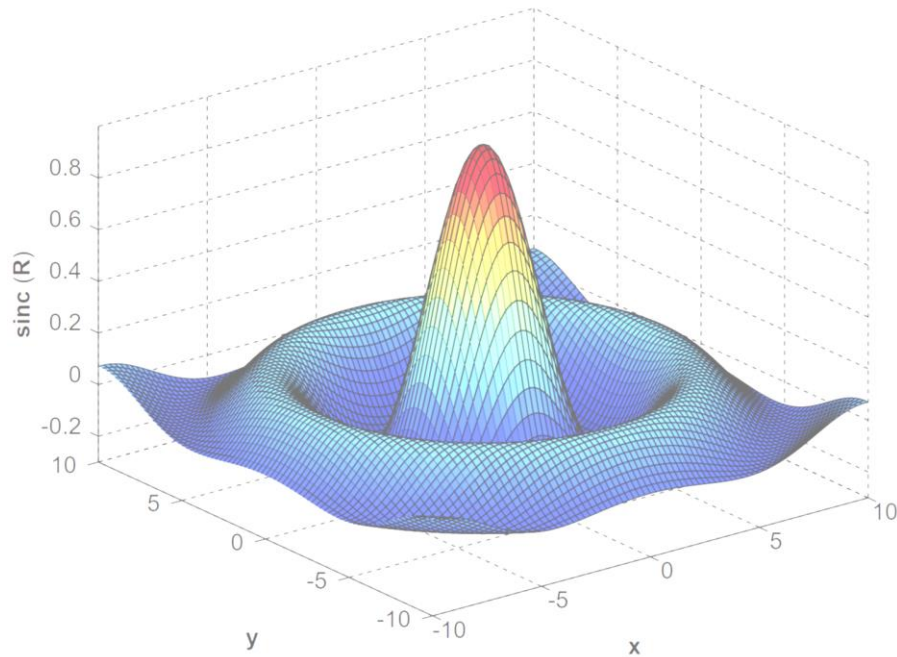


# 응용전산및실습 I

## (강의자료 #5)



교과목명 : 응용전산 및 실습 I

담당교수 : 이 수 형

E-mail : [soohyong@uu.ac.kr](mailto:soohyong@uu.ac.kr)

