MATLAB 정리 _{윤형준}

な署

3p - MATLAB이란

5p - 기초 command window(명령창) 이용

8p - Script(편집기) 이용

11p - 문법

16p - 함수

20p - 파일 입출력

22p - 2차원 그래프

32p - 3차원 그래프

41p - 엑셀 데이터 이용

MATLAB이 라



•

MATLAB이란 Matrix Laboratory의 약자로, matrix 기반의 연산 및 실행을 하는 프로그램.

MATLAB의 장점으론, 쉽고 교육하기 좋음. 내장 함수가 많아 복잡한 알고리즘을 간단히 사용 가능. 신호 분석, 머신러닝 패키지 등 패키지가 많음.

MATLAB의 단점으론, 프로그램 자체 용량이 크고 무거움. 다른 언어들에 비해 계산 속도가 느림. 가격이 비쌈.



기초 command window(명령창) 이용



기호

- = (등호) : 대입 ex) s = 1, a = c 등
- [] (대괄호) : 행렬/벡터 ex) [1 2 3], [4; 5; 6] 등
- () (중괄호) : 행렬 내 숫자 특정 ex) a(1, 2)이면 1행 2열인 2출력
- ; (세미콜론) : 행 생성, 명령창 결과 출력X ex) [1; 2; 3], a = b; 등
- •:(콜론):범위 지정 ex) c(2, 1:end)이면 ans = [34] -> c(2, :) 똑같음
- ' (작은따옴표) : 전치행렬 ex) s = b'이면 s = [4; 5; 6]
- . (마침표) : 행렬 각 원소에 적용

• ... (마침표x3): 줄넘기기/엔터 역할 가능

• <>(부등호) : 참 거짓 판단

• +-*/: 사칙연산 ex) 3*2, a + b, 등

간단 명령어

- mod(x, y) : x를 y로 나눈 나머지 ex) mod(13, 4) -> 1출력
- whos (s): 작업공간 데이터 확인 ex) whos, whos a 등
- zeros(x, y) : x행 y열 영행렬 생성 ex) zeros(3, 3)
- eye(x):x행 x열 단위행렬 생성 ex) eye(4)
- ones(x, y) : 모든 원소가 1인 행렬 생성 ex) ones(2, 3)
- sum(A): 열끼리의 합 우선, 1열뿐이라면 행끼리의 합 ex) sum(a'), sum(sum(c)) =

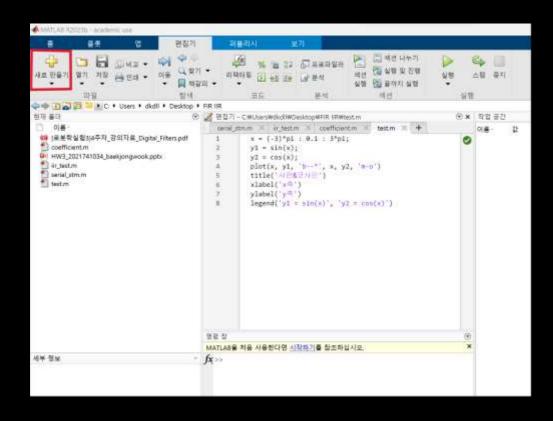
- size(A) : 행렬의 크기 ex) size(a)이면 1(행) 2(열) 출력
- round(x), floor(x), ceil(x): 차례로 반올림 내림 올림
- find(조건식): 조건식을 만족하는 인덱스 찾기

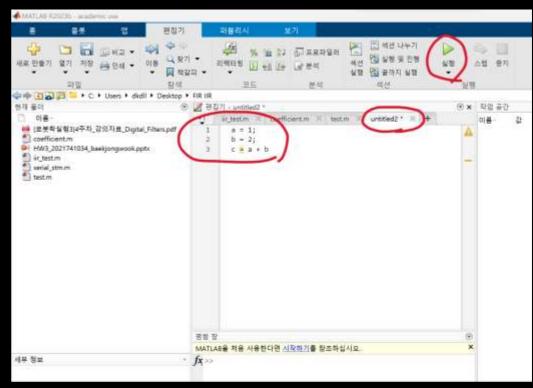
ex) find(a<b)이면 1 2 3 출력

Script(편집기) 이용



편집기 이용





새로 만들기 -> 새 스크립트

스크립트 생성 및 명령어 작성 후 실행

편집기 이용





이런 문구 나오면 폴더 변경 클릭



편집기 명령에 의해 명령창에 결과 출력.

clear;: 작업공간 초기화

clc;:명령창 초기화

clear; clc;은 앞에 써두는 게 좋음.

