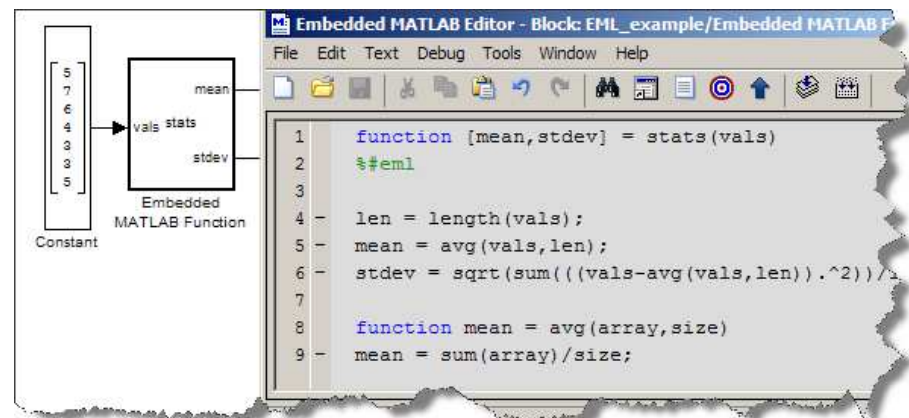
3

Programming with MATLAB

CHAPTER OBJECTIVES

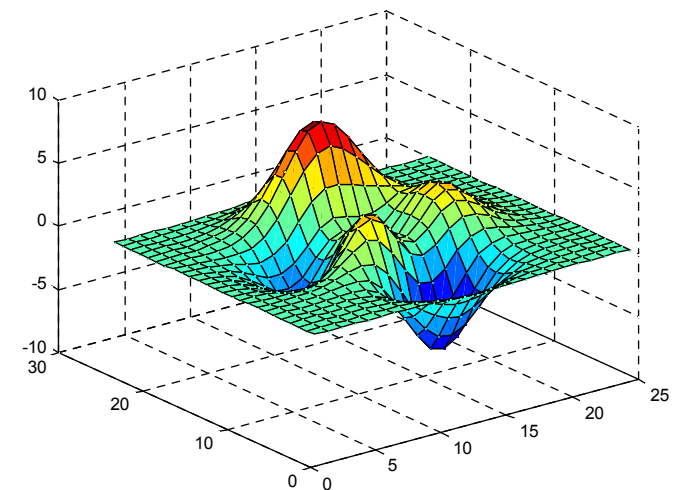
The primary objective of this chapter is to learn how to write M-file programs to implement numerical methods. Specific objectives and topics covered are

- Learning how to create well-documented M-files in the edit window and invoke them from the command window.
- Understanding how to set up M-files so that they interactively prompt users for information and display results in the command window.
- Learning how to write clear and well-documented M-files by employing structured programming constructs to implement logic and repetition.
- Understanding what is meant by vectorization and why it is beneficial.
- Understanding how functions can be passed to M-files.



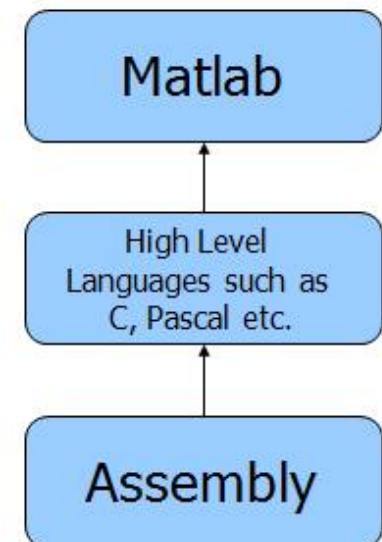
Context

- MATLAB 특징
- MATLAB Desktop Window
- 스칼라 연산
- 행렬 연산
- 2차원 그래픽
- 스크립트 파일



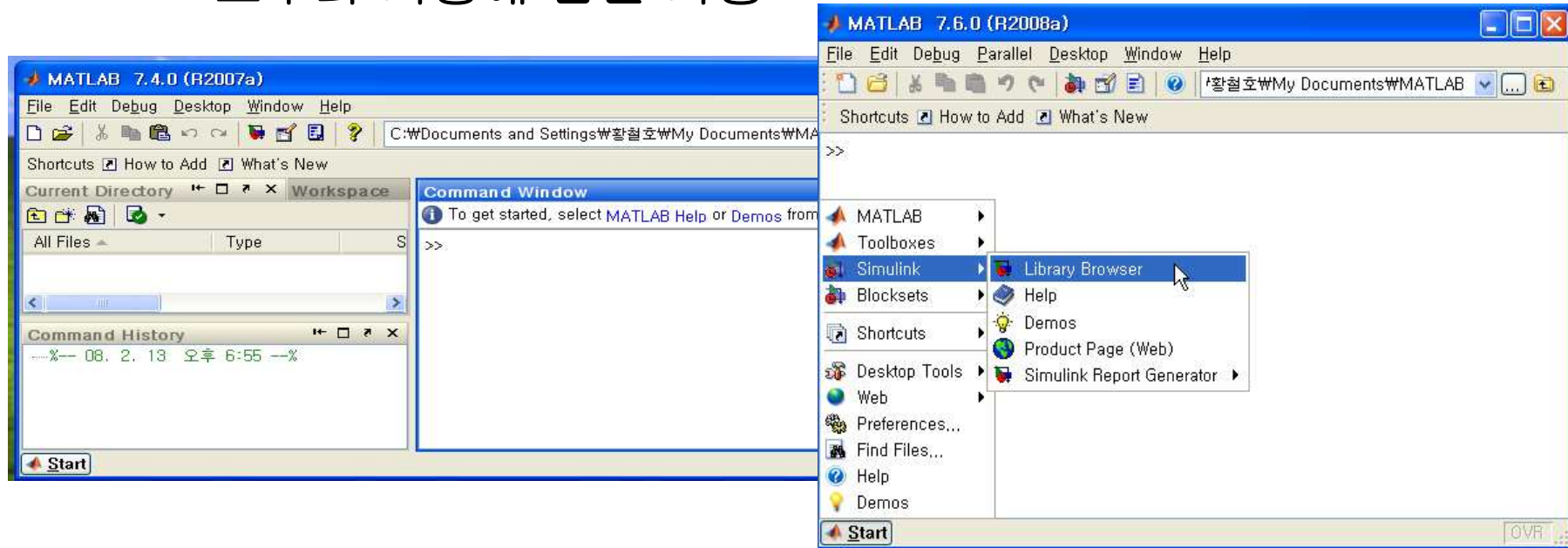
MATLAB의 특징

- Interpreter 언어 (**very** High Level Language)
 - (cf.) Compiler 언어(C, Pascal, Fortran 등)
- 수학기산 및 가시화(visualization)에 매우 편리
 - 선형대수, 데이터 분석, 신호처리, 수치적분 등
 - Graph 기능, 많은 과학계산용 내장함수
- 사용자에게 의한 함수 작성



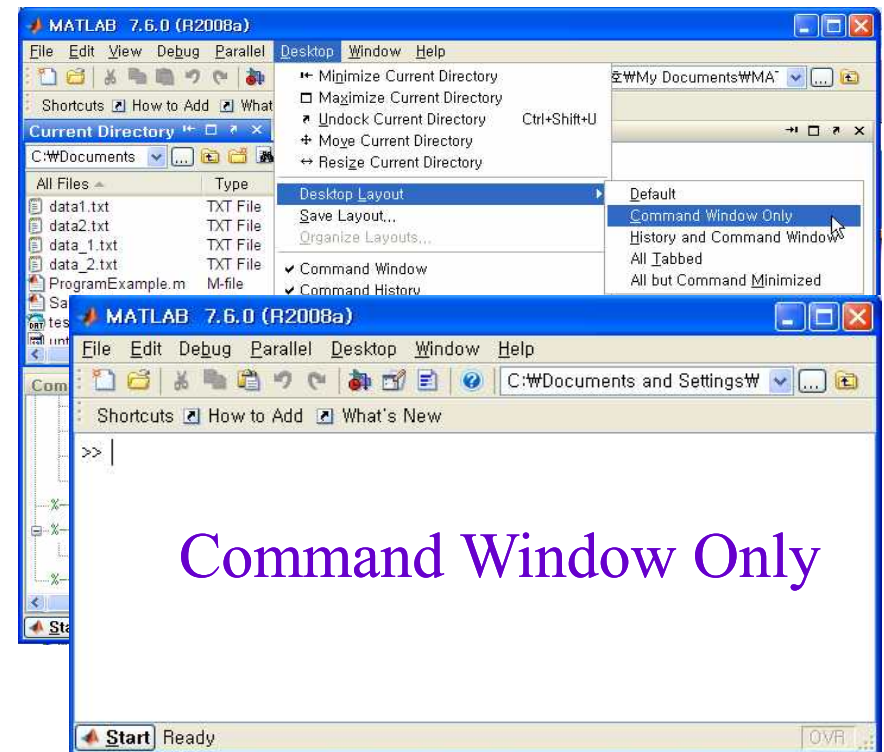
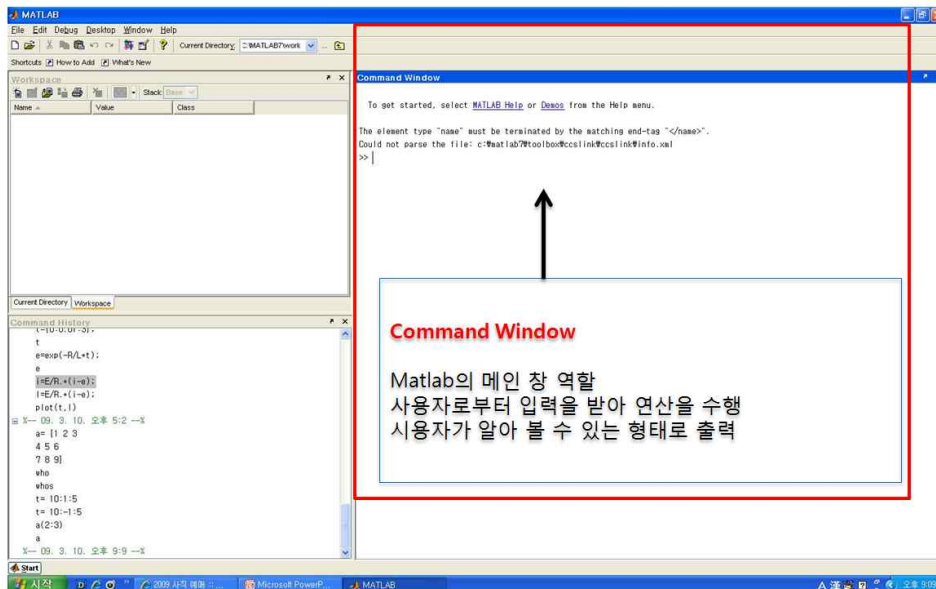
MATLAB Desktop Windows

- Desktop windows : matlab 실행시 기본화면
 - 기본 화면
 - Command Window, Current Directory Window, Command History Window 등 3개의 window
 - 왼쪽 하단부의 **Start** 버튼을 이용하여 MATLAB의 여러 도구와 기능에 접근 가능



MATLAB Desktop Windows

- 명령어 창(Command Window)
 - MATLAB을 실행시키면 나타나는 메인 창.
 - Desktop 메뉴 → Desktop Layout 메뉴에서 명령어 창 선택만 선택 가능
 - “Command Window Only” 선택



Command Window Only

MATLAB Desktop Windows

The screenshot displays the MATLAB 7.6.0 (R2008a) desktop environment. The main window is the Command Window, showing the execution of MATLAB code. The Workspace Window is open, displaying the current directory and the contents of the workspace. The Editor Window is also open, showing a script file. The Command History Window is visible at the bottom, listing the commands entered in the Command Window. A 3D surface plot titled 'Peaks' is shown in the Figure Window.