교재명 : 유인물

# 중간고사 공지

- 일시 : 2021년 4월 21일 오후 3시
- 장소 : 강의실
- 시험 방법 : 이론 및 실기시험

# 2차원 그래프 #1

# 2차원 그래프

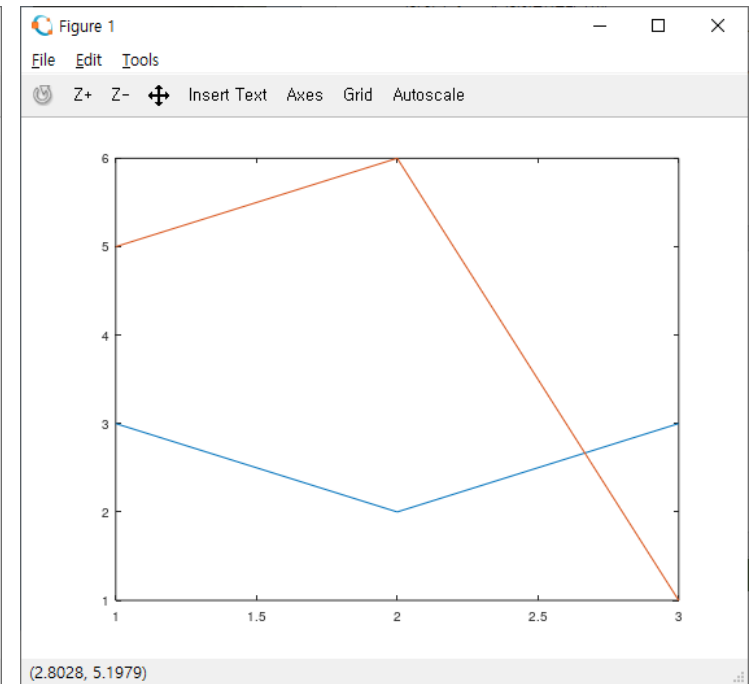
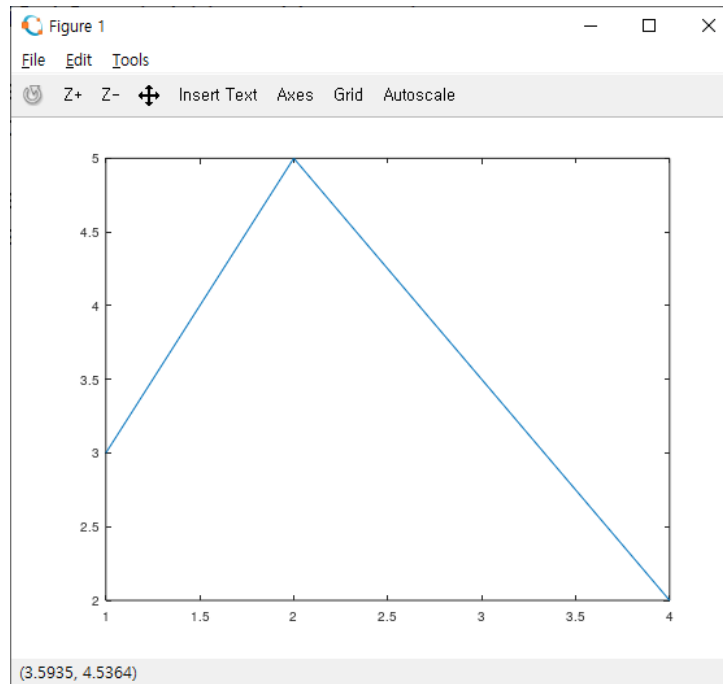
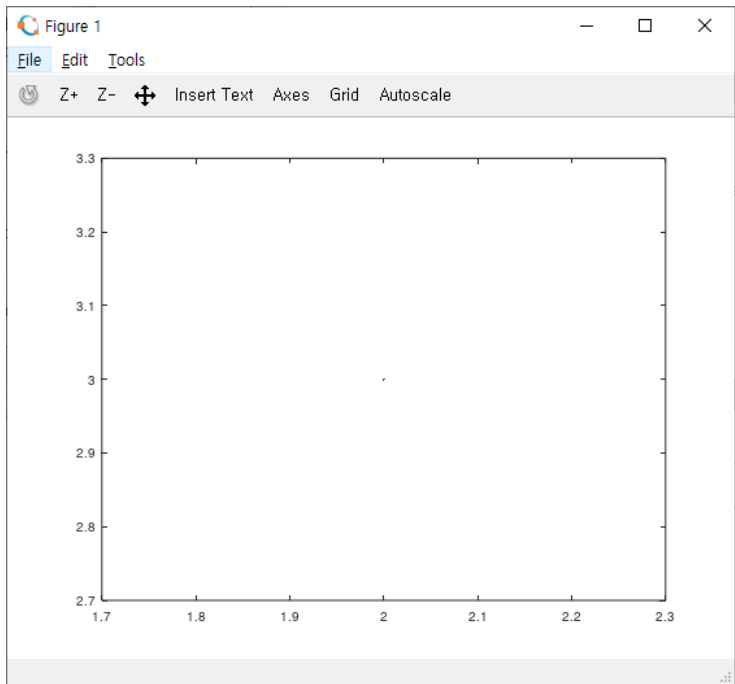
- 2차원 그래프 함수

- `plot(x, y)` : 기본적인 2차원 그래프 함수
- `x` : `x`축 데이터, `y` : `y`축 데이터 → 두 데이터의 크기는 동일해야 함
- `x, y`의 크기에 따른 그래프 표현
  - 스칼라 : 점으로 표시, Matlab에서는 표시 안됨 → 표식(mark) 지정해야 함
  - 벡터 : 선으로 표시
  - 행렬 : 여러 개의 선으로 표시

# 2차원 그래프

- plot() 함수 사용

```
>> plot(2, 3)
>> plot([1 2 4], [3 5 2])
>> plot([1 1; 2 2; 3 3], [3 5; 2 6; 3 1])
```



# 2차원 그래프

- 함수 그래프 그리기 :  $f(x) = \sin x, 0 \leq x \leq 2\pi$ 
  - 데이터의 준비 :  $x = [0, 0.2, 0.4, \dots, 2\pi], y = [\sin 0, \sin 0.2, \dots, \sin 2\pi]$

```
>> x = 0:0.2:2*pi;
```

```
>> y = sin(x); ⇒ sin 함수 : 벡터 입력 → 벡터 출력
```