문법



반복문

•

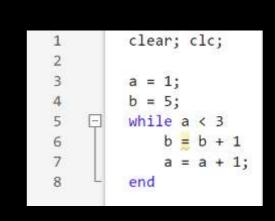
for 변수명 = 초기값:증분:최종값 문장 end



예시)

```
1 clear; clc;
2
3 for x = 0:1:3
4 a = x
end
```







조건문

```
•
```

```
if 조건 1
문장 1
elseif 조건 2
문장 2
else
문장 3
end
```

```
clear; clc;
           a = 2;
           if a < 1
               b = 1
           elseif a < 3
               b = 2
           else
               b = 3
 10
           end
명령 창
  b =
```

switch문

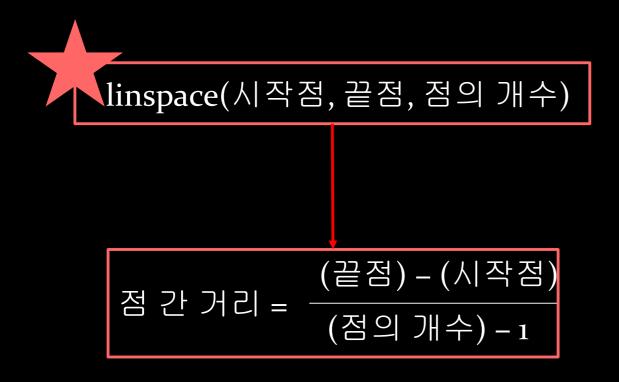
•

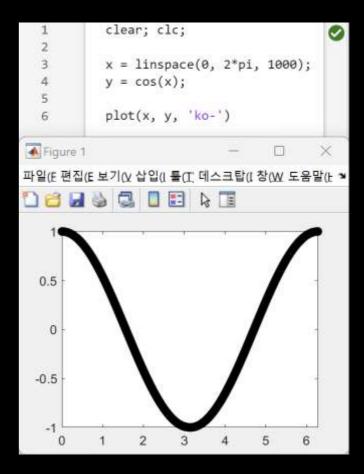
```
switch 조건
case 조건 1
문장 1
case 조건 2
문장 2
Otherwise
문장 3
end
```

```
clear; clc;
           i = 1;
           switch i
               case 1.5
                   b = 1
               case 3
                   b = 3
               otherwise
                   b = 5
 10
 11
           end
명령 창
 b =
        5
```

linspace 선 나누기



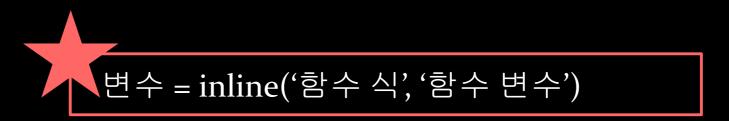




함수



inline 함수



함수 및 변수 설정 후 숫자 대입에 따른 결과 확인 가능

```
명령 창

>> f = inline('x^4+5*x^2+x+10', 'x')

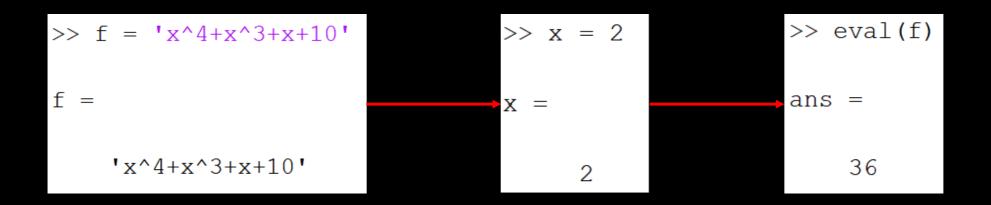
f =

인라인 함수:
f(x) = x^4+5*x^2+x+10
```

```
>> f(5)
ans =
765
```

eval(변수): char형 함수 식이어도 변수 설정 후 계산 가능한 함수

함수 설정 -> 변수 설정 -> eval(함수명) 함수 계산



사용자 함수 생성 및 사용



function 출력변수 = 함수이름(입력변수)