Workspace Window
사용된 변수들에 대한 정보 제공

Editor Window
프로그램 작성 및 편집

Toolbar

그림창
그래프 명령어가 실행되면 자동 생성됨

Command History Window
명령어 창에서 입력된 명령어들이 기록되어 있음

Command Window

```
>> x1=0:pi/50:4*pi;  
>> x2=0:pi/10:4*pi;  
>> plot(x1, sin(x1), 'r', x2, sin(x2), 'b');  
>> peaks  
  
z = 3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) ...  
- 10*(x/5 - x.^3) ...  
- 1/3*exp(-(x+1)^2)
```

Figure 1

Peaks

Chap10Ex2.m Prob8_12a.m

MATLAB Help Windows

The image displays four screenshots of MATLAB help windows:

- Top-left:** MATLAB 7.6.0 (R2008a) Command Window. The Command Window shows the command `>> helpdesk` entered. The MATLAB interface includes a menu bar (File, Edit, Debug, Parallel, Desktop, Window, Help) and a toolbar.
- Top-right:** Help Navigator window. The search bar contains "plot". The search results show "plot :: Functions (MATLAB Function Reference)".
- Bottom-left:** Help Navigator window. The search bar contains "MATLAB". The search results show a list of MATLAB topics under the "MATLAB" category.
- Bottom-right:** MATLAB Help window. The search bar contains "det". The search results show "MATLAB Functions: det". The window displays the "det" function reference, including its syntax, description, remarks, and algorithm. A context menu is open over the window, showing options like "Full Product Family Help", "MATLAB Help", "Using the Desktop", "Using the Command Window", "Demos", and "About MATLAB".

helpdesk will be removed in a future release. Use doc instead. (2014. 3.)
`doc [name]` : opens the Help browser.

창의 분리(undock)와 재결합(dock)

The image displays the MATLAB 7.6.0 (R2008a) interface. The main window is titled "MATLAB 7.6.0 (R2008a)" and contains several docked panes: "Command Window", "Command History", "Editor", and "Figures - Figure 1". The "Figures - Figure 1" pane shows a plot of "Data" (blue circles) and "Model" (green line). A blue callout box points to the "undock button" (a square icon with a diagonal line) in the pane's menu bar and the "dock button" (a square icon with a plus sign) in the window's menu bar. A red callout box points to the "undock" button in the "Figures - Figure 1" window's menu bar. A purple callout box points to the "dock (default)" button in the "Figures - Figure 1" window's menu bar.

undock

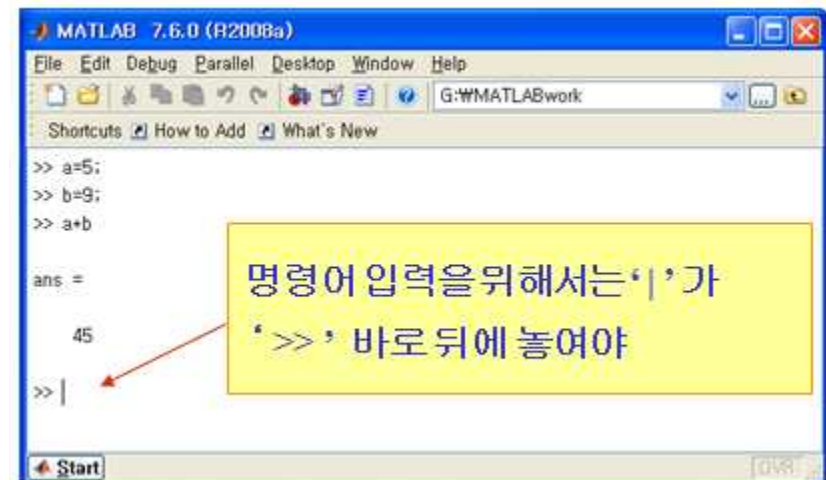
Dock (default)

undock button in the pane's menu bar

dock button in the window's menu bar

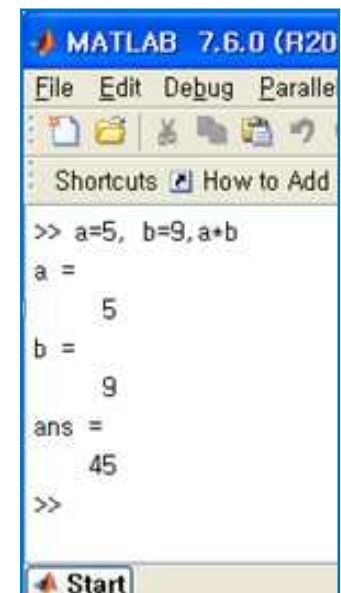
Command Window에서의 작업

- 명령어 창은 MATLAB의 메인 창
 - 명령어의 실행, 사용자가 작성한 프로그램의 실행, 다른 창 열기, 소프트웨어의 관리 등에 사용
- 명령어 입력을 위해서는 cursor (|) 가 prompt (>>) 바로 뒤에 놓여야
- 명령어를 표시하고 Enter 키를 누르면, 명령어가 실행
 - 항상 직전에 입력한 명령어만 실행 (전에 실행된 다른 것들은 변동 없음)
- 명령어 창에서 커서를 윗줄로 옮겨서 이전 명령어를 수정하여 다시 실행시키는 것 불가능



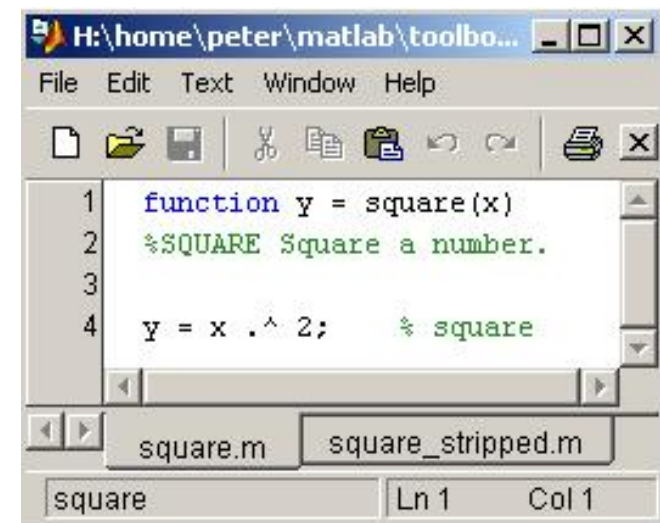
Command Window에서의 작업

- 명령어와 명령어 사이에 콤마(,)를 넣어 여러 명령어를 한 줄에 표시 가능
 - Enter 키를 누르면 왼쪽에서 오른쪽 순서대로 명령어 수행
- 위쪽 방향키 (↑)를 누를 때마다
 - 이전에 입력했던 명령어가 역순으로 프롬프트 >> 다음에 표시
 - 명령어를 수정하거나 그대로 실행 가능
 - 아래쪽 방향키 (↓)는 위쪽 방향키와 반대의 순서대로 이전 명령어들 호출 가능
- 명령어가 너무 길어 한 줄에 쓸 수 없는 경우
 - 마침표 3개 (...) 찍고 Enter 키를 누르면 다음 줄에서 이어서 작성 가능 (총 4096 글자까지)



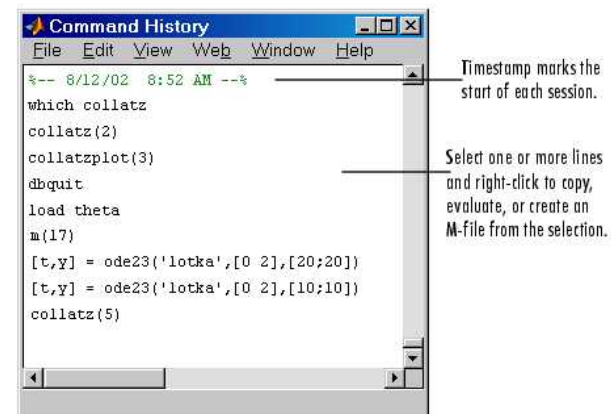
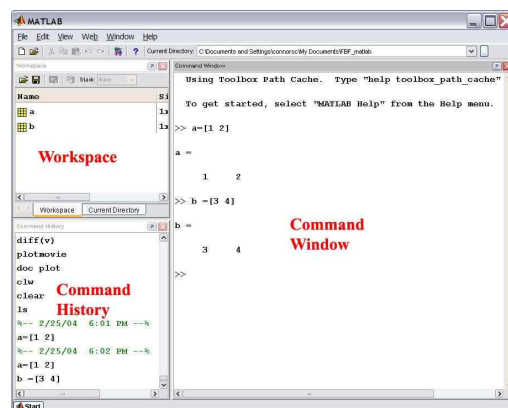
Command Window에서의 작업

- 명령어 끝에 세미콜론(;)을 붙이면 출력이 표시되지 않음
 - Output 양이 많거나 결과를 이미 알고 있을 때 유용
 - 여러 명령어를 한 줄에 쓸 때 명령어와 명령어 사이에 콤마 대신 세미콜론을 쓰면 출력이 화면에 표시되지 않음
- % : 명령어 줄 제일 앞에 쓰면 주석문(comment)
 - Enter 키를 눌러도 실행이 되지 않음
 - 같은 줄에서 명령어 다음에 % 기호와 텍스트를 같이 쓰게 되면 이하 주석문
- clc 명령어 (or Menu → Edit → Clear Command Window) : 명령어 창 clear
 - 이전에 수행된 어떠한 것도 변하지는 않음
 - 이전 정의 변수들 재사용이 가능,
 - 방향키 (↑) 이용 이전 명령어 호출 가능



Command History Window

- 그 동안 입력했던 명령어들의 목록 표시
 - 현재 명령어 창에서 입력한 명령어 뿐만 아니라 이전에 입력했던 명령어들도 포함
- Command window에서 다시 사용 가능
 - 명령어 실행
 - 명령어를 마우스로 command window 로 drag
 - 명령어를 마우스 double click 으로 실행
 - 명령어 삭제
 - 명령어 선택 후 delete 키를 누르거나
 - Edit 메뉴에서 Delete Selection을 선택
 - 명령어 위에서 마우스 우측 버튼 → 팝업메뉴 → Delete Selection 선택



스칼라 산술 연산

- 왼쪽 나눗셈을 제외한 나머지 기호들은 대부분의 계산
기에서와 동일
 - 왼쪽 나눗셈(left division)은 오른쪽 나눗셈(right division)
의 역수

| 연산 | 기호 | 예 |
|---------|-------|--------------------------|
| 덧셈 | + | $5+3$ |
| 뺄셈 | - | $5-3$ |
| 곱셈 | * | $5*3$ |
| 오른쪽 나눗셈 | / | $5/3$ |
| 왼쪽 나눗셈 | \ (₩) | $5\backslash 3 (=3/5)$ |
| 지수연산 | ^ | $5^3 (5^3 \text{을 의미함})$ |

주) 한글 Windows에서는 '\ '가
'₩'(키보드의 ₩)로 표시됨

| 우선순위 | 수학 연산 |
|------|----------------------------------|
| 1순위 | 괄호 ※ 괄호가 중첩된 경우, 가장 안쪽의 괄호부터 수행됨 |
| 2순위 | 거듭제곱 |
| 3순위 | 곱하기, 나누기(우선순위가 동등함) |
| 4순위 | 더하기와 빼기 |

내장 함수

| 함수 | 설명 | 예 |
|----------------------|---|-----------------------------------|
| sqrt(x) | 제곱근 | >> sqrt(81) ans = 9 |
| nthroot(x, n) | 실수 x 의 실수 n 제곱근. (x 가 음수이면, n 은 홀수 정수이어야 함) | >> nthroot(80, 5) ans = 2.4022 |
| exp(x) | 지수 함수(e^x) | >> exp(5) ans = 148.4132 |
| abs(x) | 절대값 | >> abs(-24) ans = 24 |
| log(x) | 자연로그. 밑이 e 인 로그(ln) | >> log(1000) ans = 6.9078 |
| log10(x) | 밑이 10인 로그 | >> log10(1000) ans = 3.0000 |
| factorial(x) | 계승 함수 $x!$ (x 는 양의 정수이어야 함) | >> factorial(5) ans = 120 |

❖ 전체 내장함수 목록은 **Help Window**에서 종류별로 분류된 목록을 참조

| 함수 | 설 명 | 예 |
|------------------|--|--------------------------------|
| round(x) | 가장 가까운 정수로 반올림(사사오입)함 | >> round(3.4) ans = 3 |
| fix(x) | 0쪽에 가까운 정수로 어림함 | >> fix(2.6) ans = 2 |
| ceil(x) | 양의 무한대에 가까운 정수로 어림함 | >> ceil(2.2) ans = 3 |
| floor(x) | 음의 무한대 쪽에 더 가까운 정수로 어림함 | >> floor(-2.25) ans = -3 |
| rem(x, y) | x를 y로 나눈 나머지를 돌려준다. | >> rem(13,5) ans = 3 |
| sign(x) | Signum 함수로서, $x > 0$ 이면 +1을, $x < 0$ 이면 -1을, $x = 0$ 이면 0을 돌려준다. | >> sign(5) ans = 1 |