-  $x$ 축 데이터의 간격 ?

```
>> x = 0:0.5:2*pi;
```

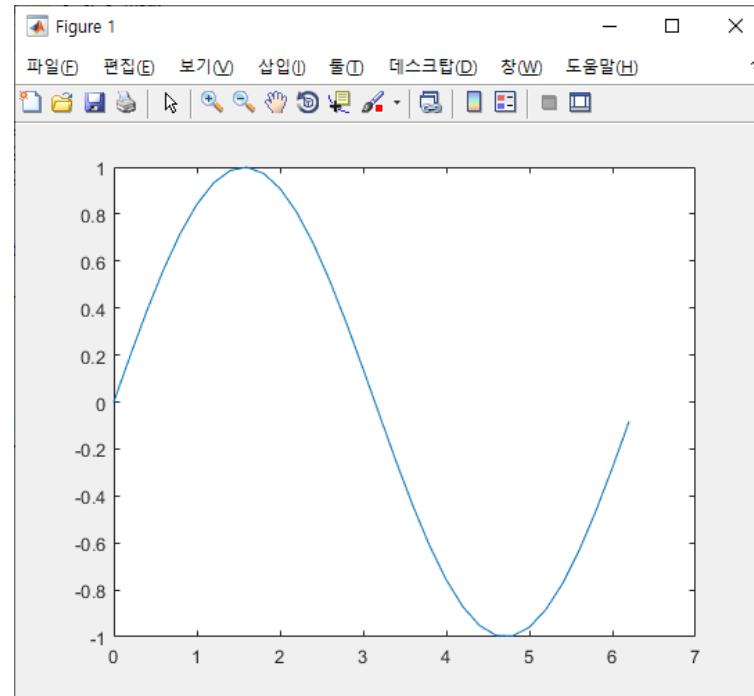
```
>> y = sin(x);
```

```
>> x = 0:0.1:2*pi;
```

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

```
>> x = 0:0.01:2*pi;
```

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



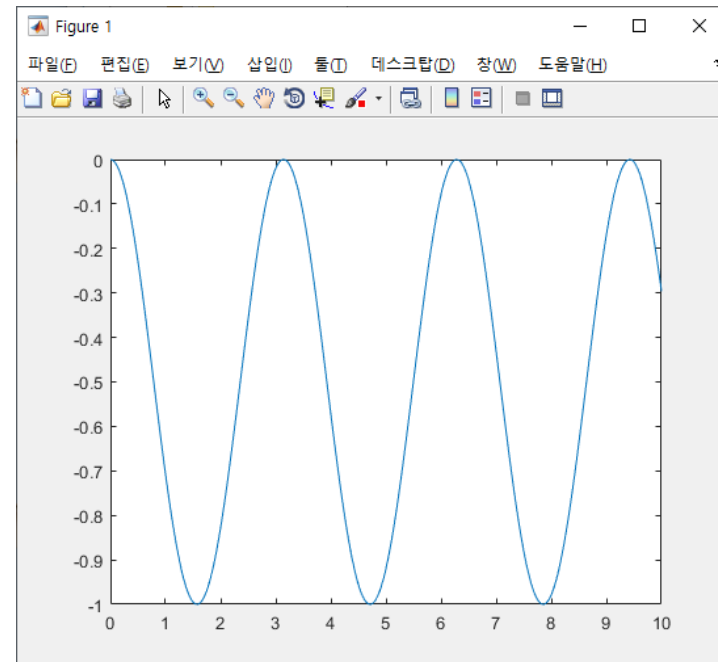
# 2차원 그래프

- 함수 그래프 그리기 :  $f(x) = \sin x \cos\left(x + \frac{\pi}{2}\right), 0 \leq x \leq 10$

- 데이터의 준비 : 벡터의 연산

`>> y = sin(x) .* cos(x + pi / 2);` ⇒ 벡터의 각 요소끼리 연산 `.*` `.^` 등 사용

```
>> x = 0:0.01:10;  
>> y = sin(x) .* cos(x + pi / 2);  
>> size(x)  
ans =  
      1      1001  
  
>> size(y)  
ans =  
      1      1001  
  
>> plot(x, y)  
>>
```



# 2차원 그래프

- 함수 그래프 그리기 : plot() 함수의 활용

- figure() 함수 : 그래프 윈도우 생성, 선택

- figure;                   ⇒ 새로운 그래프 윈도우 생성

- figure(1);               ⇒ 1번 그래프 윈도우 선택 및 활성화, 없으면 생성  
⇒ plot() 함수 등은 활성화된 윈도우에 그래프를 표시함

- plot() 함수 형식

- plot(x, y);               ⇒ 기본 형식

- plot(x, y, s);           ⇒ s : 형식(색상/선/표식) 지정

- plot(x1, y1, x2, y2);               ⇒ 2개의 그래프 동시에 그리기

- plot(x1, y1, s1, x2, y2, s2);       ⇒ 형식 및 2개의 그래프 동시에 그리기

- plot(x, y, '속성', '속성값');       ⇒ 그래프 여러 가지 속성 지정



# 2차원 그래프

- plot() 함수의 형식 설정

- `plot(x, y, 형식)`;  $\Rightarrow$  문자열로 표현된 형식(색상, 표식 모양, 선 모양)