새 스크립트 생성 -> 코드 작성 -> 저장(파일 이름은 함수 이름과 동일) -> 현재 폴더에 파일 존재하면 함수 사용 가능

```
function f = multimat(x)

f = [x(1) x(2) x(3);

2*x(1) 2*x(2) 2*x(3);

x(3) x(2) x(1)];
```

```
function f = multimat(x)
f = x^4+2*x^2+x+1;
```

```
명령 창

>> x = [1 2 3];
>> multimat(x)

ans =

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

```
명령 창

>> a = 2;
>> multimat(a)

ans =

27
```

파일 입출력



파일 입출력

•

MATLAB 파일 입출력은 C언어와 매우 유사.

```
fopen('파일 이름', '형식'): 파일 생성 및 열기
fprintf(파일 변수, '내용'): 내용 작성
fclose(파일 변수): 파일 닫기
```

예시)

1	clear; clc;
2	
3	<pre>fp = fopen('filetest.m', 'w');</pre>
4	fprintf(fp, '%d %d \n', 10, 21);
5	fprintf(fp, '%c %c \n', 77, 80);
6	<pre>fclose(fp);</pre>

filetest.m

10 21
MP

2차원 그래프



n차 다항식 구하기

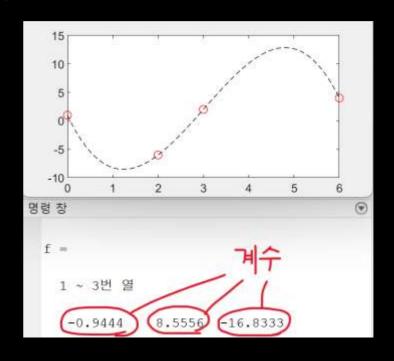
•

- polyfit(x좌표, y좌표, 다항식 차수):계수 찾기
- polyval(polyfit변수, linspace변수): y좌표 계산

```
clear; clc;

x = [0 2 3 6];
y = [1 -6 2 4];

f = polyfit(x, y, 3)
xx = linspace(0, 6, 100);
yy = polyval(f, xx);
plot(x, y, 'ro', xx, yy, 'k--')
```



그래프 관련 명령어

- clf (=clear figure) : figure창에서 그림 지우기
- close all : figure창 모두 닫기
- figure(x):x값을 갖는 figure창을 띄움/생성
- hold on : 여러 개의 그래프를 한 figure에 겹쳐 그릴 때 사용
- hold off: 겹쳐 그리기 끝낼 때 사용
- plot(x, y, '옵션'...) : 그래프 그리기
- subplot(a, b, c): figure창을 a행 b열로 임의로 나눠 c번째 칸 설정
- title('이름') : figure창에 그림 위쪽 이름 작성
- xlabel('이름'), ylabel('이름'): x축, y축 이름/설명 작성
- legend('이름1', '이름2'...) : 그림 우측상단 그래프 이름 작성
- grid on : 그래프 격자선 보이기/생성
- axis([xmin, xmax, ymin, ymax]) : 괄호 안 x, y 범위만을 보이기

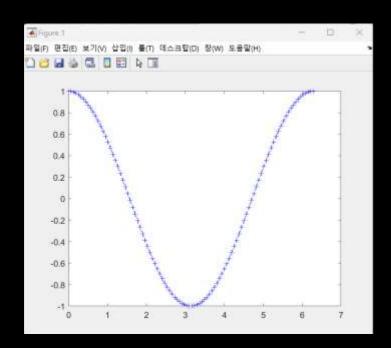
그래프 그리기 예시

: hold on

```
clear all; clc; close all;

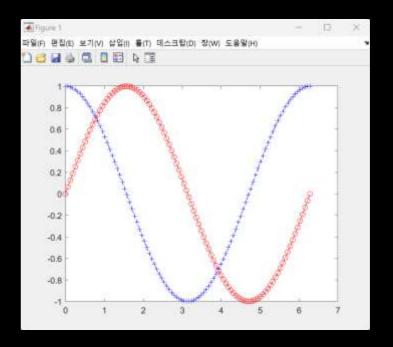
x = linspace(0, 2*pi, 100);
y1 = sin(x);
y2 = cos(x);
plot(x, y1, 'ro-')