MATLAB 삼각함수

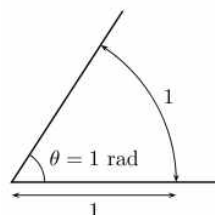
- 각이 라디안(radian) 단위일 때
 - 삼각함수 : $\sin(x)$, $\cos(x)$, $\tan(x)$, $\cot(x)$
 - 역삼각함수 : $\arcsin(x)$, $\arccos(x)$, $\arctan(x)$, $\text{acot}(x)$ 이다.
- 각이 도(degree) 단위일 때
 - 삼각함수 : $\text{sin}(x)$, $\text{cos}(x)$, $\text{tan}(x)$, $\text{cot}(x)$
 - 역삼각함수 : $\text{asin}(x)$, $\text{acos}(x)$, $\text{atan}(x)$, $\text{acot}(x)$
- 쌍곡삼각함수(hyperbolic trigonometric function)
 - $\sinh(x)$, $\cosh(x)$, $\tanh(x)$, $\coth(x)$

| 함수 | 설명 | 예 |
|-------------------------------|--|--|
| $\sin(x)$ $\text{sin}(x)$ | 각도 x 의 사인(x 는 라디안) 각도 x 의 사인(x 는 도) | <pre>>> sin(pi/6) ans = 0.5000</pre> |
| $\cos(x)$ $\text{cos}(x)$ | 각도 x 의 코사인(x 는 라디안) 각도 x 의 코사인(x 는 도) | <pre>>> cosd(30) ans = 0.8660</pre> |
| $\tan(x)$ $\text{tan}(x)$ | 각도 x 의 탄젠트(x 는 라디안) 각도 x 의 탄젠트(x 는 도) | <pre>>> tan(pi/6) ans = 0.5774</pre> |
| $\cot(x)$ $\text{cotd}(x)$ | 각도 x 의 코탄젠트(x 는 라디안) 각도 x 의 코탄젠트(x 는 도) | <pre>>> cotd(30) ans = 1.7321</pre> |

1 라디안(radian)은 원둘레 위에서 반지름의 길이와 같은 길이를 갖는 호에 대응하는 중심각의 크기로 무차원의 단위이다.

$$1 \text{ rad} = \frac{180^\circ}{\pi} \approx 57.2958^\circ$$

$$1^\circ = \frac{\pi}{180^\circ} \approx 0.0175 \text{ rad}$$



| | | |
|----------------|-----------------|--------------|
| arcsine | $y = \arcsin x$ | $x = \sin y$ |
|----------------|-----------------|--------------|

| Function | Abbreviation | Identities (using radians) |
|-----------|--------------|--|
| sine | sin | $\sin \theta = \cos \left(\frac{\pi}{2} - \theta \right)$ |
| cosine | cos | $\cos \theta = \sin \left(\frac{\pi}{2} - \theta \right)$ |
| tangent | tan | $\tan \theta = \frac{\sin \theta}{\cos \theta} = \cot \left(\frac{\pi}{2} - \theta \right)$ |
| cotangent | cot | $\cot \theta = \frac{\cos \theta}{\sin \theta} = \tan \left(\frac{\pi}{2} - \theta \right) = \frac{1}{\tan \theta}$ |
| secant | sec | $\sec \theta = \csc \left(\frac{\pi}{2} - \theta \right) = \frac{1}{\cos \theta}$ |
| cosecant | cosec | $\csc \theta = \sec \left(\frac{\pi}{2} - \theta \right) = \frac{1}{\sin \theta}$ |

Hyperbolic Definitions

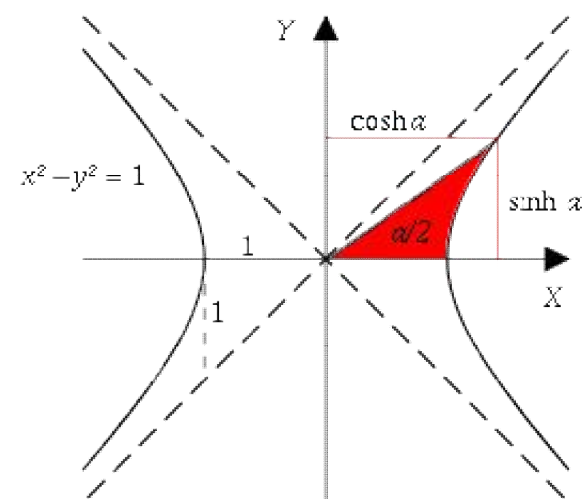
$$\sinh(x) = (e^x - e^{-x})/2$$

$$\cosh(x) = (e^x + e^{-x})/2$$

$$\tanh(x) = \sinh(x)/\cosh(x)$$

2차원 평면상에서 매개변수 t 를 사용한 자취 $(\cosh t, \sinh t)$ 가 단위원 $x^2 + y^2 = 1$ 을 그리는 것처럼,

$(\cosh t, \sinh t)$ 은 쌍곡선 $x^2 - y^2 = 1$ 을 그린다.



```
>> 4*50+3*100+10^2-400/2
ans =
    400
>> 2^5/(2^5-1)
ans =
    1.0323
>> 2*(sqrt(5)-1)/(sqrt(5)+1)^2-1
ans =
   -0.7639
>> 10*acos(-1)
ans =
    31.4159
>> x=2; y=6+x, x=y+7
y =
     8
x =
    15
>> exp(3);           % e^3
>> log10(10^5);     % log10 105
```

```
>> 7+8/2    % 8/2가 먼저 수행됨
ans =
    11
>> (7+8)/2
ans =
    7.5000
>> 4+5/3+2  % 5/3이 먼저 수행됨
ans =
    7.6667
>> 5^3/2    % 5^3이 먼저 수행됨
ans =
    62.5000
>> 27^(1/3)+32^0.2 % 27^(1/3)+32^0.2
ans =
     5
```

```
>> 27^1/3+32^0.2    % (271)/3 + 320.2
ans =
    11
>> 0.7854-(0.7854)^3/(1*2*3)+0.785^5/...
(1*2*3*4*5)-(0.785)^7/(1*2*3*4*5*6*7)
ans =
    0.7071
>> 7/2
ans =
    3.5000
>> 7/2    % \ 대신 ₩가 표시됨(한글원도)
ans =
    0.2857
>> ans
ans =
    0.2857
```

출력 형식

- 기본(default) 출력 형식
 - 5개 숫자, 소수점 이하 네 자리 고정소수점(short)
- 출력형식은 format 명령어로 변경 가능
 - format 명령어가 실행된 이후의 모든 출력은 규정된 형식으로 표시
 - 자세한 형식 : 명령어 창에서 help format
- 수를 화면에 표시하는 형식은 MATLAB이 수를 계산하고 저장하는 방법에는 영향을 미치지 않음

```
>> 4+5/3+2
ans =
    7.6667
```

| 명령어 | 설명 | 예 |
|---------------------|---|--|
| format short | 0.001<수≤1000인 수를 소수점 이하 네 자리수의 고정소수점으로 표시함. 그 외의 범위의 수는 short e 형식으로 표시함. | >> 290/7 ans = 41.4286 |
| format long | 0.001<수≤1000인 수를 소수점 이하 15자리의 고정소수점으로 표시함. 그 외 범위의 수는 long e의 형식으로 표시함. | >> 290/7 ans = 41.42857142857143 |

| 명령어 | 설명 | 예 |
|-----------------------|---|---|
| format short e | 소수점 이하 네 자리수의 과학적 표기법으로 표시함. | >> 290/7 ans = 4.1429e+001 |
| format long e | 소수점 이하 15 자리수의 과학적 표기법으로 표시함. | >> 290/7 ans = 4.142857142857143e+001 |
| format short g | 고정소수점 표시와 부동소수점 표시 중에서 더 편한 방법으로 표시. 유효숫자는 5개 | >> 290/7 ans = 41.429 |
| format long g | 고정소수점 표시와 부동소수점 표시 중 더 편한 방법으로 표시. 유효숫자는 15개. | >> 290/7 ans = 41.4285714285714 |
| format bank | 소수점 이하 두 자리까지만 표시함. | >> 290/7 ans = 41.43 |
| format compact | 화면에 많은 정보가 표시되도록 하기 위해 빈 줄을 제거함 | |
| format loose | format compact 와 반대로 빈 줄을 삽입함 | |

Check the current format setting:

```
get(0, 'format')  
ans =  
    short
```

Set the value for x and display in 5-digit scaled fixed point:

```
x = [4/3 1.2345e-6]  
x =  
    1.3333    0.0000
```

Set the format to 5-digit floating point:

```
format short e  
x  
x =  
    1.3333e+00    1.2345e-06
```

Set the format to 15-digit scaled fixed point:

```
format long  
x  
x =  
    1.333333333333333    0.000001234500000
```

Set the format to 'rational' for small integer ratio output:

```
format rational  
x  
x =  
    4/3    1/810045
```

Set an integer value for x and display it in hexadecimal (base 16) format:

```
format hex  
x = uint32(876543210)  
x =  
    343efcea
```

uint32 function:
converts the element
(integer or real) into
unsigned 32-bit (4-
byte) integers

int32 : signed integer

Variable

- 변수 : 여러 문자/숫자의 조합 (최대 63개의 문자)
 - a~z, 0~9, 밑줄(_)
 - 문자로 시작, 대소문자 구분
 - (ex) AA, Aa, aA, aa (모두 다른 변수들)
 - 내장함수, keyword 사용 불가
 - keyword 확인 명령어: `iskeyword('str')`
 - 명령어 창에서 키워드를 입력 → 키워드 파란 색으로 표시
- Assignment operator (=)

variable_name = 수치 값 또는 계산 가능한 식

```
>> a=12, B=4; C=(a-B)+40-a/B*10
a =
    12
C =
    18
```

```
>> x*2=6
??? x*2=6 Error
```

Variable

- 사전 정의 변수

| | |
|---------------------------|---|
| ans | 가장 최근의 계산값을 저장하고 있는 임시 변수 |
| eps | 두 수 사이의 최소 차이로 $2^{-52} \approx 2.2204 \times 10^{-16}$. |
| i, j | 허수 |
| Inf | 무한히 큰 수(Infinity) |
| NaN | Not-a-Number. 0/0과 같이 정의할 수 없는 수치결과 |
| pi | $\pi = 3.1415926535897...$ |
| realmax, realmin 등 | ◆ 위 변수들을 다른 값으로 다시 정의할 수도 있다. |

```
>> a=4; b=3; sum=a+b
```

```
sum = 7
```

```
>> a=7; sum
```

```
sum = 7
```

} a의 값이 바뀌어도 Excel과 달리 sum의 값은 변동이 없다.

- 명령이 길어서 한 줄에 입력할 수 없는 경우,
 - 마침표(.) 세 개를 찍고 Enter 키를 눌러도
 - 커서의 위치만 다음 줄로 바뀔 뿐 명령은 실행되지 않음
 - 연산자는 분리할 수 있으나, 변수명은 분리 불가

```
>> Class_1=20; Class_2=15; Class_3=14; Class_4=17; Total_Number=240;
```

```
>> ave = (Class_1 + Class_2 + Class_3 + Class_4) ...
```

```
/ Total_Number * 100
```

```
ans = 27.5
```



반드시 공백이 들어
가야 함

```
>> ave=(Class_1+Class_2+Class_3+Class_4)/Total_ ...
```

```
??? ave=(Class_1+Class_2+Class_3+Class_4)/Total ...
```

Error: The input character is not valid in MATLAB statements or expressions.

ans The MATLAB[®] software creates the `ans` variable automatically when you specify no output argument.