- 형식 : 문자색상 + 표식 모양 + 선모양

- 예] `plot(x, y, 'ro--')`;  $\Rightarrow$  빨간색 'o' 표식의 쇠선 모양

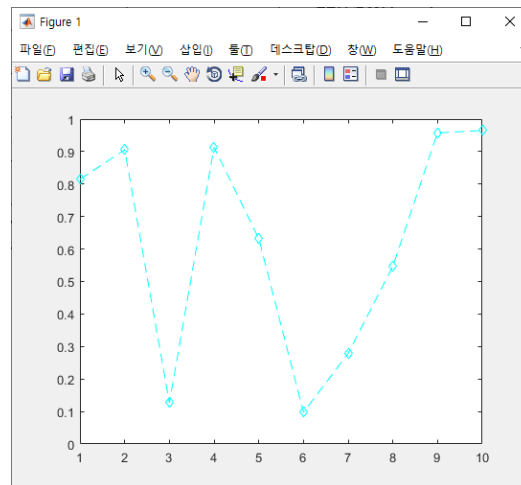
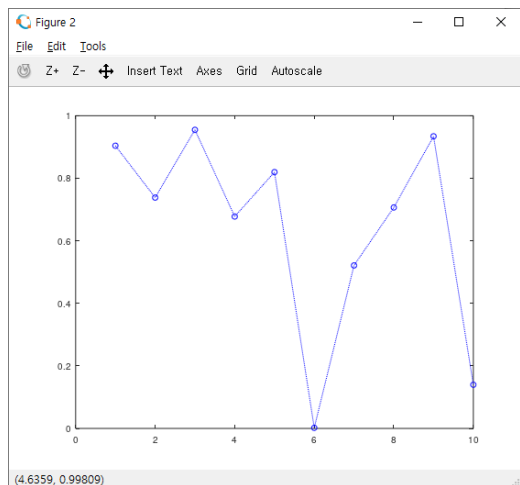
문자색상	표식모양	표식모양	선모양
Red	+	o	- (실선)
Green	*	.	--(쇠선)
Blue	x	^ $\triangle$	-.(1점쇠선)
Magenta	v $\nabla$	> $\triangleright$	:(점선)
Cyan	< $\triangleleft$	s $\square$	
black	d $\diamond$	p $\star$	
White	h (6점별)		

# 2차원 그래프

## • plot() 함수의 형식 설정

```
>> x=1:10;
>> y=rand(1,10);
>>
>> plot(x,y);
>> plot(x,y,'r');
>> plot(x,y,'bo');
>> plot(x,y,'bo:');
>> plot(x,y,'cd--');
>> plot(x,y,'-mo');
```

문자색상	표식모양	표식모양	선모양
Red	+	o	- (실선)
Green	*	.	--(쇄선)
Blue	x	^ △	-.(1점쇄선)
Magenta	v ▽	> ▷	:(점선)
Cyan	< ◁	s □	
black	d ◇	p ☆	
White	h (6점별)		



# 출석 과제

- 실습 내용을 따라서 수행하고 결과 화면을 캡처해서 제출하시오.
- 마지막 plot() 함수의 경우 색상/선모양/표식을 조합하여 10가지 이상의 형태를 임의로 만들어서 그래프 윈도우를 캡처해서 제출하시오.