plot(x, y2, 'b+--')
```



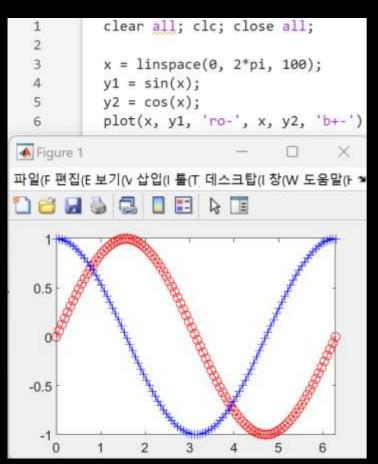
```
clear all; clc; close all;

x = linspace(0, 2*pi, 100);
y1 = sin(x);
y2 = cos(x);
plot(x, y1, 'ro-')
hold on
plot(x, y2, 'b+--')
```



그래프 그리기 예시 : plot

옵션 종류



		마커		설명	결과로 생성되는 마커
		-0-		원	0
색 이름	짧은 이름	-,-		플러스 기호	+
TAN DESC.		***		별표	*
"red"	"r"	*(*):		점	
"green"	"g"	"x"		십자	×
		* *		가로선	+:
"blue"	"ь"	-1-		세로선	1
"cyan"	"c"	"square"		정사각형	
Cyan	710 4 00	"diamond"		다이아몬드	0
"magenta"	"m"	-A=		위쪽 방향 삼각형	Δ
New Street at 1999 as	090034	*v*		아래쪽 방향 삼각형	∇
"yellow"	"у"	-5-		오른쪽 방향 삼각형	>
"black"	"k"	*c*		왼쪽 방향 삼각형	△
	2007	"pentagram"		편타그램	ŵ
"white"	"w"	"hexagram"		핵사그램	ø
선 스타일			설명		결과 선
"-"			실선		
""			파선		

선 스타일	설명	결과 선
"_"	실선	
""	파선	
":"	점선	
""	일점 쇄선	

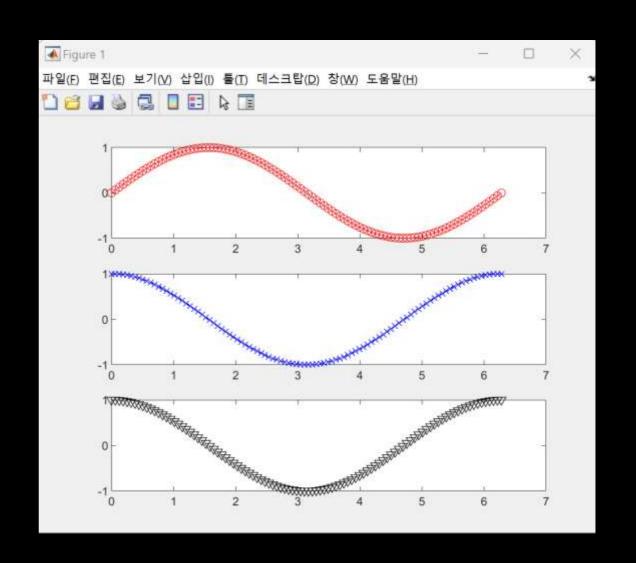
그래프 그리기 예시

•

: subplot

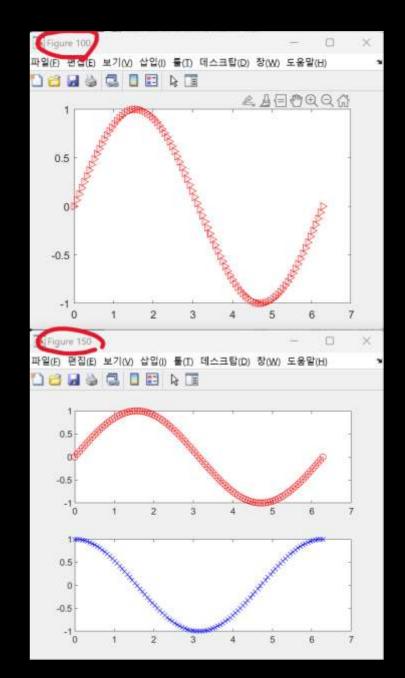
```
clear all; clc; close all;
x = linspace(0, 2*pi, 100);
y1 = sin(x);
y2 = cos(x);
y3 = tan(x);
subplot(3, 1, 1)
plot(x, y1, 'ro-')
subplot(3, 1, 2)
plot(x, y2, 'bx--')
subplot(3, 1, 3)
plot(x, y2, 'kv:')
```

3행 1열로 나눈 공간에서의 순서



그래프 그리기 예시 : figure

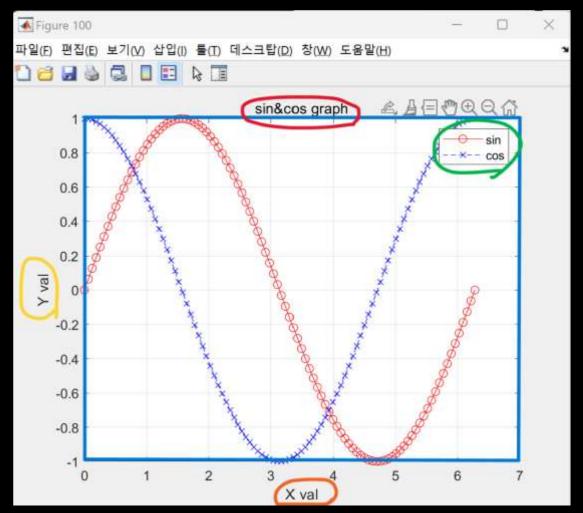
```
clear all; clc; close all;
x = linspace(0, 2*pi, 100);
y1 = sin(x);
y2 = cos(x);
figure(100);
plot(x, y1, 'r>:')
figure(150);
subplot(2, 1, 1)
plot(x, y1, 'ro-')
subplot(2, 1, 2)
plot(x, y2, 'bx--')
```



그래프 그리기 예시

: title, label, legend, grid on

