



FMBA 2018 사전 교육

프로그래밍(MATLAB)

KAIST

경영공학과 석박사통합과정 유승현

rambor12@business.kaist.ac.kr

목 차

1. 개요
2. MATLAB 설치
3. MATLAB 인터페이스
4. MATLAB 연산
5. MATLAB 프로그래밍
6. MATLAB 응용 : Black–Scholes Option Pricing

Github Repository

□ 프로그래밍 교육 자료

https://github.com/Seung-hyeon/KAIST_FMBA_Programming_2018

This repository Search Pull requests Issues Marketplace Explore

Seung-hyeon / KAIST_FMBA_Programming_2018 Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

KAIST FMBA Programming Prior-Education Materials Edit

Add topics

5 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Seung-hyeon some files		Latest commit 871ad68 2 minutes ago
Excel VBA	some files	2 minutes ago
Matlab	some files	2 minutes ago
R	some files	2 minutes ago
README.md	some files	2 minutes ago

README.md

KAIST FMBA Programming 2018

This is KAIST FMBA Programming Prior-Education Materials.

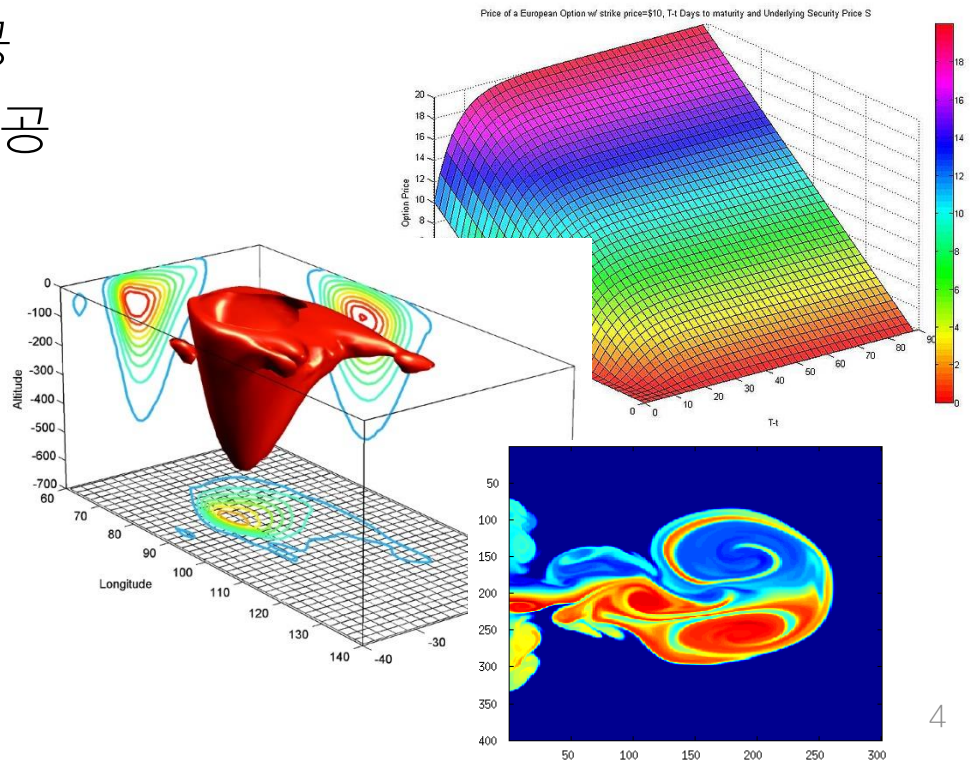
This materials cover Excel VBA , Matlab , and R .

자료 다운로드

1. 개요

□ MATLAB (MATrix LABoratory)

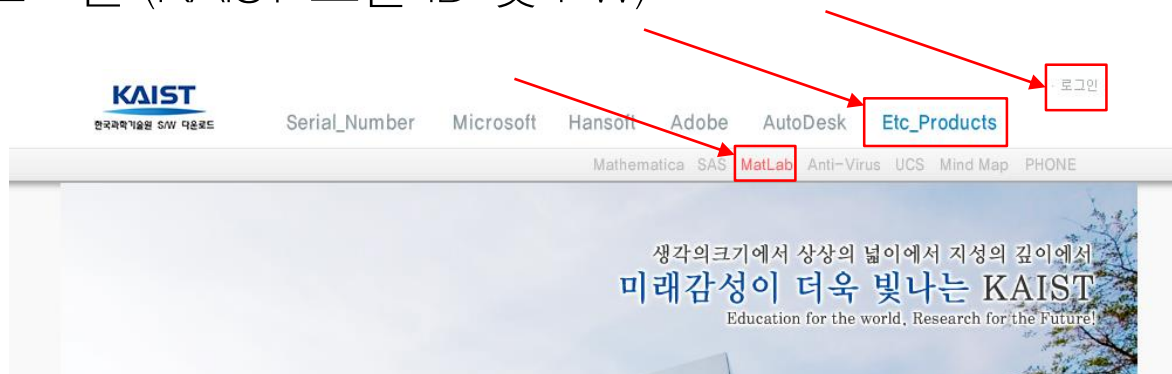
- ✓ MathWorks사가 유료로 제공하는 수치해석 소프트웨어 (2017년 1월 Concurrent 라이선스 기준 1천 6백만원)
- ✓ 사용자의 입력에 대한 결과 즉시 확인 가능 (VBA와 다름)
- ✓ 강력한 시각화 그래프 제공
- ✓ 다양한 툴박스(ToolBox) 제공
 - Control System Toolbox
 - Signal Processing Toolbox
 - Image Processing Toolbox
 - Wavelet Toolbox
- ✓ 주로 공대에서 많이 사용



2. MATLAB 설치

□ MATLAB 설치 방법

- ① KAIST 소프트웨어 다운로드 사이트 접속
<http://kftp.kaist.ac.kr/>
- ② 로그인 (KAIST 포탈 ID 및 PW)

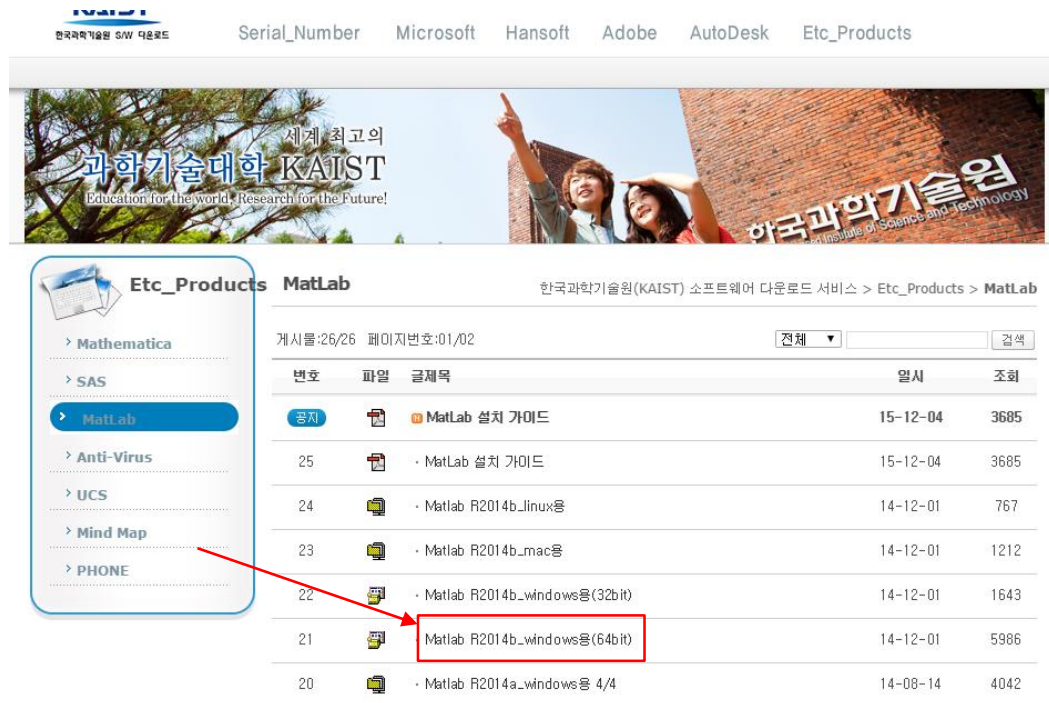


- ③ Etc_Products → Matlab 클릭

2. MATLAB 설치

□ MATLAB 설치 방법

④ 자신의 운영체제에 맞는 파일 선택하여 클릭



한국과학기술원 S/W 다운로드

Serial_Number Microsoft Hansoft Adobe AutoDesk Etc_Products

세계 최고의
과학기술대학 KAIST
Education for the world, Research for the Future!