# 2차원 그래프

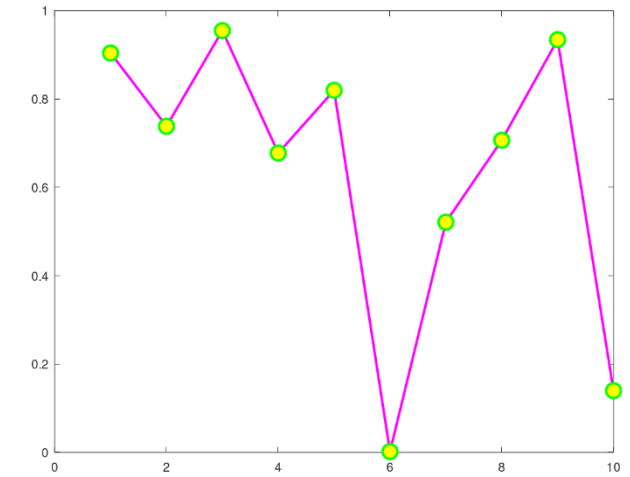
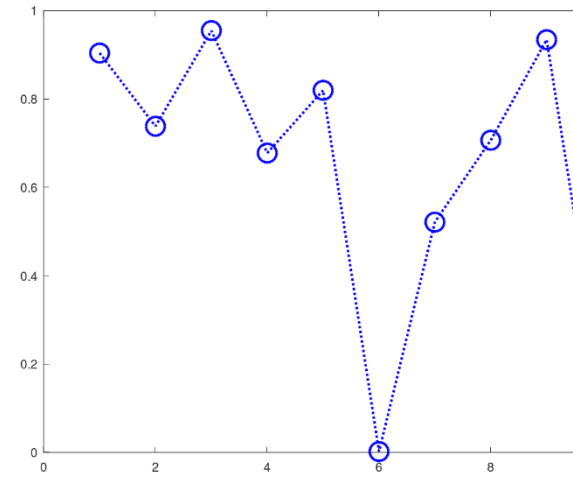
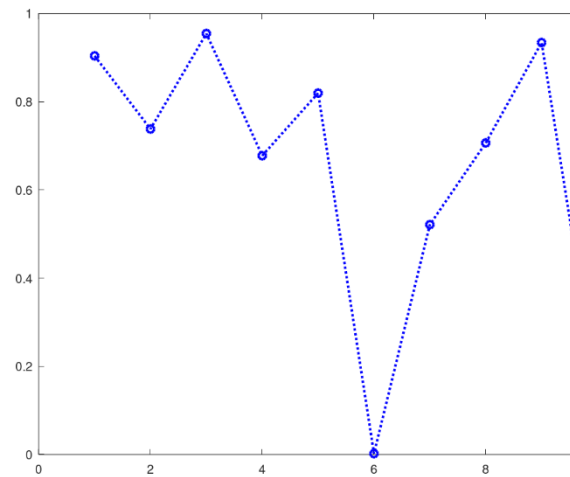
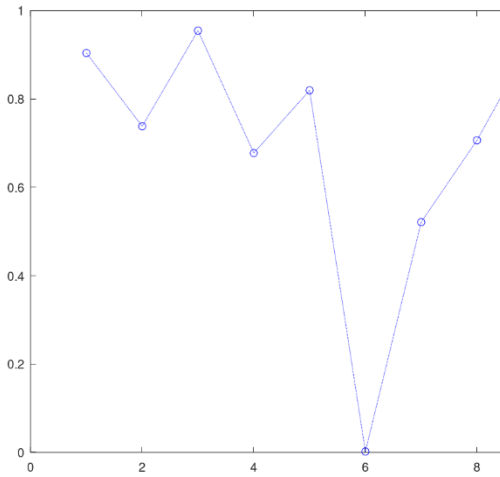
- plot() 함수의 속성 설정
  - 속성(property) : 그래프의 다양한 속성들을 지정할 수 있음
  - 속성 + 속성 값의 형식으로 인수들을 입력
    - 예] `plot(x, y, 'LineWidth', 2);` ⇒ 선의 굵기를 2로 지정 (기본 : 0.5)

Property	설명	속성값
LineWidth	선의 굵기	포인트 단위(0.5)
MarkerSize	표식의 크기	포인트 단위
MarkerEdgeSize	표식의 색 (테두리)	문자열 표시 (rgb..)
MarkerFaceColor	표식의 배경색	문자열 표시 (rgb..)

# 2차원 그래프

- plot() 함수의 속성 설정

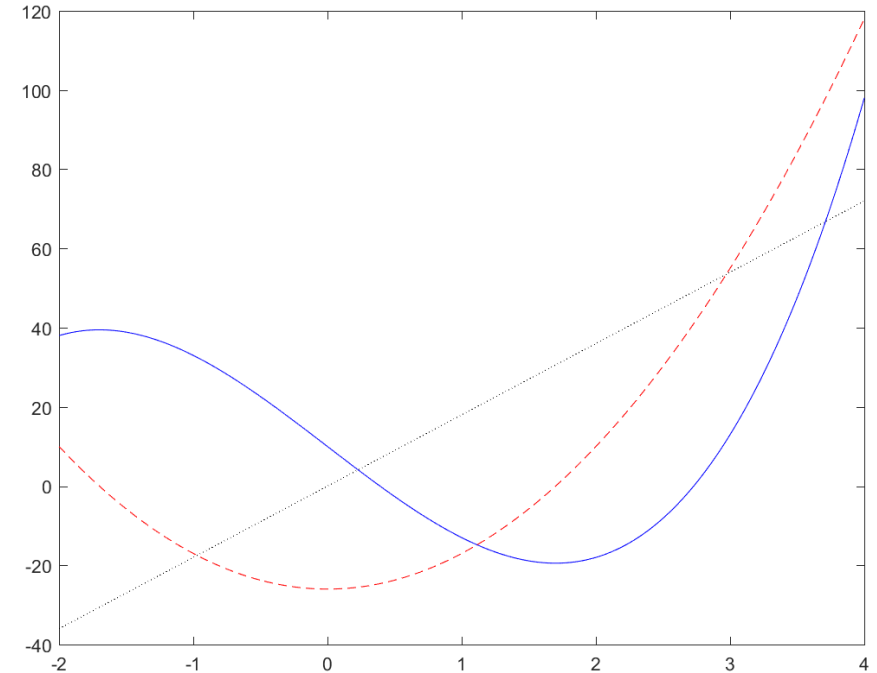
```
>> plot(x, y, 'bo:');  
>> plot(x, y, 'bo:', 'LineWidth', 2);  
>> plot(x, y, 'bo:', 'LineWidth', 2, 'MarkerSize', 5);  
>> plot(x, y, 'bo:', 'LineWidth', 2, 'MarkerSize', 15);  
>> plot(x, y, '-mo', 'LineWidth', 2, 'MarkerSize', 12, 'MarkerEdgeColor', 'g',  
'MarkerFaceColor', 'y')
```