The statement `2+2` is the same as `ans = 2+2`

변수 관리에 유용한 명령어들

| 명령어 | 설 명 |
|--------------------|--|
| clear | 메모리에서 모든 변수들을 제거한다. |
| clear x y z | 메모리에서 변수 x, y, z 만을 제거한다. |
| who | 현재 메모리에 있는 변수들의 목록을 화면에 출력한다. |
| whos | 현재 메모리에 있는 변수들의 이름과 크기, 바이트와 클래스에 대한 정보를 화면에 출력한다. |

- ❖ Sometimes slowing down the execution is done deliberately for observation purposes.
You can use the command "pause"

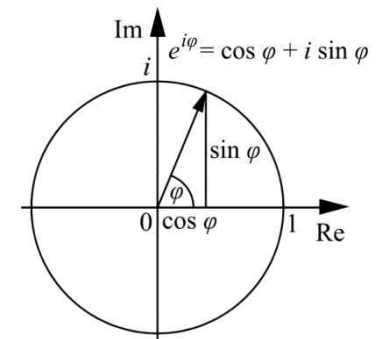
pause %wait until any key
pause(3) %wait 3 seconds

- ❖ **clc** 명령창의 내용을 모두 지운다
(**C**lears the **C**ommand window)
- ❖ **exist('var')** '**var**' 이름의 파일이나 변수가 있는지 확인한다
- ❖ **quit** **MATLAB**을 끝낸다
- ❖ **helpwin** 도움말 창을 연다
- ❖ **help 명령어** 명령창에서 '명령어'에 간단한 설명을 보여준다
- ❖ **lookfor 단어** 단어와 관련된 명령어를 찾아준다
(명령어를 잘 모를 때 사용함)
- ❖ **cd** 디렉토리 위치를 변경하거나 현재 위치를 알려준다

복소수(complex number) 연산

- MATLAB은 i 또는 j 를 이용하여 복소수를 표현
- 변수와 i 사이에는 '*'가 필요하지만 ($a+b*i$),
 - 숫자와 i 사이에 '*'를 넣지 않아도 됨 ($2i = 2*i$)
 - 숫자와 i 가 붙은 경우에는 한 개의 허수로 다뤄지지만,
 - 숫자와 i 사이에 '*'가 있는 경우에는 숫자와 i 의 곱으로 취급되므로 주의
 - 연산 순서에 따라 복소수 표현이 달라질 수도
- '+' 또는 '-' 부호 전후에 공백을 두어서는 안됨.
 - 만약 복소행렬을 '5 + 5i'와 같이 입력한다면
 - 이 값은 '5+5i'의 한 원소가 아니라 '5'와 '5i'의 두 복소수의 합으로 인식

```
>> c=10-2i    % c=10-2j  
c = 10.0000 - 2.0000i
```




```
>> re = 2; im = 5;
>> c = re + im*i
    c = 2.0000 + 5.0000i
>> (1+3i)/(1-3i)
    ans = 0.8000 + 0.6000i
>> sqrt(-1)
    ans = 0 + 1.0000i
```

```
>> c=10-2*i    % c=10-i*2
    c = 10.0000 - 2.0000i
>> y = 10/2*i    % y=(10/2)i
    y = 0 + 5.0000i
>> y = 7/2i      % y=7/(2i)
    y = 0 - 5.0000i
```

```
>> i=10; im=4; % 변수 i에 10이 할당됨
>> c=1+0.2i    % 0.2i는 허수임
    c = 1.0000 - 0.2000i
>> c=1+0.2*i    % 0.2와 변수 i(=10)를 곱함
    c=3
>> c=1+im*i    % 변수 im(=4)과 변수 i(=10)를 곱함
    c=41
>> 5 + sin(.5)i
Error: Unexpected MATLAB expression.
>> 5 + sin(.5)*i
    ans = 9.7943
```

```
>> c = 1-2i;
>> abs(c), angle(c)
    ans = 2.2361
    ans = -1.1071
>> conj(c), real(c), imag(c)
    ans = 1.0000 + 2.0000i
    ans = 1
    ans = -2
```

angle() : returns the phase angles, in radians. ($\pm\pi$)

행렬(matrix, vector) 생성

- 행렬(벡터) 생성 방법
 - 전체 원소들은 대괄호([])로 둘러쌘
 - 각 행의 원소들은 공백 또는 쉼표(,)로 분리
 - 한 행이 끝나면 원소 끝에 세미콜론(;)을 붙이거나 엔터키

```
x = [1 2 3; 5 1 4; 3 2 -1]
```

```
x =
```

```
1     2     3
5     1     4
3     2    -1
```

```
>> B=[ 1  2  3
        4  5  6
        7  8  9]
```

```
>> B= 1  2  3
        4  5  6
        7  8  9
```

```
>> A=[1 2 3]      % 1×3의 행벡터 생성
```

```
>> A= 1  2  3
```

```
>> B=[1; 2; 3]    % 3×1의 열벡터 생성
```

```
>> B= 1
```

```
2
```

```
3
```

B =

| | | | | |
|----|----|---|----|----|
| 1 | 0 | 9 | 4 | 3 |
| 0 | 8 | 4 | 2 | 7 |
| 14 | 90 | 0 | 43 | 25 |

>> C=B(2,:)

C =

| | | | | |
|---|---|---|---|---|
| 0 | 8 | 4 | 2 | 7 |
|---|---|---|---|---|

>> D=[B(1,2),B(1,4);B(3,2),B(3,5)]

D =

| | |
|----|----|
| 0 | 4 |
| 90 | 25 |

You may also delete rows and columns from a matrix using a pair of square brackets. For example, to delete the third column of matrix B, you simply enter

>> B(:,3)=[]

B =

| | | | |
|----|----|----|----|
| 1 | 0 | 4 | 3 |
| 0 | 8 | 2 | 7 |
| 14 | 90 | 43 | 25 |

Deleting Rows and Columns

You can delete rows and columns from a matrix using just a pair of square brackets. Start with

```
X = A;
```

Then, to delete the second column of X, use

```
X(:,2) = []
```

This changes X to

```
X =  
    16     2    13  
     5    11     8  
     9     7    12  
     4    14     1
```

If you delete a single element from a matrix, the result is not a matrix anymore. So, expressions like

```
X(1,2) = []
```

result in an error. However, using a single subscript deletes a single element, or sequence of elements, and reshapes the remaining elements into a row vector. So

```
X(2:2:10) = []
```

results in

```
X =  
    16     9     2     7    13    12     1
```

행렬 생성 (concatenation)

```
x = [1 2], y = [4 5], z=[ 0 0]
```

```
A = [ x y]
```

```
1 2 4 5
```

```
B = [x ; y]
```

```
1 2  
4 5
```


행렬 생성

- 일정 간격으로 증가 및 감소하는 벡터의 생성
 - 콜론(:) 연산자 이용

A = 시작값 : 증감값 : 최종값

% 증감값 생략시 디폴트 1

```
>> A=1: 2: 10  
>> A=  
      1  3  5  7  9
```

```
>> B=1:10  
>> B=  
      1  2  3  4  5  6  7  8  9 10
```

```
x = [1:4; 5:8]  
  
x =  
      1      2      3      4  
      5      6      7      8
```

행렬 생성 (내장함수)

| 함 수 | 설 명 |
|--------------------------------|---|
| <code>ones(m,n)</code> | 1로 구성된 $m \times n$ 행렬을 생성한다. |
| <code>zeros(m,n)</code> | 0으로 구성된 $m \times n$ 행렬을 생성한다. |
| <code>rand(m,n)</code> | 0과 1 사이의 값으로 균등 분포(uniformly distributed)된 난수들로 이루어진 $m \times n$ 행렬을 생성한다. |
| <code>randn(m,n)</code> | 0과 1 사이의 정규 분포(normally distributed)된 난수들의 $m \times n$ 행렬을 생성한다. |
| <code>eye(n)</code> | $n \times n$ 크기의 단위 행렬(identity matrix)을 생성한다. |
| <code>linspace(X1,X2,N)</code> | X1에서 X2까지 균등한 간격으로 N개의 원소를 갖는 행벡터를 생성한다. |

```
B = ones(2)
B = [1,1;1,1]
B = ones(2,2)
B = [1,1;1,1]
```

```
I = eye(4)
```

```
I =
```

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

```
I = eye(2,3)
```

```
I =
```

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

```
x = zeros(1,3)
```

```
x =
```

```
0    0    0
```

```
x = ones(1,3)
```

```
x =
```

```
1    1    1
```

```
x = rand(1,3)
```

```
x =
```

```
0.9501  0.2311  0.6068
```

행렬 생성 (index)

- 첨자(index) 이용 : 1부터 (not 0 (as in C))
 - keyword `end` : the last row or column

`A(i), B(i, j), C(i, j, k)`

```
A =  
  
     3     5     3  
     6     8     2  
     2     7     3
```

```
>> A(6)
```

```
ans =
```

```
7
```

```
>> A(3,2)
```

```
ans =
```

```
7
```

```
>> A(2, :)
```

```
ans =
```

```
6     8     2
```

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

```
ans =
```

```
5  
8
```

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

```
>> A(2)=-2
```

```
A=
```

```
1    -2    5
```

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

```
>> B(2,2)=-1
```

```
>> B=
```

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

computes the sum of the elements in the last column of A:

```
sum(A(:,end))
```

행렬 연산

- 행렬의 덧셈과 뺄셈
 - 연산의 대상인 두 행렬의 크기는 동일해야 함
- 행렬의 곱셈
 - 행렬 X와 Y의 곱셈에서
 - X의 열크기와
 - Y의 행 크기가 일치해야 함

+ addition
- subtraction
* multiplication
/ division
^ power
' complex conjugate transpose

Power

```
>> B=[1 2; 3 4];  
>> B ^ 2  
ans =  
    7    10  
   15    22
```

정방행렬(square matrix)이어야

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

A =

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

```
>> B = [3 5 2; 5 2 8; 3 6 9]
```

B =

| | | |
|---|---|---|
| 3 | 5 | 2 |
| 5 | 2 | 8 |
| 3 | 6 | 9 |

Addition

```
>> X = A + B
```

X =

| | | |
|----|----|----|
| 4 | 7 | 5 |
| 9 | 7 | 14 |
| 10 | 14 | 18 |

Subtraction

```
>> Y = A - B
```

Y =

| | | |
|----|----|----|
| -2 | -3 | 1 |
| -1 | 3 | -2 |
| 4 | 2 | 0 |

Product

```
>> Z = A * B
```

Z =

| | | |
|----|-----|-----|
| 22 | 27 | 45 |
| 55 | 66 | 102 |
| 88 | 105 | 159 |

Transpose

```
>> T = A'
```

T =

| | | |
|---|---|---|
| 1 | 4 | 7 |
| 2 | 5 | 8 |
| 3 | 6 | 9 |

행렬 연산 (left/right division)

- 행렬의 나눗셈
 - $A \setminus B$: (A의 역행렬)과 B의 곱셈 (W)
 - $A * X = B$ 의 해 ($X = \text{inv}(A) * B$)
 - A / B : A와 (B의 역행렬)의 곱셈
 - $X * B = A$ 의 해 ($X = A * \text{inv}(B)$)

```
>> C=A\B
```

```
C =
```

```
-4.5556 -5.3333 -4.5556  
 5.1111  5.6667  4.1111  
-1.2222 -0.6667  0.7778
```

```
>> X=inv(A)*B
```

```
X =
```

```
-4.5556 -5.3333 -4.5556  
 5.1111  5.6667  4.1111  
-1.2222 -0.6667  0.7778
```

```
>> 4\2           % 2/4와 동일한 의미
```

```
ans =  
    0.5000
```

```
>> 4/2           % 4/2와 동일한 의미
```

```
ans =  
     2
```

행렬 연산 (element by element)

- `.(dot)` 연산자
 - 각 원소끼리 곱 또는 나누는 등의 연산 시 사용

`.*` : element-by-element multiplication
`./` : element-by-element division
`.^` : element-by-element power

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

```
x = A(1,:)
```

```
x =
     1     2     3
```

```
y = A(3,:)
```

```
y =
     3     4    -1
```

```
b = x.*y
```

```
b =
     3     8    -3
```

```
c = x./y
```

```
c =
    0.33    0.5   -3
```

```
d = x.^2
```