한국과학기술원
Korea Institute of Science and Technology

Etc_Products MatLab

한국과학기술원(KAIST) 소프트웨어 다운로드 서비스 > Etc_Products > MatLab

게시물:26/26 페이지번호:01/02 전체 검색

번호	파일	글제목	일시	조회
	공지	MatLab 설치 가이드	15-12-04	3685
25		MatLab 설치 가이드	15-12-04	3685
24		Matlab R2014b_linux용	14-12-01	767
23		Matlab R2014b_mac용	14-12-01	1212
22		Matlab R2014b_windows용(32bit)	14-12-01	1643
21		Matlab R2014b_windows용(64bit)	14-12-01	5986
20		Matlab R2014a_windows용 4/4	14-08-14	4042

⑤ 첨부 파일(.exe) 다운로드

2. MATLAB 설치

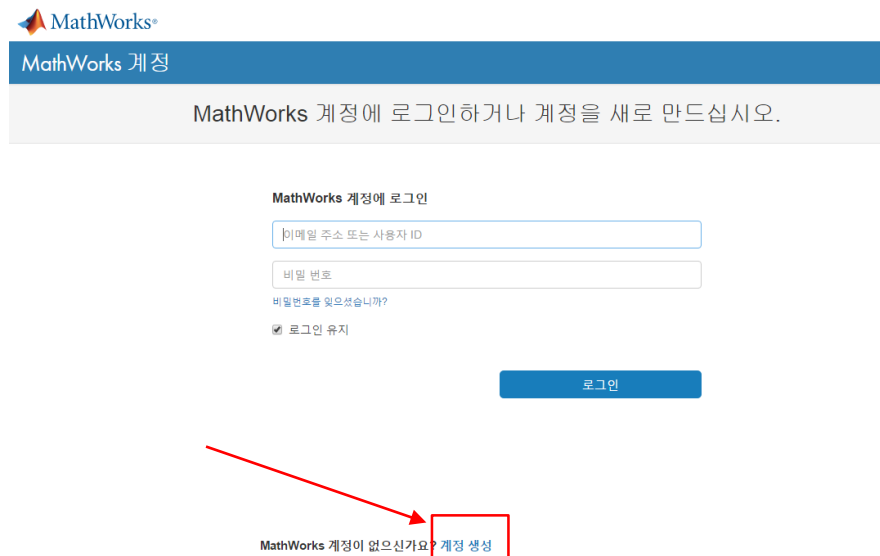
□ MATLAB 설치 방법

⑥ MathWorks 계정 생성 (kaist.ac.kr 이메일 필요!)

- <https://kr.mathworks.com/> 접속
- 우측 상단 '로그인' 클릭



- '계정 생성' 클릭



2. MATLAB 설치

□ MATLAB 설치 방법

⑥ MathWorks 계정 생성 (**kaist.ac.kr 이메일 꼭 필요!**)

- 이메일 주소 입력 및 아래와 같이 옵션 설정 (**중요!**)

MathWorks 계정 생성하기

이메일 주소 

이메일 주소가 맞는지 확인하십시오.

국가/지역

MathWorks 소프트웨어를 어떻게 사용하실 계획입니까?

13세 이상입니까? ☒ 예 ☐ 아니요

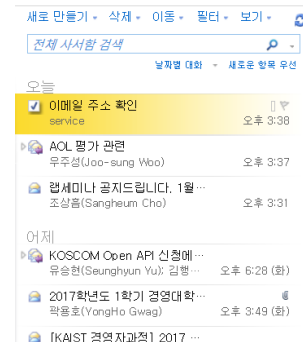
이메일 주소를 확인하십시오.

1. rambor12@business.kaist.ac.kr의 받은 편지함을 확인하십시오.
2. MathWorks에서 발송된 이메일의 링크를 클릭합니다.

이메일을 받지 못하셨습니까?

1. 스팸 폴더를 확인하십시오.
2. 이메일을 다시 보내 주십시오.
3. 추가적인 도움이 필요하실 경우, [이메일 확인](#)을 하시거나, [고객 지원팀에 문의](#) 주시기 바랍니다.

- 이메일 들어가서 메일 확인



이메일 주소 확인

service [service@mathworks.com]

받는 사람: 유승현(Seunghyun Yu)

MathWorks에 등록해주셔서 감사합니다.

등록 절차를 완료하려면 이 링크를 클릭하여 귀하의 이메일 주소를 확인해 주십시오.

[이메일 주소 확인하기](#)

감사합니다.
MathWorks 고객 서비스팀 드림

[개인 정보 취급 방침](#)

2. MATLAB 설치

□ MATLAB 설치 방법

- ⑥ MathWorks 계정 생성 (**kaist.ac.kr 이메일 꼭 필요!**)
- 프로파일 생성 및 라이선스 번호 입력

프로파일 생성을 위해,
rambor12@business.kaist.ac.kr에 대한 다음
의 정보를 기재해 주십시오.

성	<input type="text"/>
이름	<input type="text"/>
사용자 ID (옵션)	<input type="text"/>
	3~16자 사이의 영문자와 숫자의 조합으로 이루어져야 하고 문자로 시작해야 합니다(예: jsmith55). 추가 정보 자세히 알아보기
비밀 번호	<input type="password"/>
비밀 번호 확인	<input type="password"/>
귀하의 직무는 무엇입니까?	<input type="text" value="선택하세요"/>
부서	<input type="text" value="선택하세요"/>

연락처 정보(영어)

다음의 정보를 영어로 입력하십시오.

성 (영어)	<input type="text" value="Yu"/>
이름 (영어)	<input type="text" value="Seung Hyeon"/>

계정에 라이선스를 등록하십시오.

MATLAB을 이미 보유하고 있는 경우 여기에 활성화 키 또는 라이선스 번호를 입력하십시오. 이것으로 무엇을 할 수 있습니까?

활성화 키 또는 라이선스 번호 (옵션)	<input type="text" value="17925-02416-09128-40758-01531"/>
-----------------------	--

MATLAB 명령 창에 'license'를 입력하여 라이선스 번호를 조회 하시거나, 귀하의 MATLAB 관리자로부터 활성화 키를 받으십시오.

☒ 개인 정보 취급 방침을 읽었으며, 내용에 동의합니다.

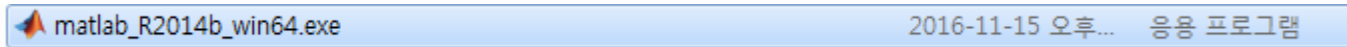
☐ 영업 담당자의 연락을 받고 싶습니다.

생성하기

2. MATLAB 설치

□ MATLAB 설치 방법

- ⑦ 아까 다운로드 받은 파일(.exe) 파일 '관리자 권한으로 실행'



- ⑧ 'MathWorks 계정으로 로그인하여 설치' 체크하고 '다음'

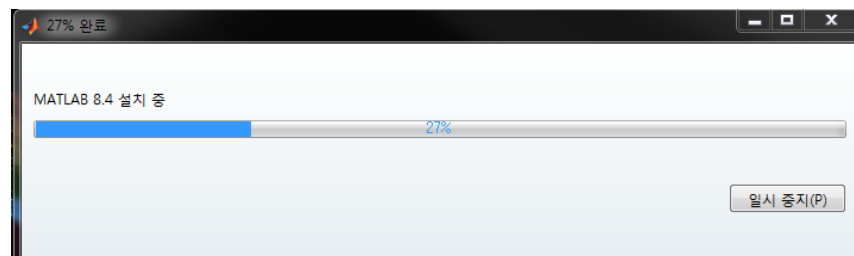
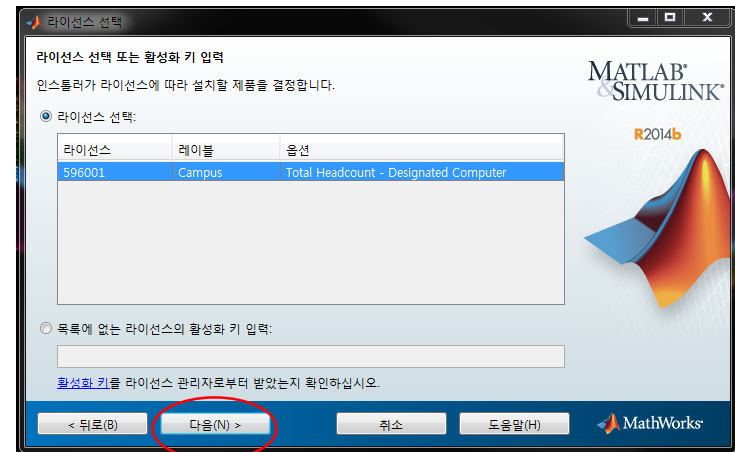
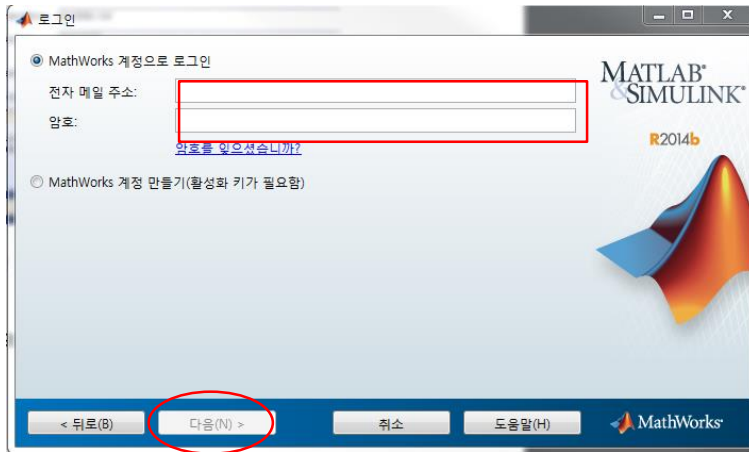
- ⑨ 약관 동의하고 다음