# 2차원 그래프

- 2개 이상의 그래프 그리기
  - `plot(x1, y1, x2, y2);`
  - `plot(x1, y1, s1, x2, y2, s2);`
  - $y = 3x^3 - 26x + 10$ 의 함수와  $y'$ ,  $y''$ 의 함수를 같이 그려라.
    - $y' = 9x^2 - 26$ ,  $y'' = 18x$

```
>> x = -2:0.01:4;  
>> y = 3 * x.^3 - 26 * x + 10;  
>> y1 = 9 * x.^2 - 26;  
>> y2 = 18 * x;  
>> plot(x, y, x, y1);  
>> plot(x, y, '--', x, y1, 'g:');  
>> plot(x, y, '--', x, y1, 'm:');  
>> plot(x, y, 'b-', x, y1, 'r--', x, y2, 'k:');
```



# 2차원 그래프

- 2개 이상의 그래프 그리기

- hold 명령어 (hold on, hold off, hold : 토글방식)

- 여러 개의 그래프를 한 그림에 나타낼 때 사용

- hold on이 되어 있는 경우에 새로운 그래프를 그려도 이전의 그래프가 남아있음

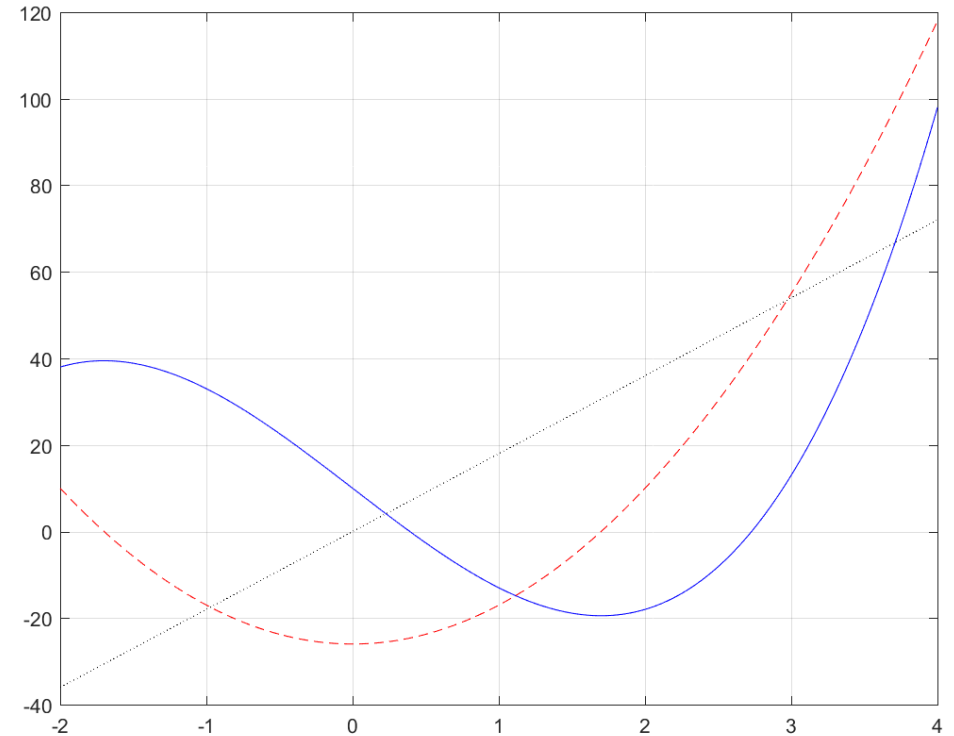
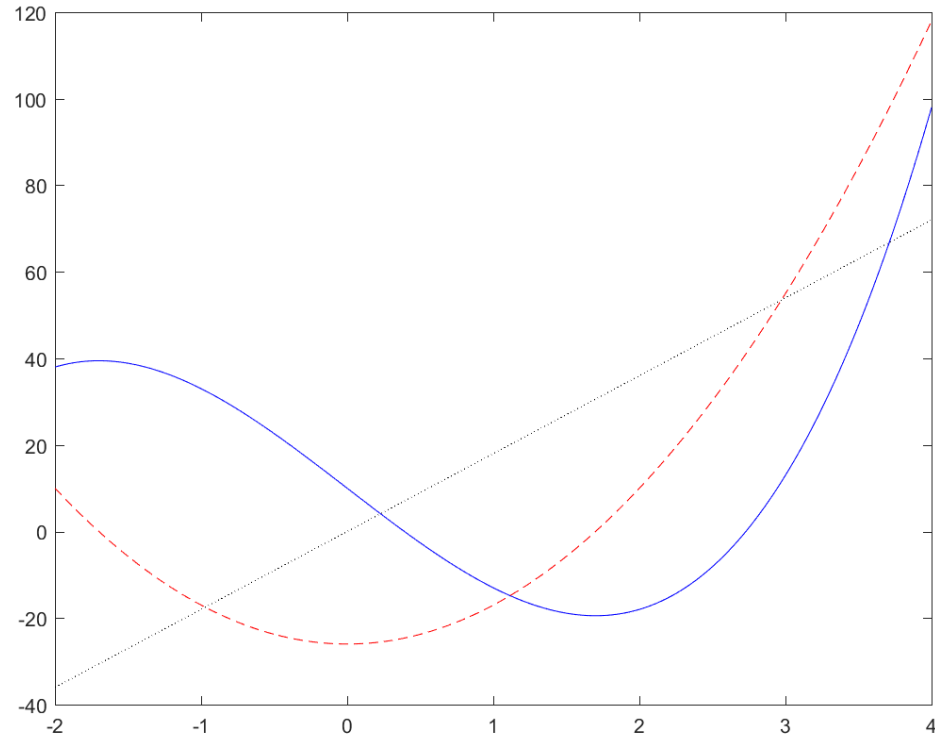
- hold off 되어 있는 경우, 새로운 그래프를 그리면 이전의 그래프는 지워짐

```
>> x = -2:0.01:4;
>> y = 3 * x.^3 - 26 * x + 10;
>> y1 = 9 * x.^2 - 26;
>> y2 = 18 * x;
>> plot(x, y, x, y1);
>> plot(x, y, '--', x, y1, 'g:');

>> plot(x, y, '--');
>> hold on;
>> plot(x, y1, 'g:');
```