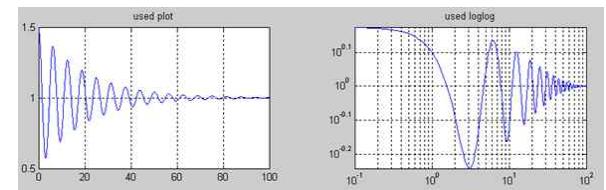
```
d =
     1
```

```
K = x^2
Error:
??? Error using ==> mpower Matrix must be square.
B = x*y
Error:
??? Error using ==> mtimes Inner matrix dimensions must agree.
```

2차원 그래픽

❖ 기본적인 그래프 함수

- ◆ **plot** : 선형 축을 사용하여 그래프 출력
- ◆ **stem** : 이산적인 축을 사용하여 그래프 출력
- ◆ **loglog** : 로그 축을 사용하여 그래프 출력
- ◆ **semilogx** : x축은 로그 스케일을 사용, y축은 선형 스케일을 사용하여 그래프 출력
- ◆ **semilogy** : y축은 로그 스케일을 사용, x축은 선형 스케일을 사용하여 그래프 출력
- ◆ **polar** : 극좌표 축을 사용하여 그래프 출력
- ◆ **subplot** : 하나의 그림창을 여러 개의 그래프 출력을 얻고자 할 경우
- ◆ **axis** : x축과 y축의 크기를 조정할 경우
- ◆ **grid** : 격자선을 그래프에 추가하여 출력
- ◆ **title** : 그래프의 제목을 추가하여 출력
- ◆ **xlabel** : x축에 축의 이름을 추가하여 출력
- ◆ **ylabel** : y축에 축의 이름을 추가하여 출력
- ◆ **gtext** : 마우스를 이용하여 지정한 위치에 문자를 출력
- ◆ **text** : 지정한 위치에 문자를 출력



2차원 그래픽

- 그래프 그리기

```
plot(x, y, 'linespec')
```

- 가장 간단한 그래픽 함수
- linespec : 선의 색상 및 종류, 마커를 나타내는 문자열
 - marker : 사용된 각 데이터 좌표값에 표시되는 기호

| 색 상 | 종 류 | 마 커 |
|-------------|--------------------|---------|
| c (cyan) | - (solid line) | + |
| m (magenta) | -- (dashed line) | ^ |
| y (yellow) | none | O |
| r (red) | : (dotted line) | * |
| g (green) | -. (dash-dot line) | . |
| b (blue) | | X |
| w (white) | | square |
| k (black) | | diamond |

그래프의 속성

| 가능한 선의 Color | | | |
|---------------|-------------|---------------|---------------|
| Matlab Symbol | Color | Matlab Symbol | Color |
| c | 하늘 | g | 초록 |
| m | 자주 | b | 파랑 |
| y | 노랑 | w | 흰색 |
| r | 빨강 | k | 검은색 |
| 가능한 선의 Style | | | |
| Matlab Symbol | Style | Matlab Symbol | Style |
| - | Solid line | : | Dotted line |
| -- | Dashed line | -. | Dash-dot line |
| none | No Line | | |

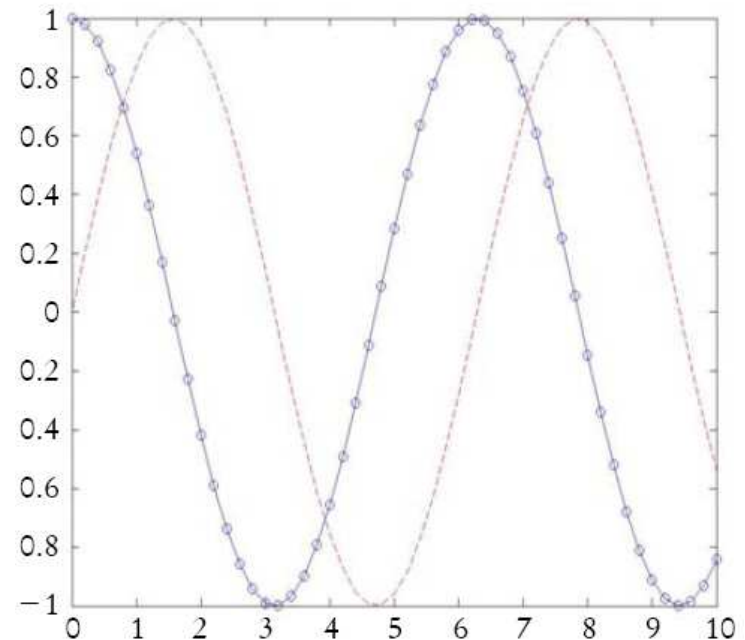
| 가능한 선의 marker | | | |
|---------------|--------------|---------------|---------------------|
| Matlab Symbol | Marker Style | Matlab Symbol | Marker Style |
| + | + | ^ | △ |
| o | o | v | ▽ |
| * | * | > | ▷ |
| . | ● | < | ◁ |
| x | x | pentagram | ☆ |
| Square | □ | hexagram | ⬡ |
| Diamond | ◇ | none | No marker (default) |

2차원 그래픽

- 하나의 그림 창에 여러 그래프를 도시할 때

```
plot(x1, y1, 'linespec1', x2, y2, 'linespec2', ..., xn, yn, 'linespecn')
```

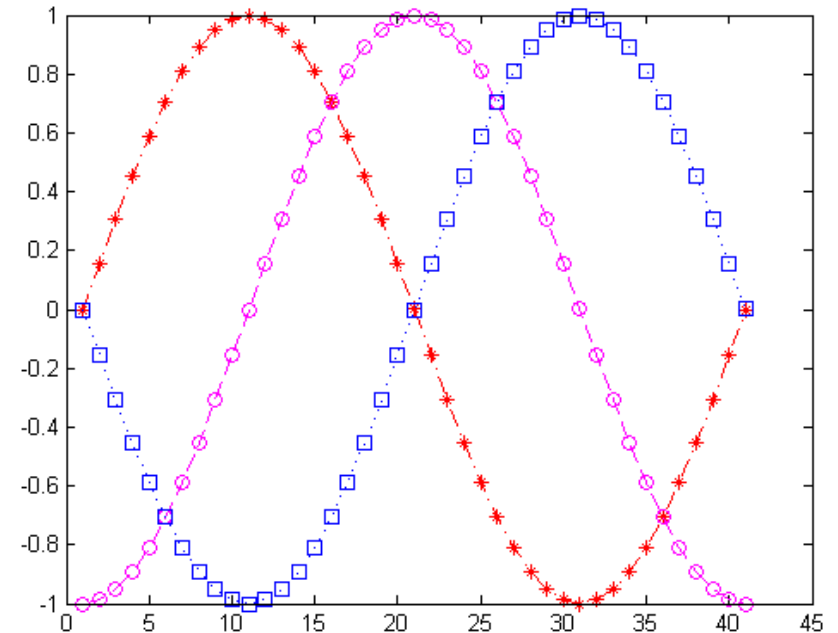
```
>> x=0 : 0.2 : 10 ;  
>> y1=cos(x) ;  
>> y2=sin(x) ;  
>> plot(x, y1, 'b-o', x, y2, 'r--')
```



2차원 그래픽

- Hold on/off
 - Hold on : 이미 존재하는 그래픽 위에 덮어서 그리고자 할 때
 - Hold off : 더 이상 같은 그림창에 그래프를 추가하지 않을 때

```
t = 0:pi/20:2*pi;  
plot(t,sin(t),'-r*')  
hold on  
plot(t,sin(t-pi/2),'--mo')  
plot(t,sin(t-pi),':bs')  
hold off
```



2차원 그래픽

- 그래프 축의 최대값과 최소값 지정

`axis(v)`

- `v`는 `x`축 및 `y`축의 최소 및 최대값의 행벡터

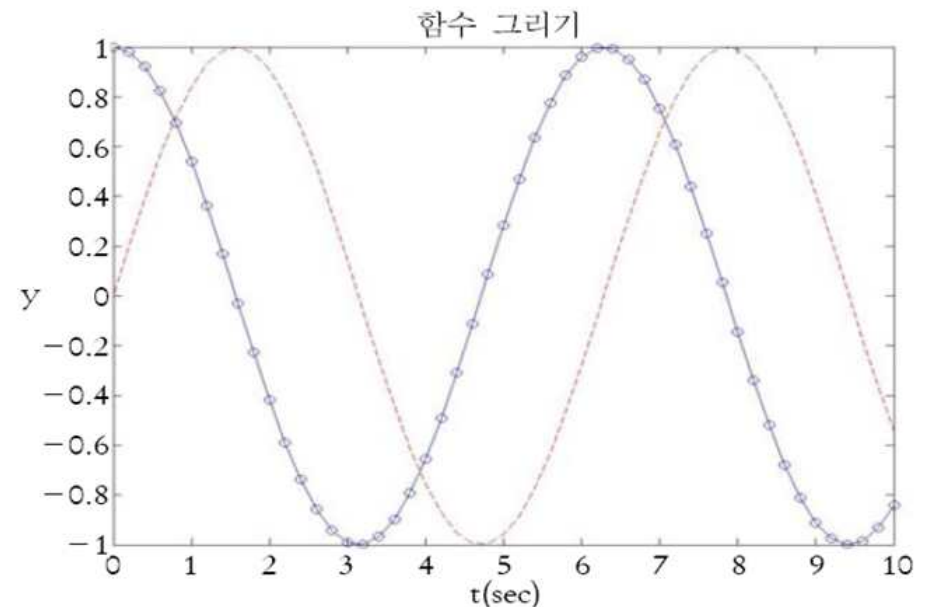
```
>> v=[0, 10, -2, 2];  
>> x=0:0.2:10;  
>> y=3*cos(x);  
>> plot(x, y);  
>> axis(v)      % x축의 최소값과 최대값은 0과 10, y축의 최소값과 최대  
                  값은 -2와 2
```

2차원 그래픽

- 그래프의 제목과 축 이름 붙이기

```
title('문자열'), xlabel('문자열'), ylabel('문자열')
```

```
>> x=0 : 0.2 : 10 ;  
>> y1=cos(x) ;  
>> y2=sin(x) ;  
>> plot(x, y1, 'b-o', x, y2, 'r--')  
>> title('함수 그리기')  
>> xlabel('t(sec)')  
>> ylabel('y')
```



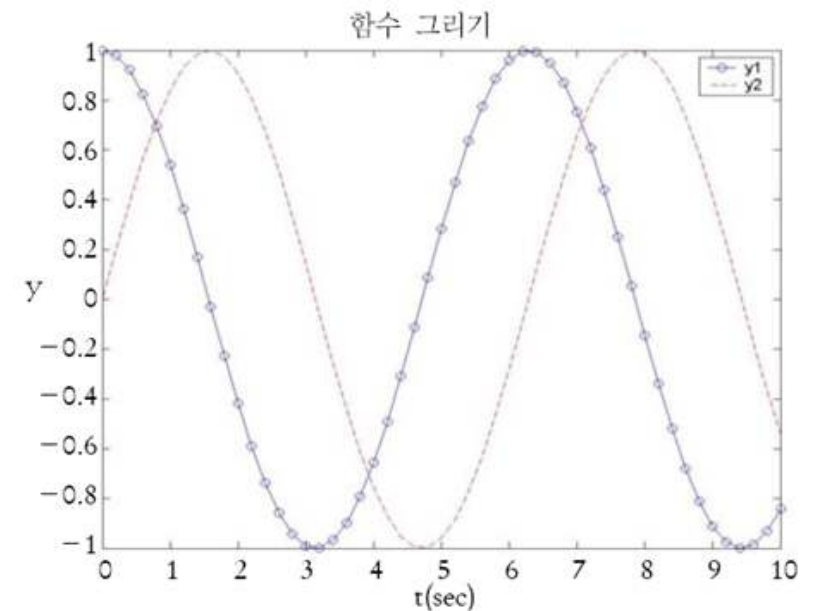
2차원 그래픽

- 범례 출력하기

```
legend('문자열')
```

- 그래프 객체에 대한 설명문
- 여러 개의 그래프가 그려졌을 때 이를 구분하기 위해 사용

```
>> legend('y1', 'y2')
```