2. MATLAB 설치

□ MATLAB 설치 방법

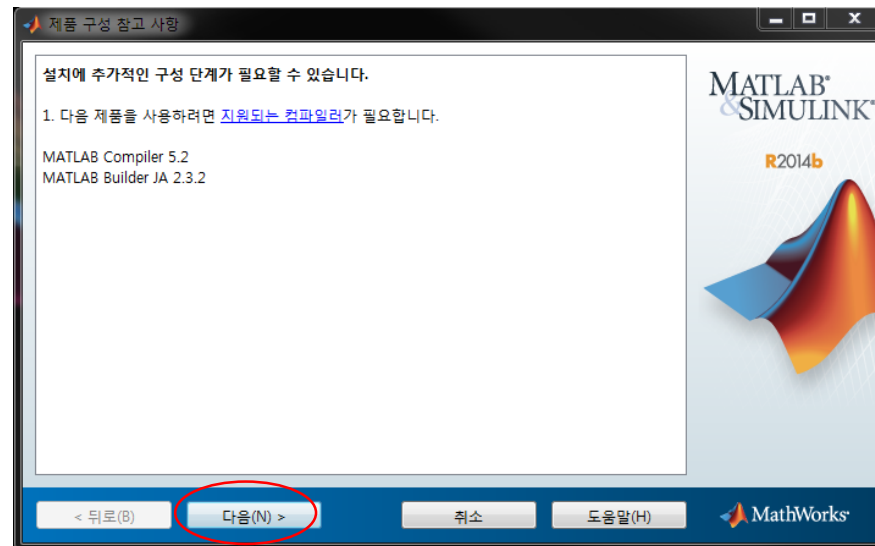
- ⑩ 방금 만든 MathWorks 계정 입력하고 다음
- ⑪ 자동으로 라이선스 확인되면 계속 다음 클릭하여 설치
(수 십분 소요)



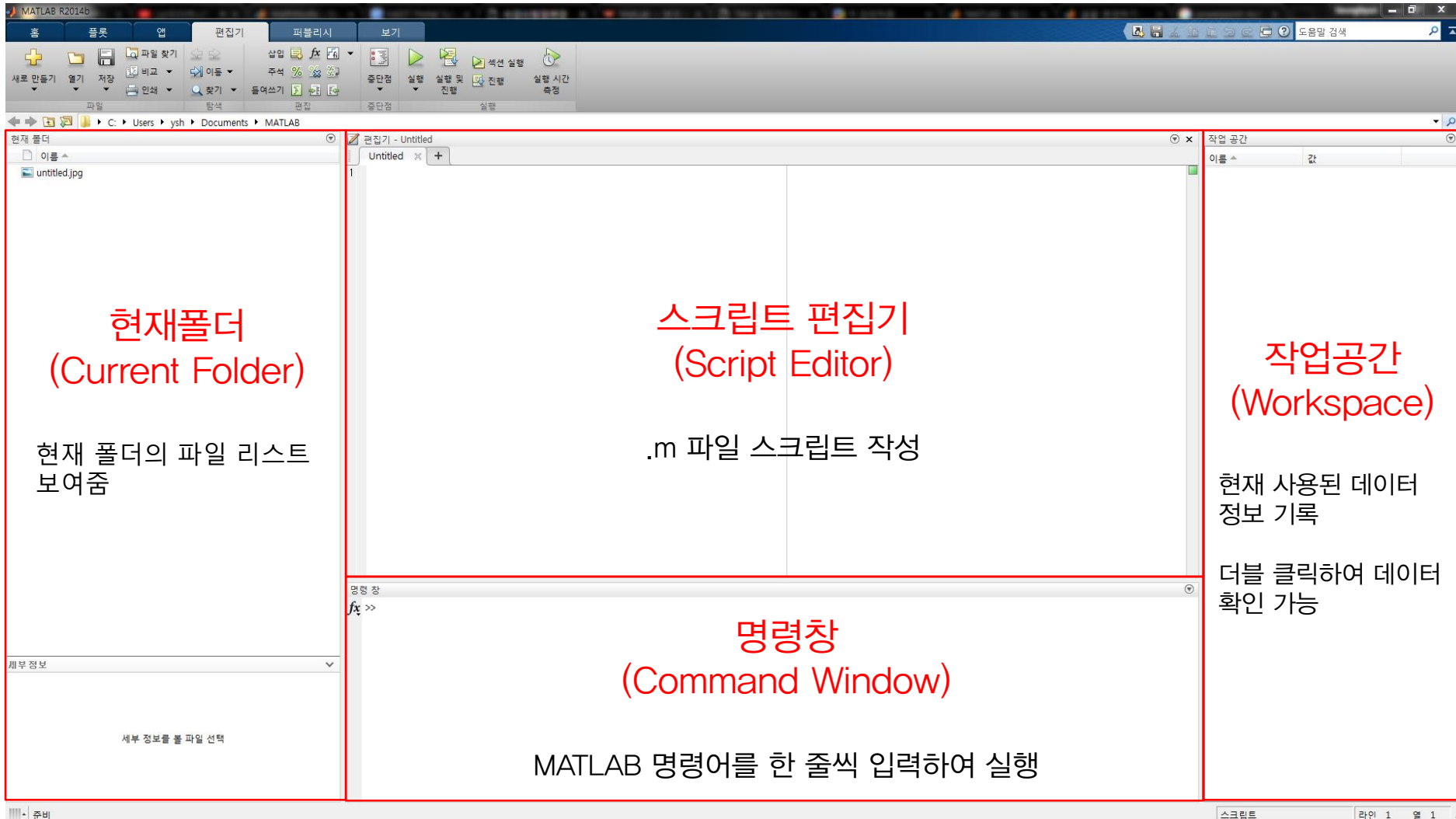
2. MATLAB 설치

□ MATLAB 설치 방법

- ⑩ 다음 클릭 및 설치 완료
- ⑪ 라이선스 활성화 실행
- ⑫ 시작 클릭하고 MATLAB 아이콘 클릭하여 실행 가능



3. MATLAB 인터페이스



3. MATLAB 인터페이스

□ **예제1** 다음 행렬의 곱을 구하여라.

$$\begin{pmatrix} 2 & 4 & 9 \\ 1 & 2 & 1 \\ 3 & 0 & 2 \end{pmatrix} \times \begin{pmatrix} 5 & 5 & 2 \\ 2 & 2 & 1 \\ 1 & 3 & 3 \end{pmatrix} = ?$$

$= A \qquad \qquad \qquad = B$


#MATLAB Code

```
% 행렬 A 생성
A = [2,4,9;1,2,1;3,0,2];

% 행렬 B 생성
B = [5,5,2;2,2,1;1,3,3];

% 행렬 곱 실행
A * B
```

행렬 생성



```
ans =

    27    45    35
    10    12     7
    17    21    12
```

```
>> |
```

3. MATLAB 인터페이스

□ 기본 명령어

세미콜론(;) : 실행값을 커맨드창에 표시하지 않기

```
% 커맨드창 깨끗하게 만들기
```

```
clc;
```

```
% MATLAB에서 사용하는 모든 변수와 함수 지우기
```

```
clear;
```

```
% MATLAB 도움말 이용하기
```

```
help [물어볼 대상]
```

```
% MATLAB 종료하기
```

```
quit
```

4. MATLAB 연산

□ 수식 표현

$$8 + \frac{6 \times (1 + \sqrt{5})}{2} \quad \gg 8 + (6 * (1 + \text{sqr t}(5))) / 2$$

$$\frac{4}{3} \times \pi \times 10^3 \quad \gg 4 / 3 * \text{pi} * 10^3$$

$$e^{\sqrt{3}} + \pi \times 10^3 \quad \gg \text{exp}(\text{sqr t}(3)) + \text{pi} * 10^3$$

$$e^{\pi \times \sqrt{-1}} \quad \gg \text{exp}(\text{pi} * \text{sqr t}(-1))$$

or

$$\gg \text{exp}(\text{pi} * i)$$

- ① π 는 pi로 기술
- ② 자연상수 e 는 exp(1)로 기술, e^x 는 exp(x)로 기술
- ③ i와 j는 허수 단위(imaginary number unit)