```
clear all; clc; close all;
          x = linspace(0, 2*pi, 100);
          y1 = sin(x);
          y2 = cos(x);
         figure(100);
          plot(x, y1, 'ro-', x, y2, 'bx--')
          title('sin&cos graph')
          xlabel('X val')
          ylabel('Y val')
11
          legend('sin', 'cos')
          grid on
```



그래프 그리기 예시

•

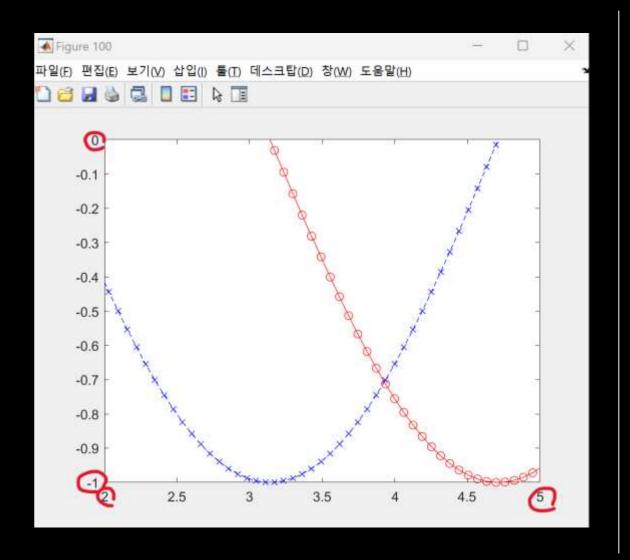
: axis

```
clear all; clc; close all;

x = linspace(0, 2*pi, 100);
y1 = sin(x);
y2 = cos(x);

figure(100);
plot(x, y1, 'ro-', x, y2, 'bx--')
axis([2 5 -1 0])
```

X축:2~5 / Y축:-1~0

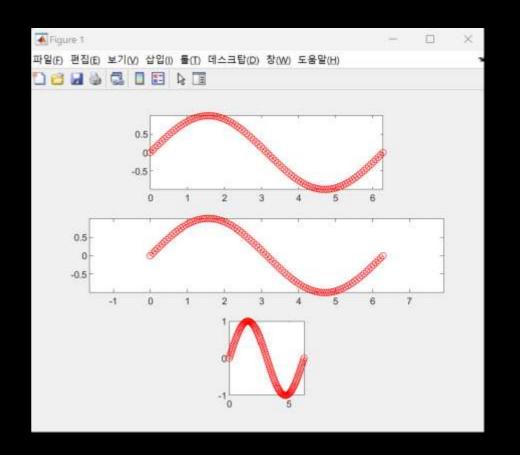


axis 그래프 보정

•

- axis image : 원래 그래프, box크기 유지
- axis equal : 원래 그래프 유지, box크기 3:4:4
- axis square : 그래프 변경, box크기 1:1:1