2차원 그래픽

- 텍스트 출력하기

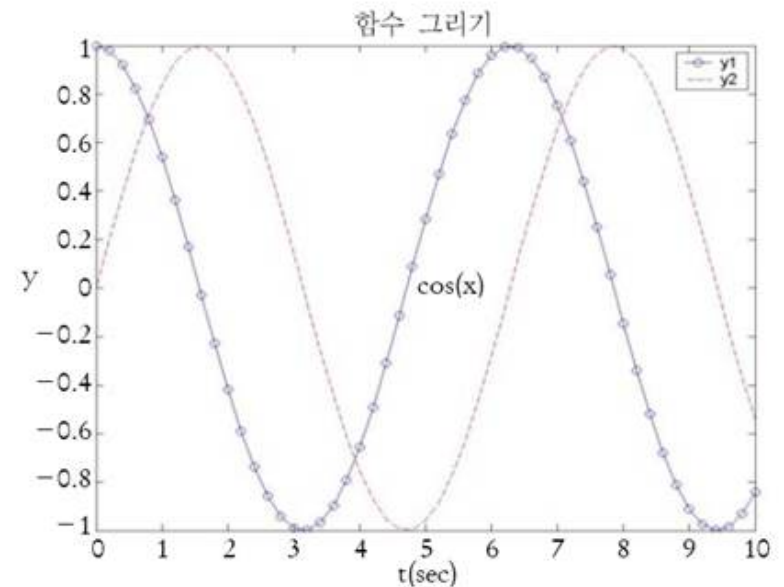
- 그래프 객체의 특정 부분에 대한 설명을 위해 추가되는 문자

`text(x 좌표, y 좌표, '문자열')` 또는 `gtext('문자열')`

- `text` 또는 `gtext` 함수 사용

- `gtext` 함수 : 문자열이 나타나기를 원하는 곳에 마우스를 클릭

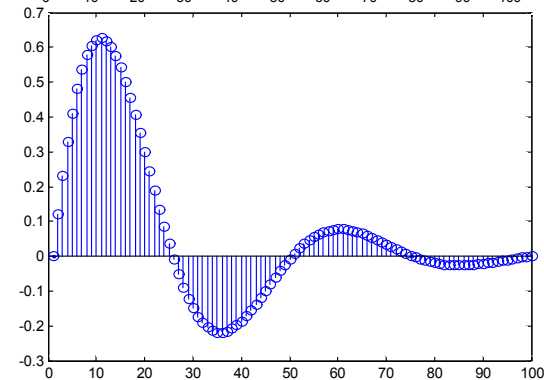
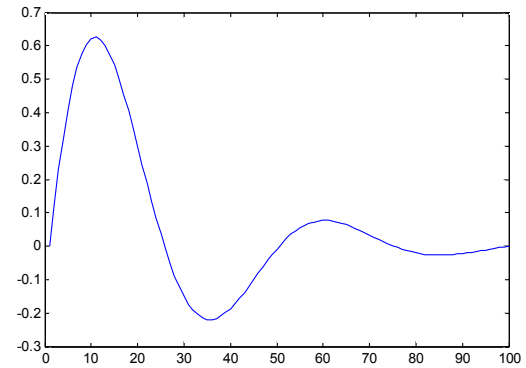
`>> gtext('cos(x)')`



2차원 그래픽

- `plot(.)`

```
>>x=linspace(0,4*pi,100);  
>>y=sin(x);  
>>plot(y)  
>>plot(x,y)
```

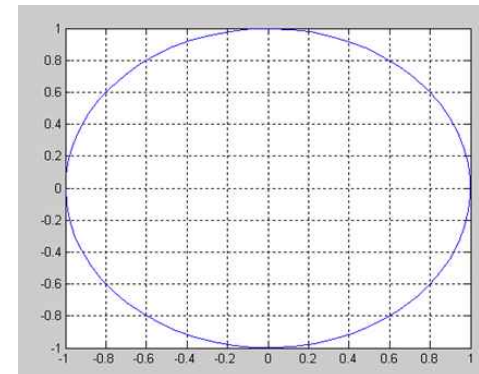


- `stem(.)`

```
>>stem(y)  
>>stem(x,y)
```

- `grid on`

```
>> i = -pi:0.1:pi;  
>> x = cos(i);  
>> y = sin(i);  
>> plot(x,y);  
>> grid on;
```



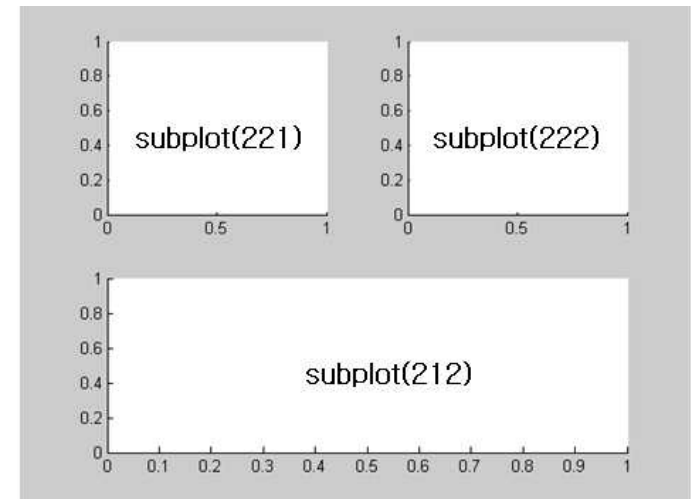
2차원 그래픽

- figure
 - 새로운 그림 창을 생성
 - 이미 존재하는 그림창을 활성화 한다.
 - figure : 새로운 그림 창 생성
 - figure(10) : 10번 그림 창 생성 혹은 활성화
- Multi graphic 객체
 - subplot(MNI) or subplot(M,N,I)
 - M(행의 개수), N(열의 개수), I(하부영역의 순서)

```
>> subplot(221)
```

```
>> subplot(222)
```

```
>> subplot(212)
```

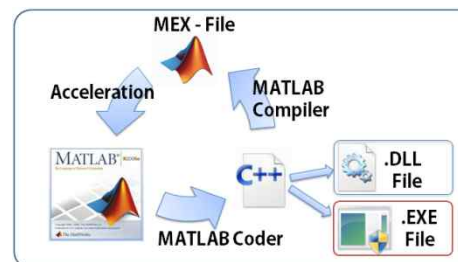
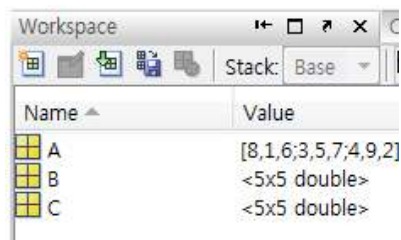


MATLAB 파일

- MATLAB은 파일에 수행할 모든 명령어를 기록하고 저장한 후 파일 수행 가능
 - 반복된 명령어들을 수행시켜야 하거나,
 - 어떤 조건이나 앞의 계산결과에 따라 후속 명령어를 결정해야 하는 경우,
 - 다양한 입력데이터에 대해 같은 일련의 명령어들을 수행해야 하는 경우
 - 파일 종류 : M 파일, MAT 파일, MEX 파일, diary 파일, 입력데이터용 text
- M 파일
 - Script file, function file 2 종류
 - 매트랩 내장된 Editor나 메모장 이용하여 표준 ASCII 파일로 작성
 - 확장자 : 'm' (예 : test.m)
 - 파일 내의 명령어들은 명령어 창에서 입력하는 것처럼 파일에 기록된 순서대로 실행

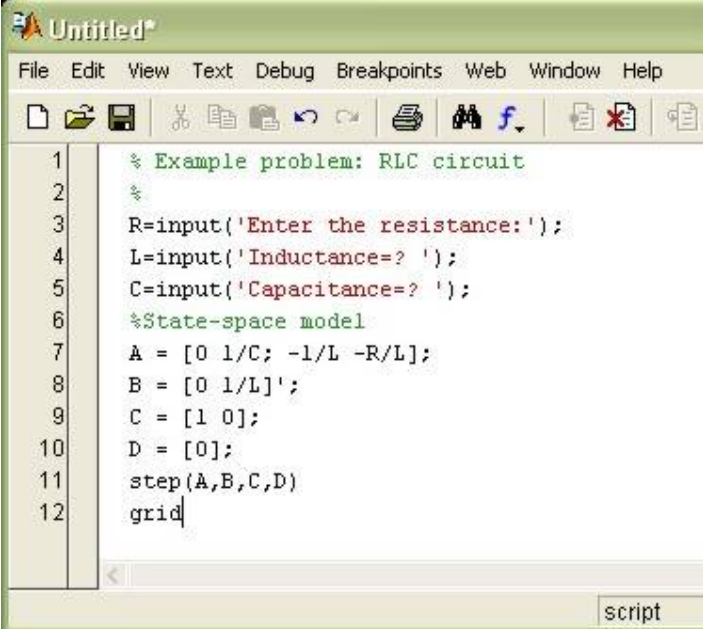
MAT/MEX/Diary File

- MAT 파일
 - 데이터 저장용 파일, 확장자 (.mat) (cf. 함수(m), GUI(fig))
 - workspace에있는 변수들을 mat-file로 하드디스크에 저장
 - >>save ('하드디스크에 저장할 이름','workspace에 있는 변수이름')
 - workspace에 있는 변수가 하드디스크에 .mat 파일로 저장
 - (ex) >>save('test','A') : workspace에 있는 A라는 변수를 하드디스크에 test.mat으로 저장
- MEX 파일
 - Matlab 에서 지원하는 이기종 언어를 인터페이스
 - MEX-file : Fortran, C/C++로 작성된 코드를 Matlab에서 직접 호출할 수 있도록 하는 인터페이스 함수들로 구성
- Diary 파일
 - The diary function : log (keyboard input and the resulting text output) 생성
 - The output of diary : ASCII file (text)
 - diary('filename') : 기존 파일 존재시 append



Script File

- Script로서의 M-파일 사용
 - 특정 작업을 여러 번 수행해야 할 경우,
 - 결과를 출력하는 명령어를 포함 → 명령어 창에 출력
 - 그래픽 → Figure Window 에 출력
 - 스크립트 파일의 실행으로 만들어진 변수는 전역변수(global variable)
 - Workspace에 저장되어 명령어 창에서 언제든지 이용 가능
 - 명령창에서 이루어질 수 있는 모든 명령문을 M-파일에 작성
 - 명령창에서 파일명을 입력하여 실행



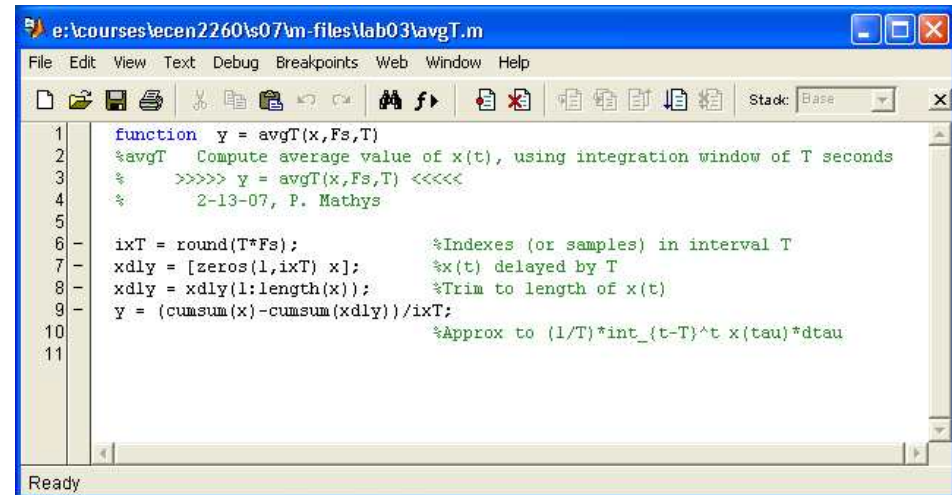
```
1  % Example problem: RLC circuit
2  %
3  R=input('Enter the resistance:');
4  L=input('Inductance=? ');
5  C=input('Capacitance=? ');
6  %State-space model
7  A = [0 1/C; -1/L -R/L];
8  B = [0 1/L]';
9  C = [1 0];
10 D = [0];
11 step(A,B,C,D)
12 grid
```

script

User Defined Functions

- Function (함수)으로서의 M-파일의 사용
 - 외부로부터 값을 받아 연산 수행 후, 다시 외부로 값을 반환
 - 입력값과 반환값의 수에 따른 형태
 - m-file의 첫줄에 matlab 에게 function 임을 표시
 - M-file 이름을 function 이름과 같게 해야함.