4. MATLAB 연산

벡터(배열) 생성

```
>> % 벡터(배열) 생성
```

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

```
a =
```

```
1    2    3
```

```
>> % 열(column) 벡터 생성
```

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

```
a =
```

```
1  
2  
3
```

벡터 변환(Transpose) : 프라임 기호(') 붙이기

```
>> % 벡터 transpose
```

```
>> a'
```

```
ans =
```

```
1    2    3
```

벡터 원소 참조 괄호() 사용

```
>> % 벡터 원소 참조
```

```
>> a(1)
```

```
ans =
```

```
1
```

```
>> a(2)
```

```
>> % 여러 원소 참조
```

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

```
ans =
```

```
ans =
```

```
2    3
```

```
2
```

다양한 원소 추출 방법

```
>> % 1. true, false를 이용한 추출
```

```
>> a([true,false,true])
```

```
ans =
```

```
1    3
```

```
>> a([false,true,false])
```

```
ans =
```

```
2
```

```
>> % 2. 인덱스 번호 배열을 이용한 추출
```

```
>> a([1,3])
```

```
ans =
```

```
1    3
```

```
>> % 3. 논리 조건을 이용한 추출
```

```
>> a(a/2+1 > 2)
```

```
ans =
```

```
3
```

```
>> a(mod(a,3) == 1)
```

```
ans =
```

```
1
```

4. MATLAB 연산

순차적인 벡터 생성

```
>> % 순차적인 벡터 생성  
>> a = 1:3
```

```
a =
```

```
1     2     3
```

```
>> % 간격이 정해져있는 벡터  
>> a = 1:0.2:2
```

```
a =
```

```
1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
```

간격의 길이



4. MATLAB 연산

□ 단일 벡터 연산

```
>> exp(a)

ans =

    2.7183    7.3891   20.0855

>> log(a)

ans =

    0    0.6931    1.0986

>> sqrt(a)

ans =

    1.0000    1.4142    1.7321

>> abs(a)

ans =

    1    2    3

>> a.^2

ans =

    1    4    9
```

각 벡터의 원소 x 에 대해
 $e^{\sqrt{3}x} + \pi \times 10^{-1} + x$ 계산

```
>> exp(sqrt(3).+a)+pi*10^(-1) + a

ans =

    6.9664   34.2619  183.8903
```

주의!
벡터 계산할 때 곱셈, 나눗셈, 거듭제곱에선
점(.)을 사용해야함!

기타 : min, max, sum, mean, var, std 등도 가능!

4. MATLAB 연산

□ 벡터 간 연산

```
>> a = [1,2,3];  
>> b = [4,5,6];  
>>  
>> % 벡터의 합과 차  
>> a + b
```

```
ans =  
  
5    7    9
```

```
>> a - b
```

```
ans =  
  
-3   -3   -3
```

```
>> % 벡터의 내적  
>> dot(a,b); %or  
>> a*b'
```

```
ans =
```

32

```
>> % 벡터의 외적(Cross Product)  
>> cross(a, b)
```

```
ans =  
  
-3    6   -3
```

```
>> % 벡터의 외적(Outer Product)  
>> a'*b
```

```
ans =  
  
4    5    6  
8   10   12  
12   15   18
```

```
>> % 원소간 곱셈  
>> a.*b
```

```
ans =  
  
4    10   18
```

```
>> % 원소간 나눗셈  
>> a./b
```

```
ans =  
  
0.2500    0.4000    0.5000
```

4. MATLAB 연산

□ 행렬 생성

행렬 생성

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

M =

1	2	3
4	5	6
7	8	9

```
>> % 방법 2  
>> r1 = [1,2,3];  
>> r2 = [4,5,6];  
>> r3 = [7,8,9];  
>> M = [r1; r2; r3]
```

M =

1	2	3
4	5	6
7	8	9

특수 행렬 생성

```
>> zeros(m,n); % 모든 원소가 0인 mXn 행렬 생성  
>>  
>> ones(m,n); % 모든 원소가 1인 mXn 행렬 생성  
>>  
>> eye(m); % 대각선 원소가 1로 구성된 mXm 정방행렬 생성  
>>  
>> rand(m,n); % 0~1 사이의 uniform 분포를 가지는 mXn 행렬 생성  
>>  
>> randn(m,n); % 표준정규분포를 가지는 mXn 행렬 생성
```

```
>> % 행렬 원소 참조  
>> M(1,3)
```

ans =

3

→ 인덱스는 1부터 시작

4. MATLAB 연산

□ 행렬 연산

→ 벡터 연산과 동일

역행렬 계산

```
>> % 역행렬 계산
```

```
>> inv(M)
```

경고: 행렬이 특이 행렬에 가깝거나 준특이 행렬(badly scaled)일 수 있습니다. 결과가 부정확할 수 있습니다. RCOND = 1.541976e-18.

```
ans =
```

```
1.0e+16 *
```

```
-0.4504    0.9007   -0.4504  
 0.9007   -1.8014    0.9007  
-0.4504    0.9007   -0.4504
```

4. MATLAB 연산 : 익명함수

- 익명함수(Anonymous function)
간단하게 만들어 사용하기 위한 함수

$$h(x) = gx$$

```
% 익명함수(Anonymous function)
```

```
g = 10;  
h = @(x) g*x;
```

```
integral(h, 1, 10)
```


$$f(x) = \sin(\alpha x)$$

```
alpha = 0.9;  
f = @(x) sin(alpha*x);
```

```
x = 0:pi/100:pi;  
area(x, f(x));
```

4. MATLAB 연산

- MATLAB 간단한 해찾기 (roots 함수)

$$x^2 - 8x + 12 = 0$$

$$\text{poly} = [1 \quad -8 \quad 12]$$

roots(poly)

```
>> poly = [1 -8 12];  
>> roots(poly)
```

```
ans =
```

```
6
```

```
2
```

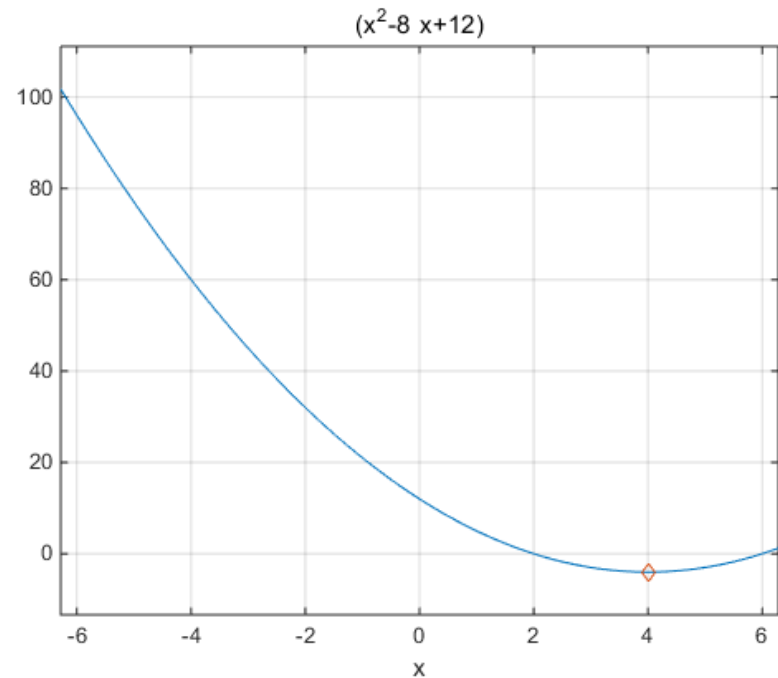

4. MATLAB 연산

- MATLAB 최소값 구하기(fminbnd 함수)

$$x^2 - 8x + 12 = 0$$

% 최소값 구하기

```
f = @(x) (x.^2 - 8*x + 12);  
ezplot(f);  
hold on;  
  
minimum = fminbnd(f, -100,100);  
plot(minimum, f(minimum), 'd');  
  
grid;  
hold off;
```