```
clear all; clc; close all;
 3 🔊
         x = linspace(0, 2*pi, 100);
         y1 = sin(x);
          subplot(3, 1, 1)
          plot(x, y1, 'ro-')
          axis image
10
          subplot(3, 1, 2)
          plot(x, y1, 'ro-')
12
          axis equal
13
14
          subplot(3, 1, 3)
15
          plot(x, y1, 'ro-')
          axis square
```



3차원 그래프



관련 명령어

- plot3(x, y, z, '옵션'): 3차원 그래프 그리기
- [X, Y] = meshgrid(minx:maxx, miny:maxy)

: 범위에 맞는 격자(grid) 좌표를 생성/반환

- mesh(x, y, z): 3차원 곡면 망사(mesh) 플롯 생성/그리기
- contour(x, y, z, n) : 2차원 n개의 등고선 그리기
- contour3(x, y, z, n): 3차원 n개의 등고선 그리기
- quiver(x, y, a, b, scale) : (x, y)에서 (a, b)로의 벡터 그리기 (scale은 화살표의 상대적 길이-기본 1)
- quiver3(x, y, z, a, b, c, scale) : (x, y, z)에서 (a, b, c)로의 벡터 그리기
- [px, py] = gradient(z): z의 그레디언트(수치적 기울기/공간에 대한 기울기)에서의 x, y성분을 반환

3차원 그래프 그리기 : plot3 예시

```
clear all; clc; close all;

t = 0:0.1:10;

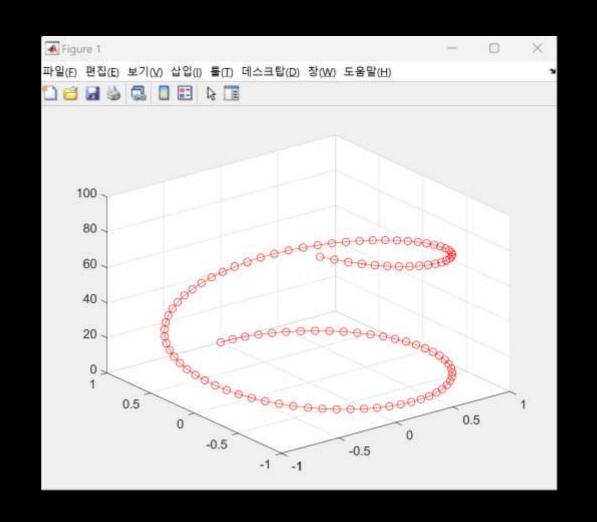
x = sin(t);

y = cos(t);

z = t.^2;

plot3(x, y, z, 'ro-');

grid on
```

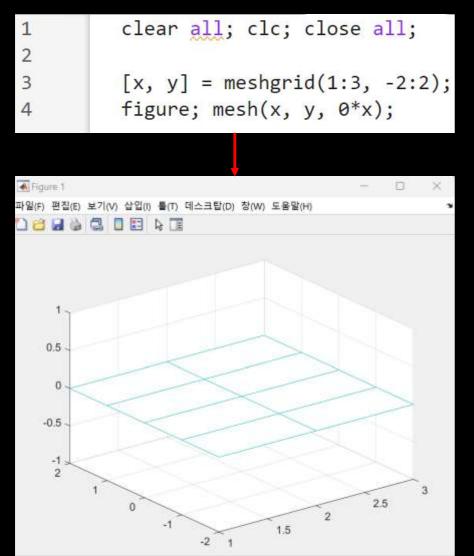




•

: meshgrid, mesh 예시

행렬 이용

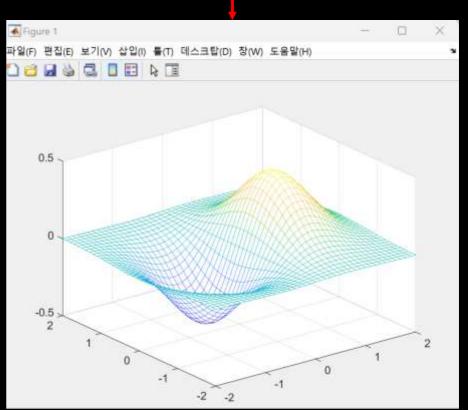