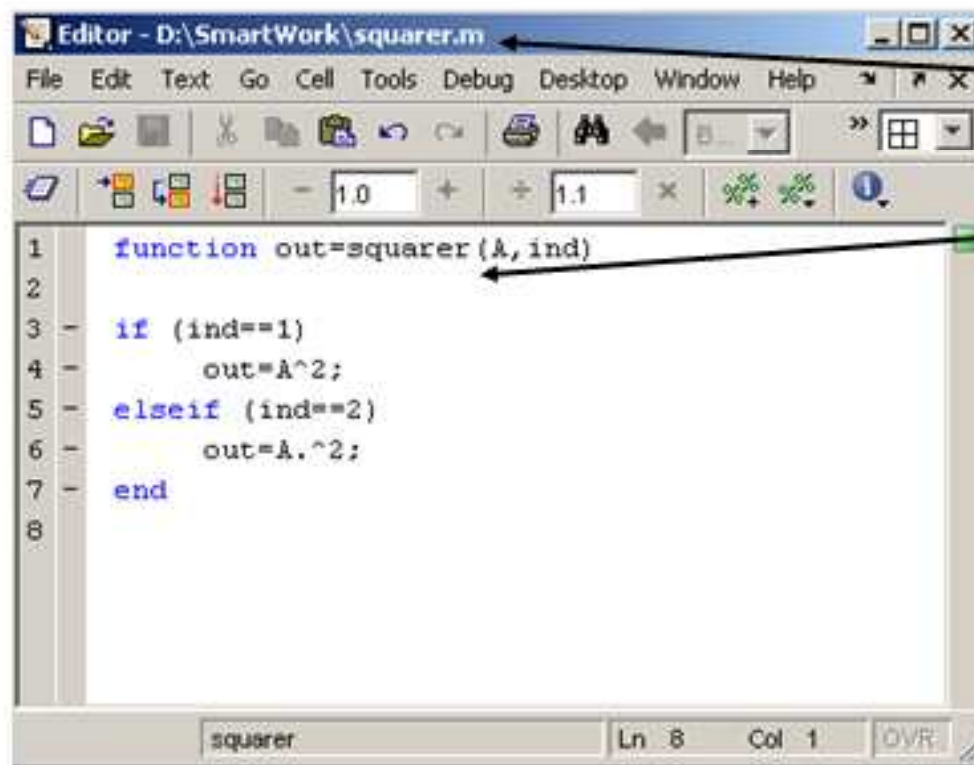
```
function out1=functionname(in1)
function out1=functionname(in1,in2,in3)
function [out1,out2]=functionname(in1,in2)
```



```
e:\courses\ecen2260\s07\m-files\lab03\avgT.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: Base
1 function y = avgT(x,Fs,T)
2 %avgT Compute average value of x(t), using integration window of T seconds
3 % >>>> y = avgT(x,Fs,T) <<<<<
4 % 2-13-07, P. Mathys
5
6 ixT = round(T*Fs); %Indexes (or samples) in interval T
7 xdly = [zeros(1,ixT) x]; %x(t) delayed by T
8 xdly = xdly(1:length(x)); %Trim to length of x(t)
9 y = (cumsum(x)-cumsum(xdly))/ixT;
10 %Approx to (1/T)*int_{t-T}^t x(tau)*dtau
11
```

❖ Examples

- ◆ Write a function : `out=squarer(A, ind)`
 - Which takes the square of the input matrix if the input indicator is equal to 1
 - And takes the element by element square of the input matrix if the input indicator is equal to 2



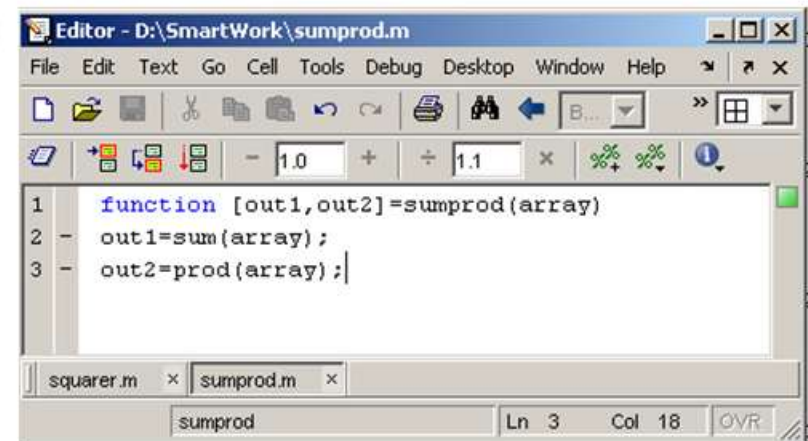
The image shows a MATLAB Editor window titled 'Editor - D:\SmartWork\squarer.m'. The window contains the following code:

```
1 function out=squarer(A, ind)
2
3 if (ind==1)
4     out=A^2;
5 elseif (ind==2)
6     out=A.^2;
7 end
8
```

The status bar at the bottom indicates 'Ln 8 Col 1' and 'OVR'.

Same Name

- ❖ Another function which takes an input array and returns the sum and product of its elements as outputs

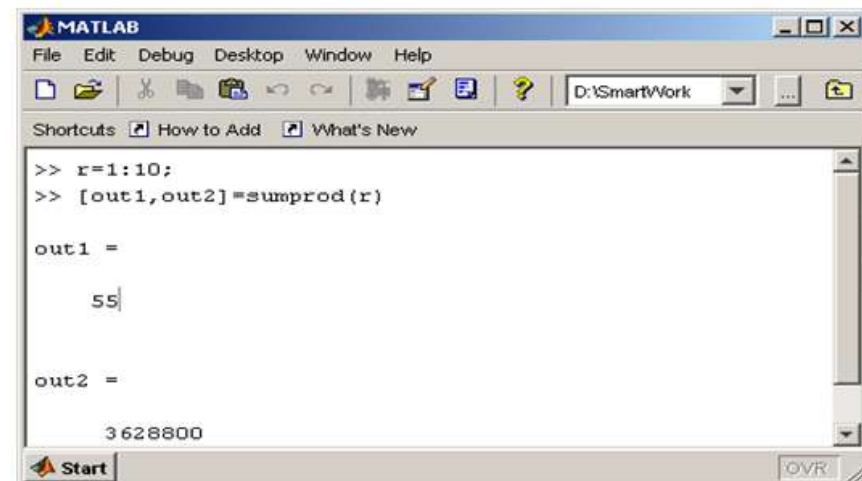


The image shows a MATLAB Editor window titled "Editor - D:\SmartWork\sumprod.m". The window contains the following code:

```
1 function [out1,out2]=sumprod(array)
2 - out1=sum(array);
3 - out2=prod(array);
```

The status bar at the bottom indicates "sumprod" and "Ln 3 Col 18 OVR".

- ❖ The function sumprod(.) can be called from command window or an m-file as



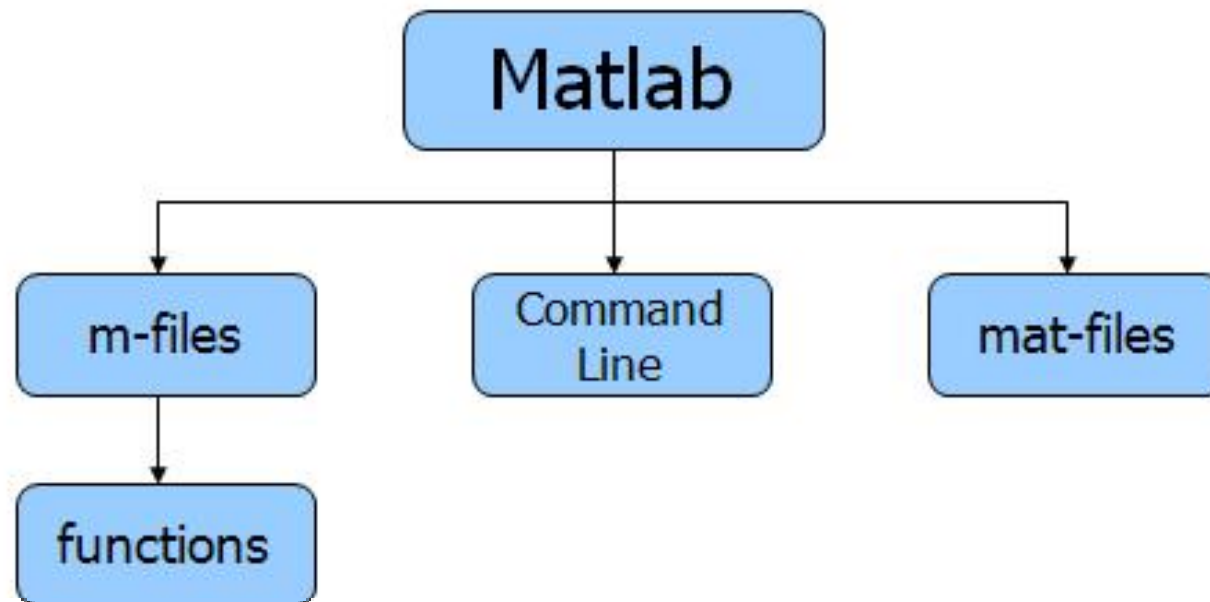
The image shows a MATLAB Command Window with the following commands and outputs:

```
>> r=1:10;
>> [out1,out2]=sumprod(r)

out1 =
    55

out2 =
  3628800
```

The status bar at the bottom indicates "Start" and "OVR".




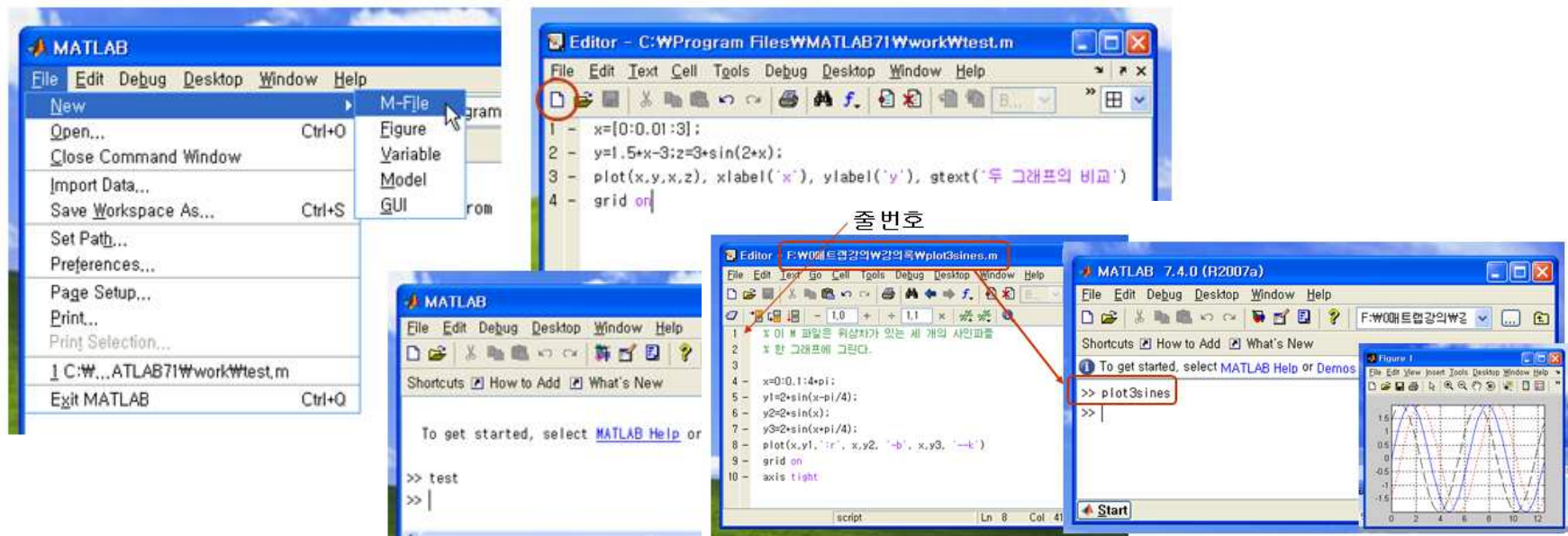
m-files : series of matlab commands

functions : input, output capability

mat-files : data storage/ loading

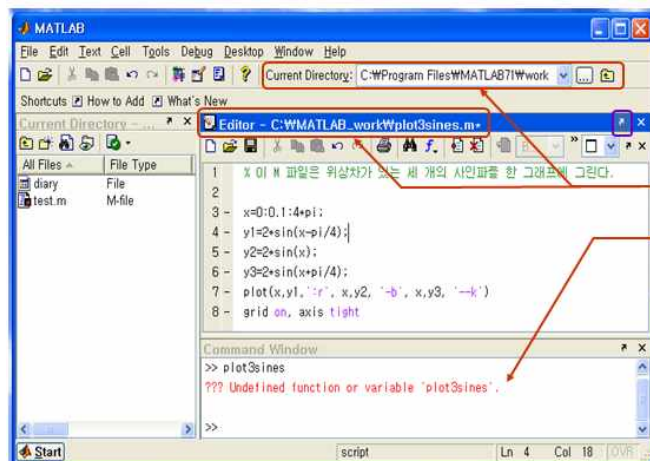
MATLAB m 파일의 생성

- MATLAB 메뉴의 File→New→M-File을 선택하거나,
- 메뉴 밑의 Toolbar에서 아이콘  을 선택하면,
- Editor 창 실행
 - Editor 창에서 프로그램을 작성하고 저장
 - (예) test.m
 - 명령어 창에서 >> test
 - 프로그램이 수행