4. MATLAB 연산

- 벡터, 행렬 계산 및 간단한 그래프 실습



- 참고

<https://kr.mathworks.com/products/matlab/examples.html>

5. MATLAB 프로그래밍

□ M-file

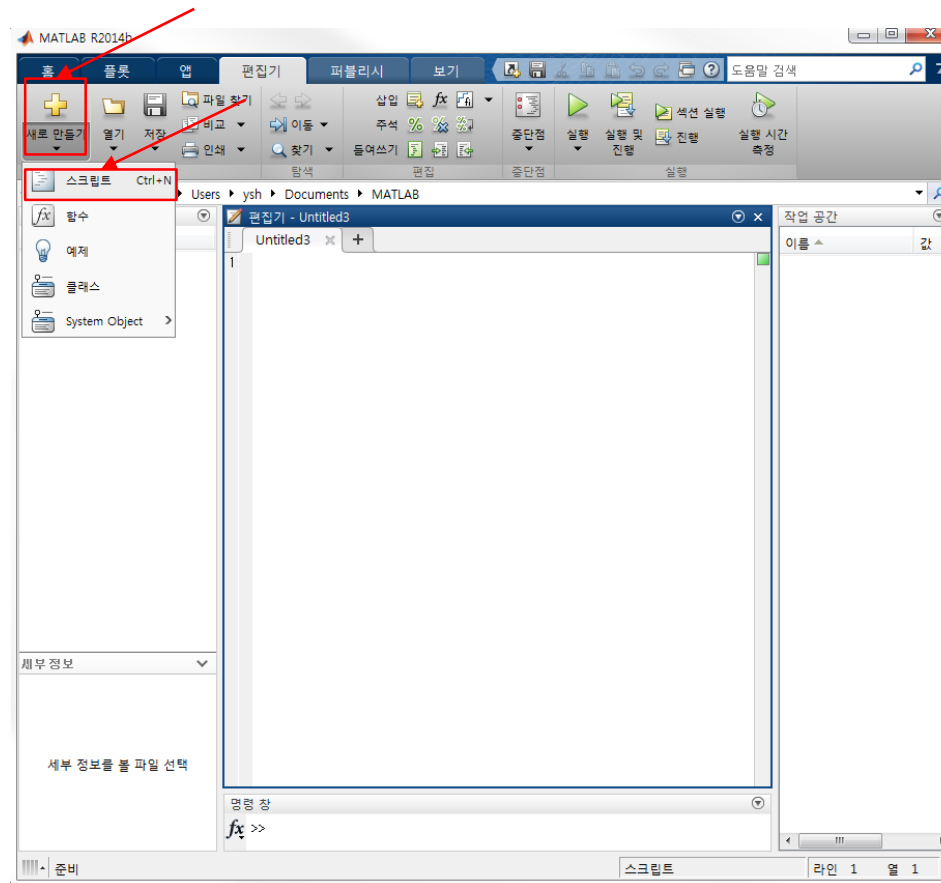
- ① MATLAB 명령어들을 포함하는 텍스트 파일
- ② 명령어가 너무 길거나 반복적으로 써야할 때 사용
- ③ 파일의 확장자는 반드시 ".m"이어야 함
- ④ 종류
 - 스크립트 파일 : 함수를 사용한 연속적인 명령어들의 모임
 - 함수 파일 : 함수를 정의해 놓은 파일

이름	수정한 날짜	유형	크기
 script.m	2017-01-12 오후...	MATLAB Code	1KB
 myfunction.m	2017-01-12 오후...	MATLAB Code	1KB

5. MATLAB 프로그래밍

□ 스크립트 M-file

"새로 만들기" → "스크립트"를 클릭하여 만들 수 있음

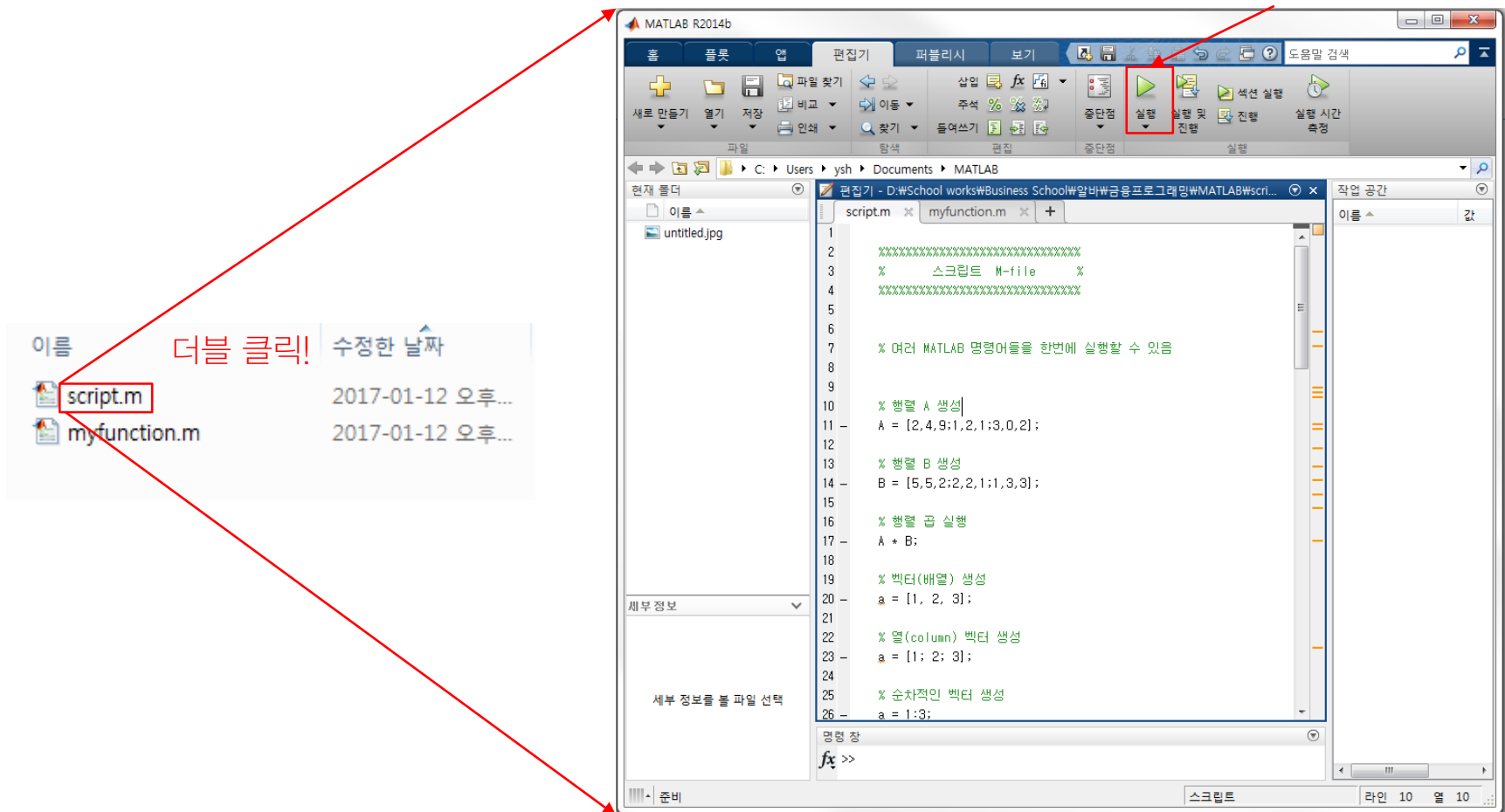


5. MATLAB 프로그래밍

□ 스크립트 M-file : 파일을 열어 "실행" 버튼 눌러서 실행.

① 실행하면 적힌 명령어들이 순서대로 실행

실행 버튼 클릭!

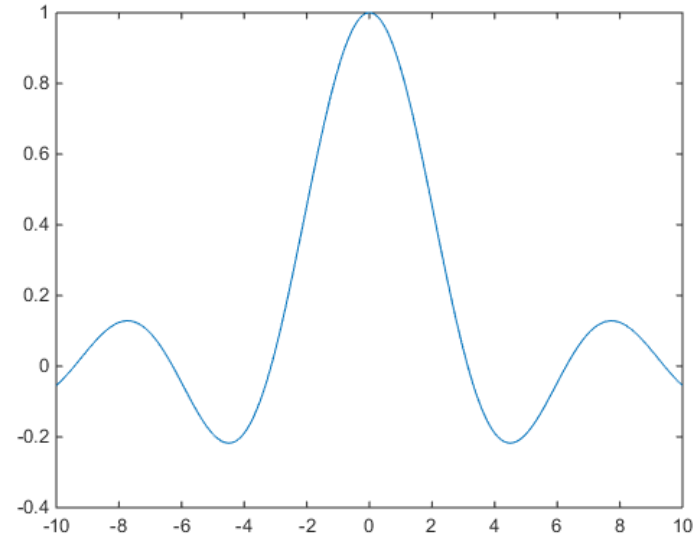


5. MATLAB 프로그래밍

□ **예제2** $\sin(x)/x$ 그래프를 그려라.