```
clear all; clc; close all;

(x, y] = meshgrid(-2:0.1:2, -2:0.1:2);

z = x.*exp(-x.^2-y.^2);

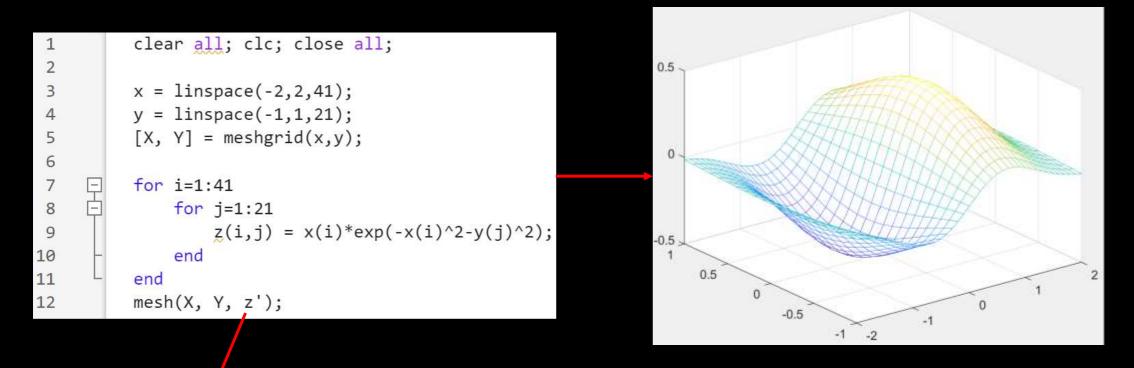
figure; mesh(x, y, z);
```



•

: meshgrid, mesh 예시

for문 이용

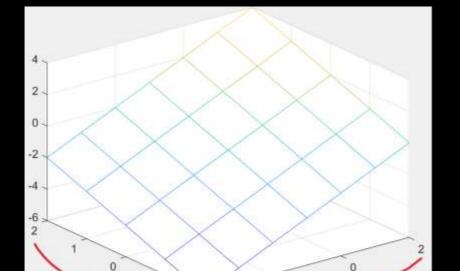


z'인 이유 : meshgrid로 X, Y는 21x41행렬이 생성되지만 for문에 의해 z는 41x21행렬로 생성됨.

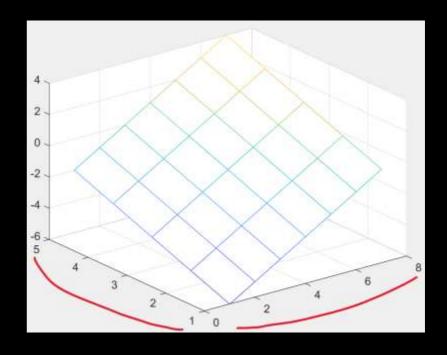
: mesh(z) vs mesh(x, y, z)

mesh(x, y, z) 명령어는 meshgrid와 같이 앞서 정해준 x, y범위를 반영해 표현하지만 mesh(z) 명령어는 범위를 반영하지 않고 그 범위 내 수들의 인덱스를 사용해 표현함.

mesh(x, y, z)



mesh(z)



: contour, contour3 예시

```
clear all; clc; close all;

(x, y] = meshgrid(-2:0.1:2, -2:0.1:2);

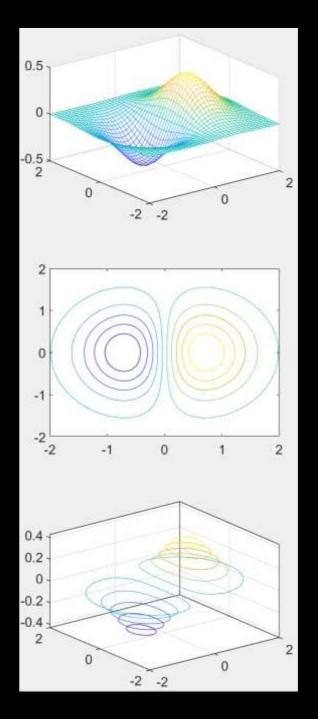
z = x.*exp(-x.^2-y.^2);

figure;

subplot(3, 1, 1)
mesh(x, y, z)
subplot(3, 1, 2)
contour(x, y, z, 10)