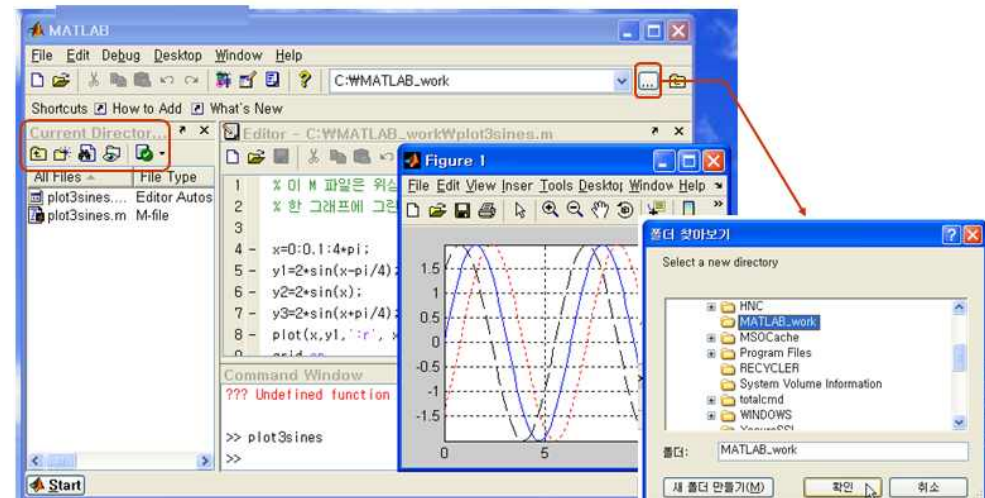
MATLAB m 파일의 생성

- 명령창에서 **>> edit 파일명 실행**
 - editor가 실행되면서 파일 열림
- Editor의 dock 아이콘 클릭
 - Editor가 MATLAB 데스크탑 환경에 통합
 - undock 아이콘 클릭 : 원상 복귀

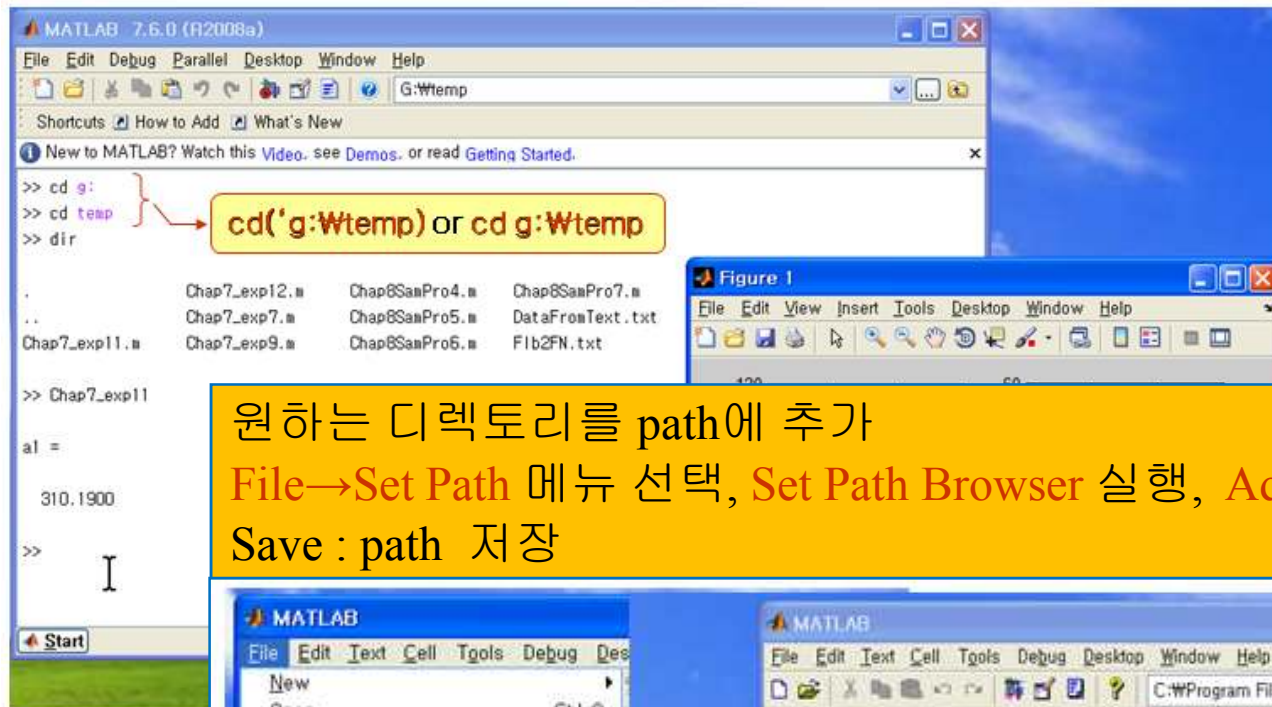


파일이 저장된 디렉토리와 현재 디렉토리가 다르면, M 파일이 실행되지 않고 에러메시지가 나온다.

❖ 폴더 찾아보기 아이콘  이나 데스크탑의 “Current Directory”를 이용하여 현재 디렉토리를 파일이 저장된 디렉토리로 변경한 후 실행한다.

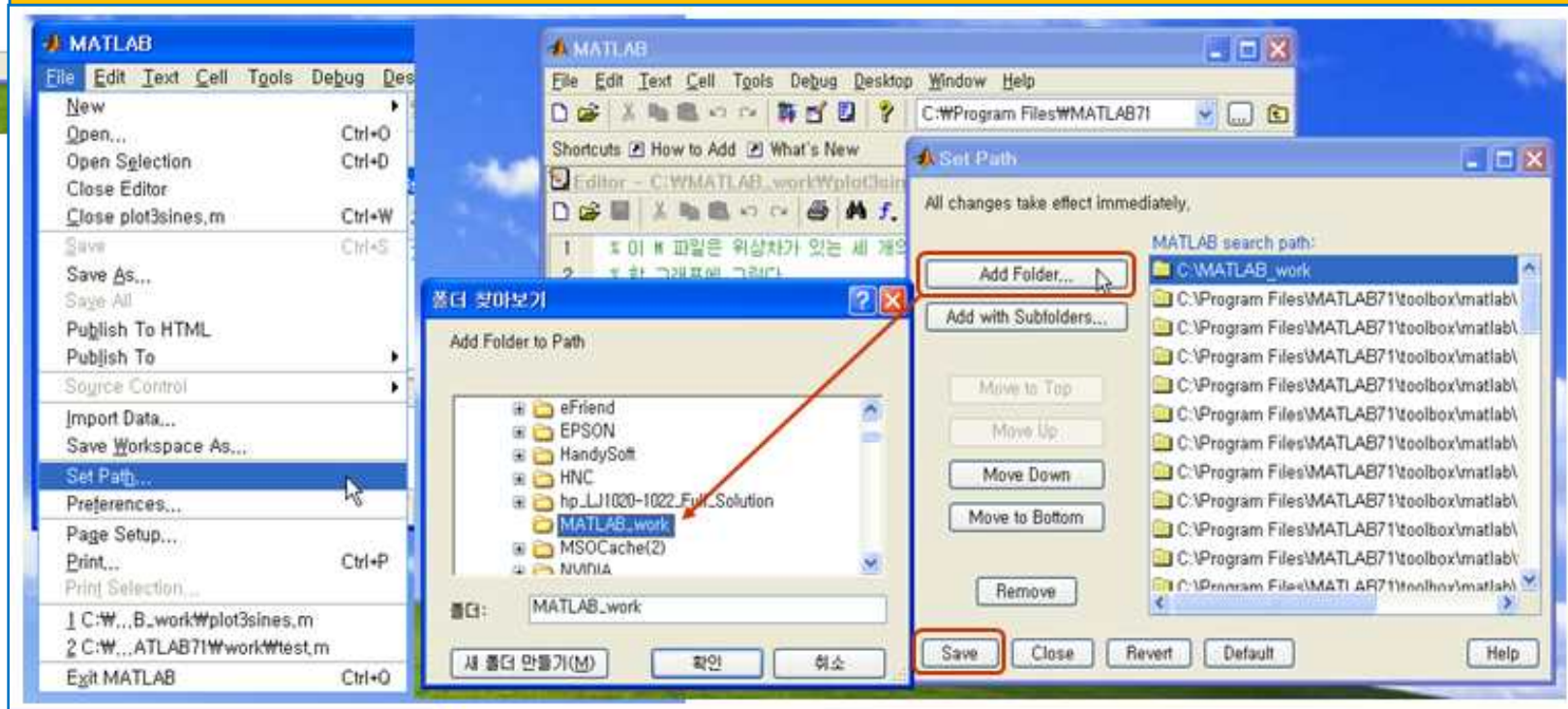


❖ 명령어 창에서 **cd** 명령어를 이용하여 디렉토리를 바꿀 수도 있다.



원하는 디렉토리를 path에 추가

File→Set Path 메뉴 선택, Set Path Browser 실행, Add Folder 이용 폴더 선택
Save : path 저장



스크립트 파일 (Flow Control)

- if

- ❖ if
- ❖ for
- ❖ while
- ❖ break
- ❖

| | | |
|---|--|--|
| <pre>if 식 문장 1 : 문장 n end</pre> | <pre>if 식 문장 1 else 문장 2 end</pre> | <pre>if 식1 문장 1 elseif 식2 문장 2 else 문장 3 end</pre> |
|---|--|--|

```
n=input('양의 정수를 입력하세요. :')
```

```
if rem(n,2) == 0
    disp('n은 짝수이다.')
else
    disp('n은 홀수이다.')
```

스크립트 파일(Flow Control)

- switch : 다중 택일 제어

```
switch 변수 또는 식
    case 값1          % 변수 또는 식의 값이 '값 1'인 경우
        문장 1
    case 값2          % 변수 또는 식의 값이 '값 2'인 경우
        문장 2
    :
    otherwise         % 변수 또는 식의 값이
                    % 위의 어떠한 case 값과도 일치하지 않는 경우
        문장 n
end
```

```
n=input('양의 정수를 입력하세요. :')
num=rem(n,2);

switch num
    case 0
        disp('n은 짝수이다.')
    otherwise
        disp('n은 홀수이다.')
end
```

스크립트 파일(Flow Control)

- for
 - 반복 제어

```
for 첨자=초기값 : 증분값 : 최종값  
    문장  
end
```

```
sum=0 ;  
for i=1 : 100  
    sum=sum + i ;  
end
```

스크립트 파일(Flow Control)

- while
 - 반복 제어

```
while 조건
    문장
end
```

```
i=1 ; sum=0 ;
while i <=100
    sum=sum + i ;
    i=i+1 ;
end
```

THE END

Homework/Report

- Matlab 설치 및 PPT 예제 실습