#Answer

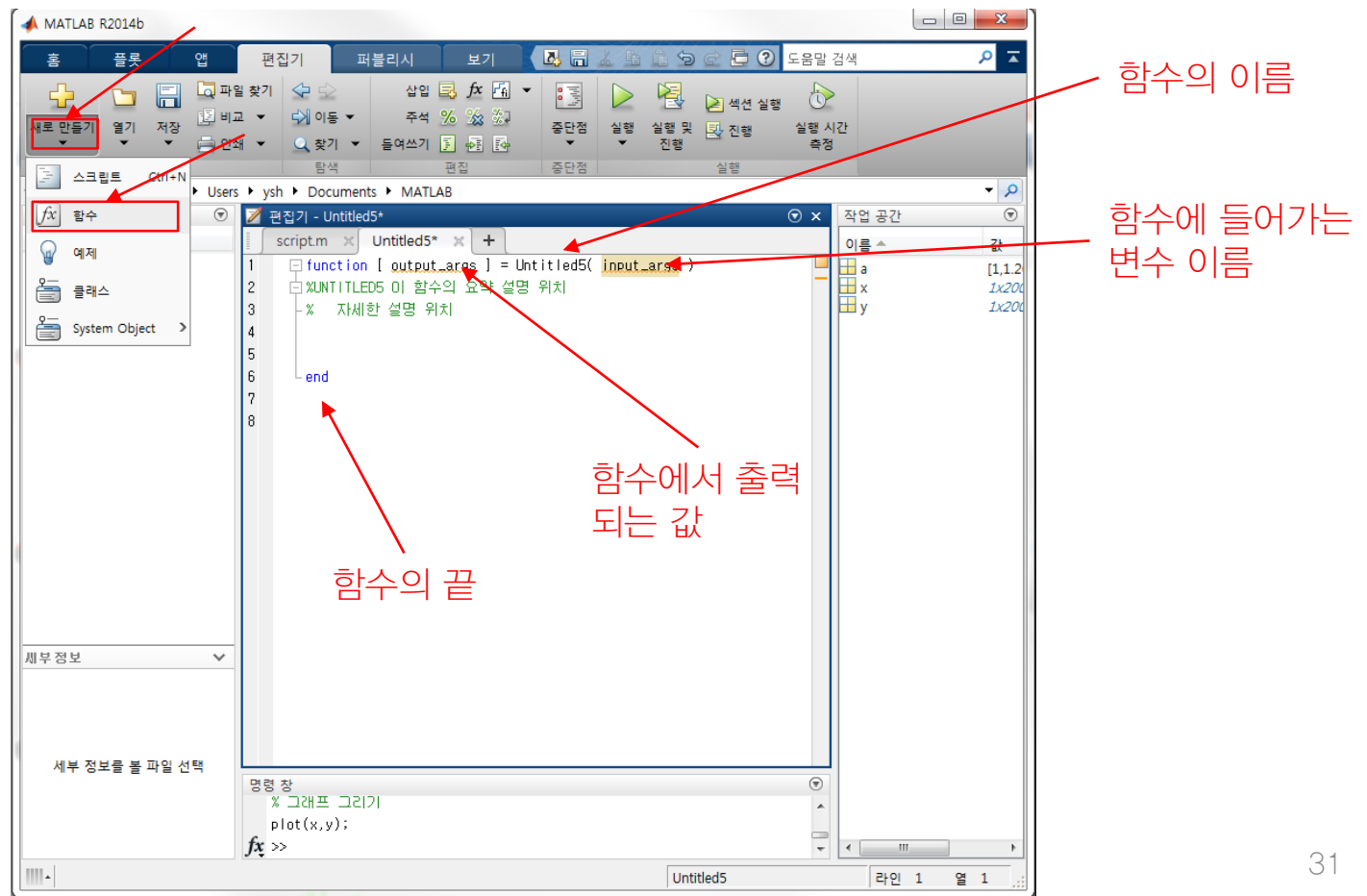
```
% x 값을 -10부터 10까지 0.01 간격으로 생성  
x = -10:0.01:10;  
  
% x 값에 대응되는 y = sin(x)/x 를 생성  
y = sin(x) ./ x;  
  
% 그래프 그리기  
plot(x,y);
```



5. MATLAB 프로그래밍

□ 함수 M-file

"새로 만들기" → "함수"를 클릭하여 만들 수 있음



5. MATLAB 프로그래밍

□ 함수 M-file

- ① 함수 이름 지정(또한 함수 이름과 파일 이름이 같아야 함!)
- ② 다른 스크립트 파일에서 함수를 호출하여 실행

예제2 $\sin(x)/x$ 그리는 'myfunction' 함수를 만들고 실행하라.

myfunction.m

```
function y = myfunction(x)
% sin(x)/x 그리기
%   자세한 설명 위치

    y = sin(x) ./ x;
    plot(x,y)

end
```

Ex2_sinPlot.m

```
% x 값을 -10부터 10까지 0.01 간격으로 생성
x = -10:0.01:10;

% myfunction 호출
y = myfunction(x);
```


5. MATLAB 프로그래밍

□ MATLAB 기초 문법

1. 변수의 선언 : (Declaration)

숫자, 벡터, 행렬의 생성 => 이미 했음.

5. MATLAB 프로그래밍

□ MATLAB 기초 문법

2. 조건문 : (Selection)

조건에 따라 어떤 코드를 실행할 것인가?

if [조건1]
실행 조건 ← [명령문]; → 실행할 명령문
elseif [조건2]
[명령문];
else
[명령문]
end → If 문 종료

```
% Example
a = 1;
if a < 0
    b = -1;
elseif a > 1
    b = 1;
else
    b = 0;
end

% 실행하면 b=0
```

5. MATLAB 프로그래밍

□ MATLAB 기초 문법

3. 반복문 : (Iteration)

이 코드를 언제까지 반복할 것인가?

반복 조건

```
% for 문 : 반복적인 작업 수행  
for 변수 = 시작값 : 간격 : 종료값  
    [명령문] → 실행할 명령문  
    if [탈출 조건]  
        break  
    end  
end
```

```
% Example  
j = 0  
for i = 0 : 2 : 100  
    j = j + i  
    if i > 4  
        break  
    end  
end  
  
% 실행하면 j=12 이 실행됨
```

5. MATLAB 프로그래밍

□ MATLAB 기초 문법

3. 반복문 : (Iteration)

이 코드를 언제까지 반복할 것인가?

% while 문 : 반복회수가 정해지지 않을 때 사용

while [반복조건]

[명령문]; → 실행할 명령문

end

반복 조건 : 이 조건이 맞지 않으면 탈출함

% Example (for 문과 정확히 동등한 예)

i = 0;

j = 0;

while i<=5

j = j + i;

i = i + 1;

end

% 실행하면 j=15 이 실행됨

5. MATLAB 프로그래밍

- 반복문 예제 : 피보나치 수열 구하기

$$f(n) = f(n - 1) + f(n - 2)$$

% 반복문 예제 : 피보나치 수열 첫번째부터 100번째까지 구하기

f(1) = 1;

f(2) = 1;

for i = 3:100

 f(i) = f(i-1) + f(i-2);

 disp(f(i))

end

% 그래프 그리기

plot(f)

5. MATLAB 프로그래밍

□ MATLAB 그래프 그리기

2차원 그래프

```
% x값 벡터로 설정
```

```
x = -50:0.01:50;
```

```
% x에 따른 y값 계산
```

```
y = sin(x)./x;
```

```
% 그래프 그리기
```

```
plot(x, y);
```

x값 y값

```
% 그래프 제목
```

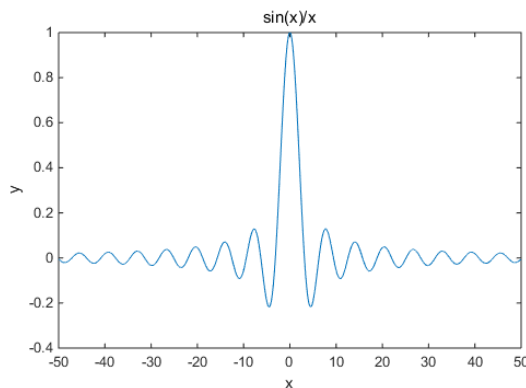
```
title('sin(x)/x')
```

```
% x축 이름
```

```
xlabel('x')
```

```
% y축 이름
```

```
ylabel('y')
```



예제3 3차원 곡면 그래프

```
% x값 벡터로 설정
```

```
x = -20:0.2:20;
```

```
% y값 벡터로 설정
```

```
y = -20:0.2:20;
```

```
% x, y 그리드 만들기
```

```
[x, y] = meshgrid(x, y);
```

중요!