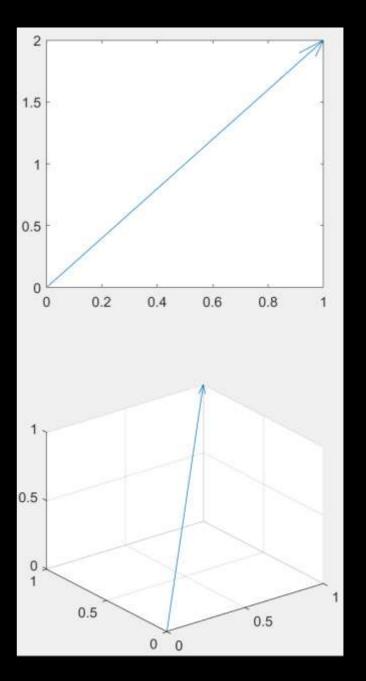
contour(x, y, z, 10)

subplot(3, 1, 3)
contour3(x, y, z, 10)
```



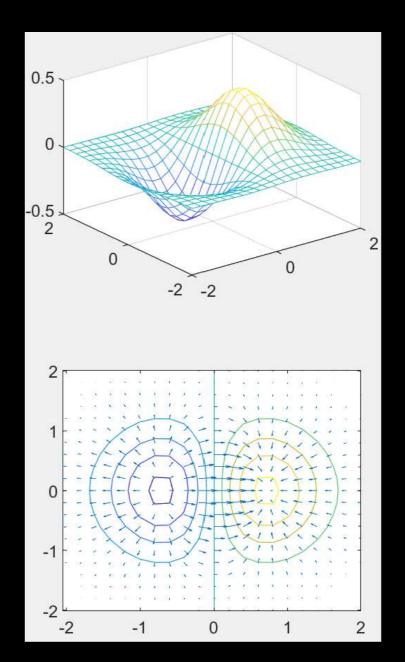
3차원 그래프 그리기 : quiver, quiver3 예시

```
1 clear all; clc; close all;
2
3 subplot(2, 1, 1)
4 quiver(0, 0, 1, 2, 1)
5 subplot(2, 1, 2)
quiver3(0, 0, 0, 1, 1, 1, 1)
```



3차원 그래프 그리기 : gradient 예시

```
clear all; clc; close all;
          [x, y] = meshgrid(-2:0.2:2);
         z = x.*exp(-x.^2-y.^2);
          [px,py] = gradient(z);
         subplot(2, 1, 1)
         mesh(x, y, z)
         subplot(2, 1, 2)
         contour(x,y,z)
         hold on
10
         quiver(x,y,px,py, 1)
11
         hold off
```



엑셀 데이터 이용



엑셀 데이터 이용 예시



4	А	В	С	D	E
1	Name	Date	Location	Weight	Height
2	사람	03월 02일	목동	79.8	169
3	사람	03월 03일	목동	79.3	171
4	사람	03월 04일	노원구	78.9	171
5	사람	03월 05일	노원구	80	173
6	사람	03월 06일	의정부	78.8	175
7	사람	03월 07일	의정부	75	175
8	사람	03월 08일	의정부	69	178

위 MatlabTest.xlsx 라는 엑셀 파일이 있을 때

1 TEST = readtable('MatlabTest.xlsx')

readtable('파일명') 으로 데이터 불러오기



TEST =				
7×5 <u>table</u>				
Name	Date	Location	Weight	Height
{'사람'}	2024-03-02	('목동')	79.8	169
('사람')	2024-03-03	('목동')	79.3	171
{ '사람 ' }	2024-03-04	{'노원구'}	78.9	171
('사람')	2024-03-05	('노원구')	80	173
('사람')	2024-03-06	['의정부']	78.8	175
('사람')	2024-03-07	('의정부')	75	175
('사람')	2024-03-08	('의정부')	69	178

데이터 저장 형식

이름, 몸무게 등 특정 데이터만을 사용할 때는 '데이터 저장 변수.데이터 이름'사용 ex) Test.Name, Test.Weight 엑셀 데이터 이용 예시

•

엑셀 데이터를 이용한 그래프 그리기

```
1    TEST = readtable('MatlabTest.xlsx')
2    subplot(2, 1, 1)
4    plot(TEST.Date, TEST.Weight, 'ro-')
5    subplot(2, 1, 2)
6    plot(TEST.Date, TEST.Height, 'bo-')
```