그리드 만들기!

```
% x, y에 따른 z값 계산
```

```
r = sqrt(x.^2 + y.^2);
```

```
z = sin(r) ./ r;
```

```
% 그래프 그리기
```

```
surf(x, y, z);
```

```
% 그래프 제목
```

```
title('sin(r)/r')
```

```
% x축 이름
```

```
xlabel('x')
```

```
% y축 이름
```

```
ylabel('y')
```

```
% z축 이름
```

```
zlabel('z')
```

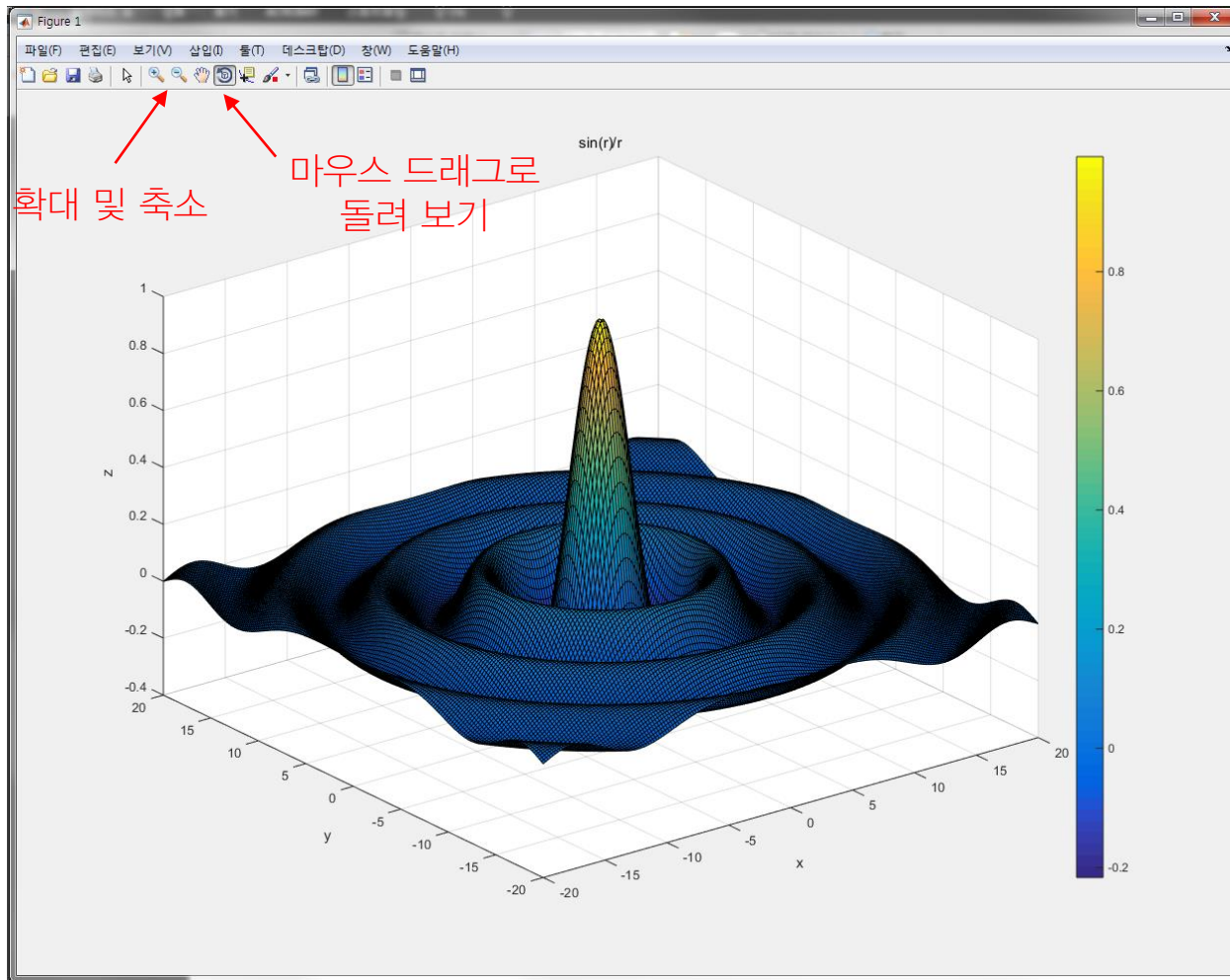
```
% 컬러바 표시
```

```
colorbar()
```

5. MATLAB 프로그래밍

□ MATLAB 그래프 그리기

3차원 곡면 그래프



5. MATLAB 프로그래밍

□ MATLAB 그래프 그리기

3차원 곡면 그래프

- ① 메시 플롯(mesh plot)
- ② 곡면 플롯(surf plot)
- ③ 곡면 플롯 셰이딩 포함
- ④ 등고선 플롯(contour plot)

6. MATLAB 응용 : Black-Scholes Option Pricing

□ Black-Scholes(-Merton) Model

- ✓ Fisher Black, Myron Scholes, (Robert C. Merton)이 1973년 제시
- ✓ 1997년에 노벨 경제학상
- ✓ 세상을 바꾼 10대 방정식 (출처 : <http://biz.heraldcorp.com/view.php?ud=20160308000285>)
- ✓ 파생상품(특히 European Option)의 가격 결정 편미분방정식
- ✓ 옵션(Option) : 미래 특정 시점(T)에 특정 상품(S)을 정해진 가격(K)에 사고(call) 팔(put) 권리

옵션 가격

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0$$

기초 자산 변동성

무위험 이자율

기초 자산 가격

6. MATLAB 응용 : Black-Scholes Option Pricing

□ Black-Scholes(-Merton) Model

옵션 가격

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0$$

기초 자산 변동성

무위험 이자율

기초 자산 가격

방정식의 해(Solution)

표준정규분포 누적 함수

$$\text{Call Price} = SN(d_1) - Ke^{-r\tau} N(d_2)$$

$$\text{Put Price} = Ke^{-r\tau} \{1 - N(d_2)\} - S \{1 - N(d_1)\}$$

만기까지 남은 기간

$$\text{where, } d_1 = \frac{\ln S/K + (r + \frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

6. MATLAB 응용 : Black–Scholes Option Pricing

□ 예제4 Black–Scholes Option Pricing & Plot

변수 설명	변수 이름	값
기초 자산 가격	S	0:0.1:20
옵션 행사 가격	K	10
무위험 이자율	r	0.1
만기까지 남은 기간	tau	0:1:90
기초자산의 변동성	sigma	0.3

6. MATLAB 응용 : Black–Scholes Option Pricing

□ **예제4** Black–Scholes Option Pricing & Plot (Sample Code)

BSPrice.m

```
function BSPrice = BSPrice(S, K, r, tau, sigma, call_put)
% Sample Code
% 자세한 설명 위치

d1 = (log(S ./ K) + (r + 0.5 * sigma.^2) .* tau) ./ (sigma .* sqrt(tau));
d2 = d1 - sigma .* sqrt(tau);

Nd1 = normcdf(d1);
Nd2 = normcdf(d2);

if strcmp(call_put, 'call')
    BSPrice = S .* Nd1 - K .* exp(-r .* tau) .* Nd2;

elseif strcmp(call_put, 'put')
    BSPrice = K .* exp(-r .* tau) .* (1 - Nd2) - S .* (1 - Nd1);

end

end
```

Ex4_BSPrice_script.m

```
% BS pricing Script %

S = 0:0.1:20;
K = 10;
r = 0.1;
tau = 0:90;
sigma = 0.3;
call_put = 'call';

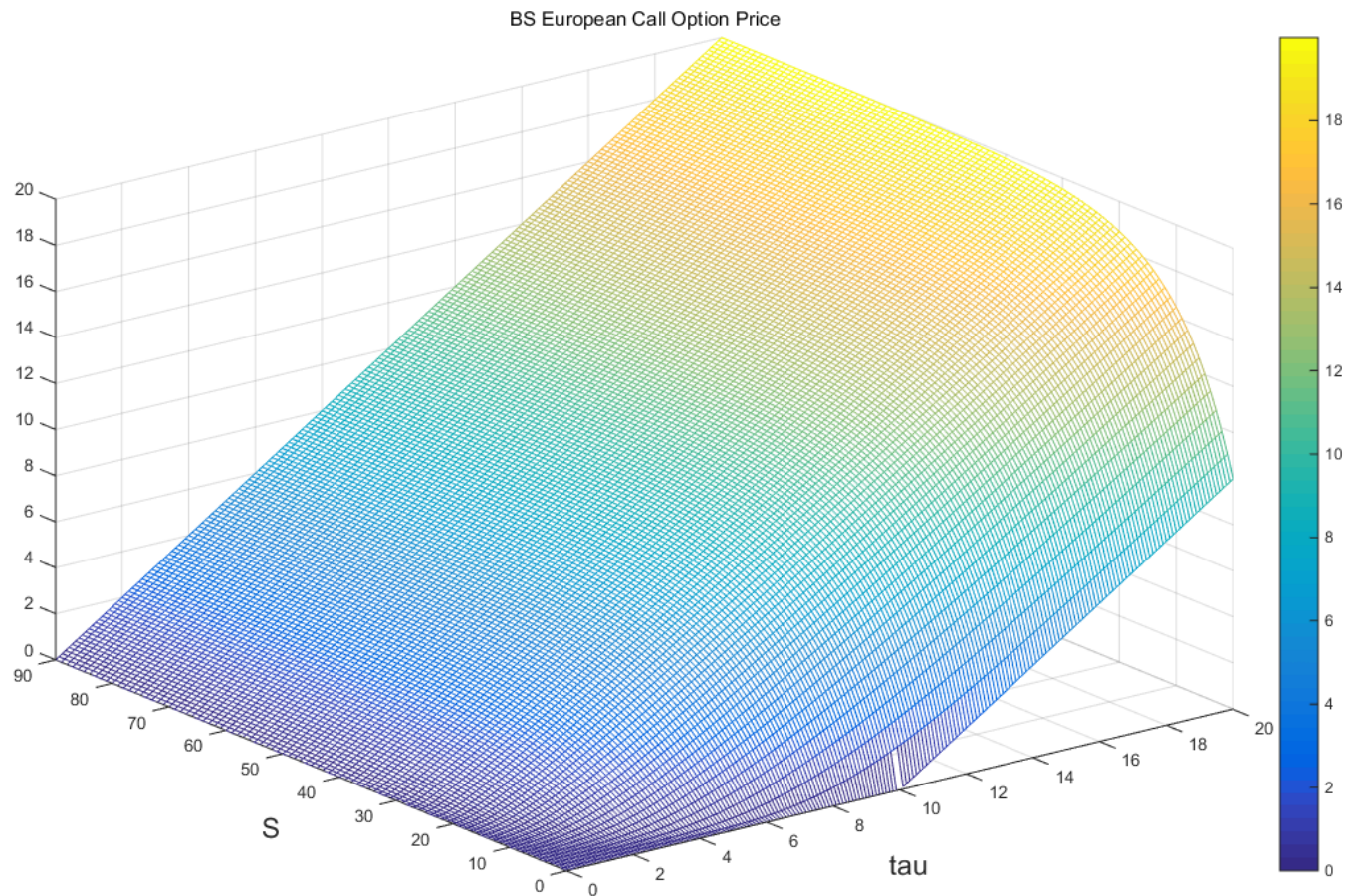
% 차원 맞춰주기 : 둘 다 같은 크기의 행렬로 만들기
[S_, tau_] = meshgrid(S, tau);

% Black Scholes Pricing
p = BSPrice(S_, K, r, tau_, sigma, call_put);

% 그래프 그리기
mesh(S_, tau_, p);
% 그래프 제목
title('BS European Call Option Price');
% x축 이름
xlabel('tau', 'FontSize', 16);
% y축 이름
ylabel('S', 'FontSize', 16);
% 컬러바 표시
colorbar();
```

6. MATLAB 응용 : Black–Scholes Option Pricing

□ 예제4 Black–Scholes Option Pricing & Plot



6. MATLAB 응용 : Geometric Brownian Motion

□ Black–Scholes(–Merton) Model

- ✓ 기본 가정 : "기초 자산의 가격은 Geometric Brownian Motion을 따른다."

기초 자산 가격 \rightarrow

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW_t$$

μ \rightarrow 평균 수익률

σ \rightarrow 기초 자산 변동성

dW_t \rightarrow Brownian Motion

6. MATLAB 응용 : Geometric Brownian Motion

□ Black–Scholes(–Merton) Model

✓ 기본 가정 : "기초 자산의 가격은 Geometric Brownian Motion을 따른다."

✓ **예제5** : Brownian Motion Plot

✓ **예제6** : Geometric Brownian Price Plot

% Normal random Variable Plot

```
for i=1:100
    y(i) = randn();
end

plot(y)
```

% 1-D Brownian Motion Animation

```
for t=1:N
    plot(1:t, y(1:t), 'r');
    grid on;
    axis([0 N -5 5])
    hold on;
    drawnow;
end
```

% 2D Brownian Motion Plot

```
N = 1000;
sigma = 0.1;
```

```
x = zeros(1, N);
y = zeros(1, N);
```

```
for j = 1:N
    for i = 1:N
        x(i+1, j) = x(i, j) + sigma*randn();
        y(i+1, j) = y(i, j) + sigma*randn();
    end
end
```

% Animation

```
for t=1:N
    plot(x(1:t), y(1:t), 'r');
    grid on;
    axis([-5 5 -5 5])
    hold on;
    drawnow;
end
```

6. MATLAB 응용 : Geometric Brownian Motion

□ 예제5 Brownian Motion Plot

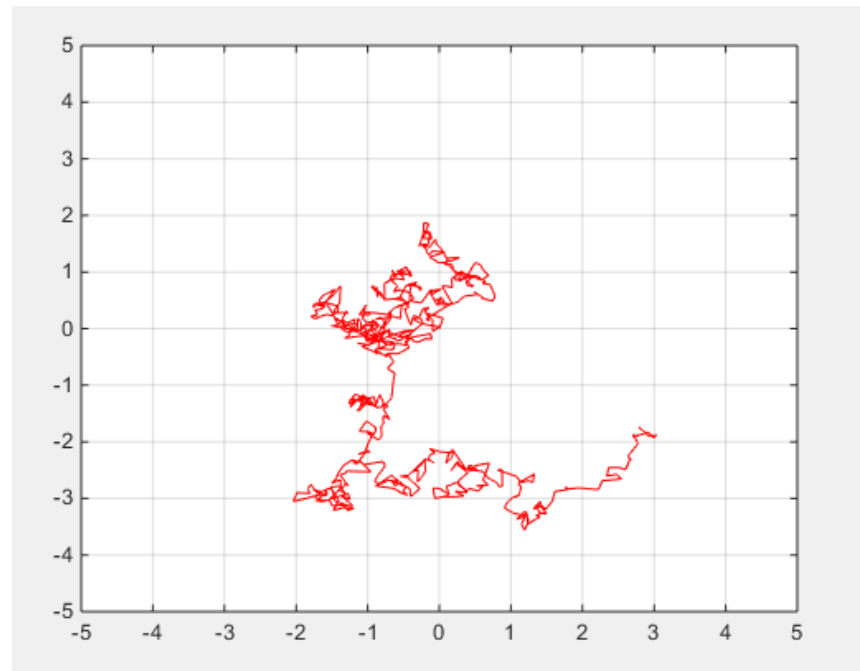
```
% 2D Brownian Motion Plot

N = 1000;
sigma = 0.1;

x = zeros(1, N);
y = zeros(1, N);

for j = 1:N
    for i = 1:N
        x(i+1, j) = x(i, j) + sigma*randn();
        y(i+1, j) = y(i, j) + sigma*randn();
    end
end

% Animation
for t=1:N
    plot(x(1:t), y(1:t), 'r');
    grid on;
    axis([-5 5 -5 5])
    hold on;
    drawnow;
end
```

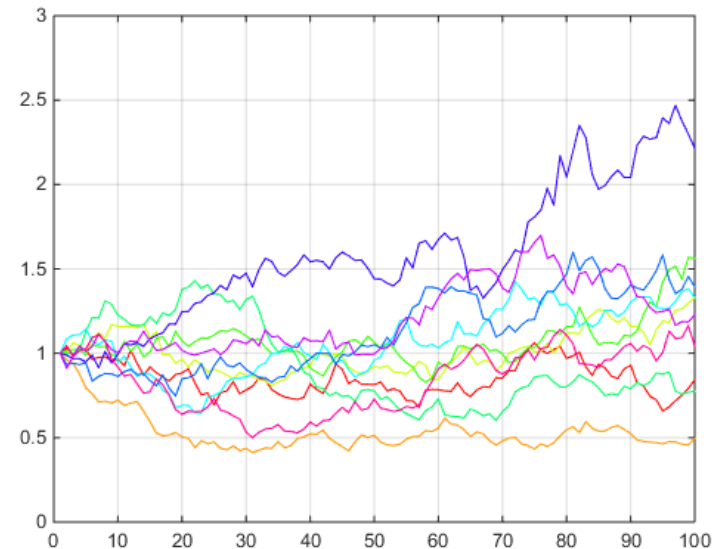


6. MATLAB 응용 : Geometric Brownian Motion

□ 예제6 Geometric Brownian Motion Plot

% Geometric Brownian Motion Animation

```
N=100;  
D=10;  
sigma = 0.5;  
mu = 0.3;  
  
cmap = hsv(D);  
for d=1:D  
    p = ones(1, N);  
  
    for t=2:N  
        p(t) = p(t-1) * (1 + mu/N + sigma*randn()/sqrt(N));  
    end  
  
    for t=1:N  
        plot(1:t, p(1:t), 'Color', cmap(d,:))  
        grid on;  
        axis([0 N 0 3])  
        hold on;  
        drawnow;  
    end  
end
```



End

Question & Comment