# 2차원 그래프

- grid : 격자선 출력 (grid on, grid off, grid)





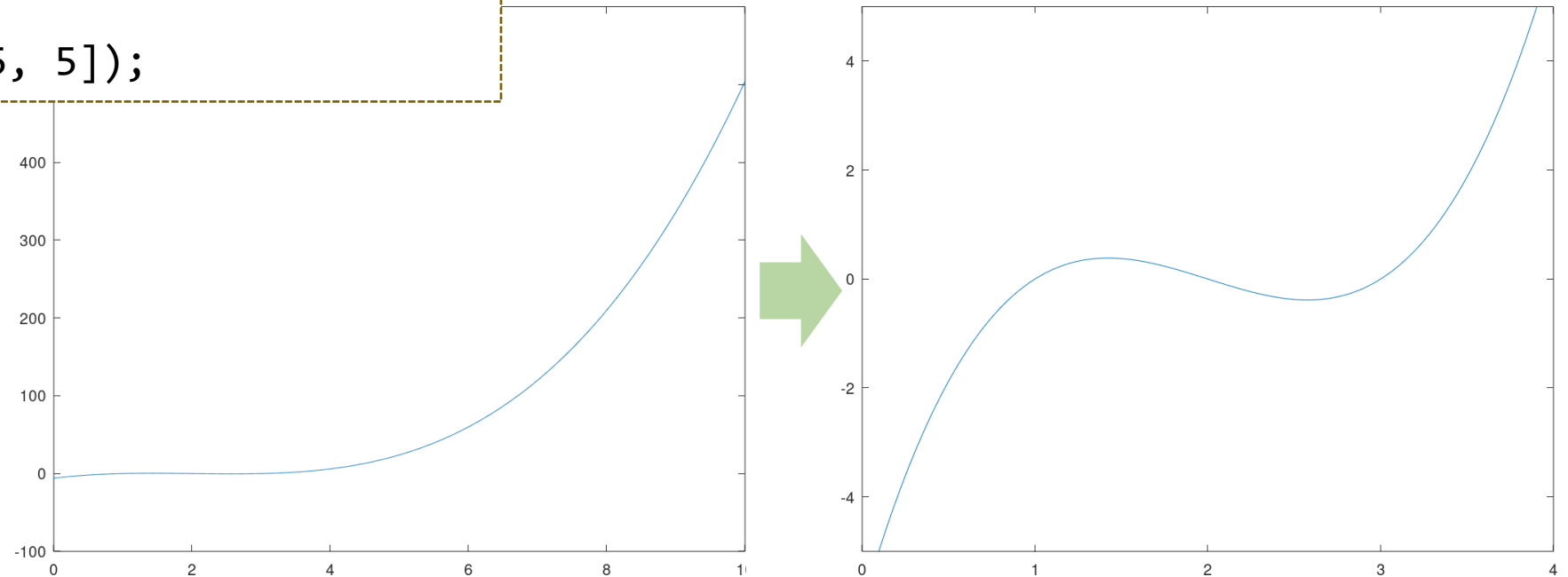
# 2차원 그래프

- axis 함수

- 축의 범위와 모양을 변경할 때 사용

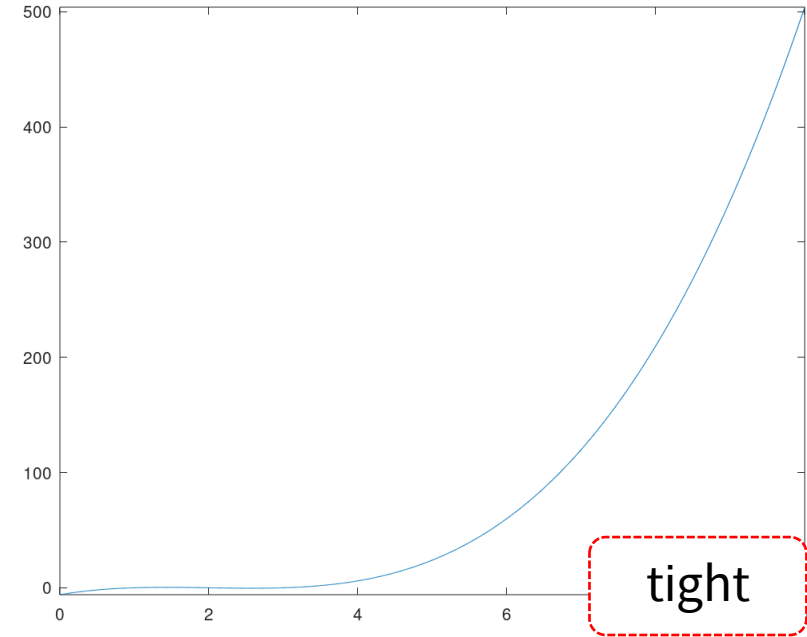
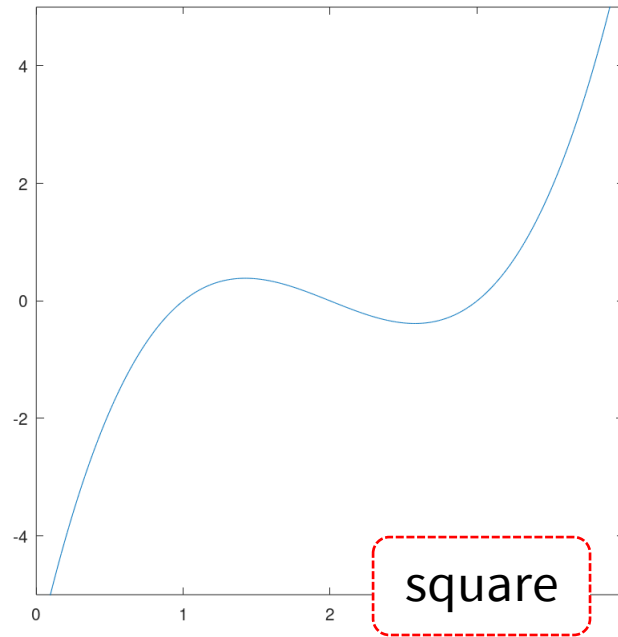
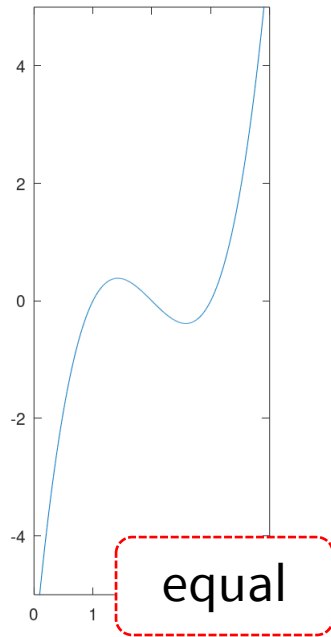
- `axis([xmin xmax ymin ymax]);` ⇒ 그래프의 범위를 재지정

```
>> x = 0:0.01:10;  
>> y = x.^3 - 6 * x.^2 + 11 * x - 6;  
>> plot(x, y);  
>> axis([0, 4, -5, 5]);
```



# 2차원 그래프

- axis 함수
  - axis equal : x축과 y축의 배율을 동일하게 설정
  - axis square : 좌표축의 영역을 정사각형으로 지정
  - axis tight : 좌표축의 범위를 데이터가 있는 영역으로 지정 (여백, 잘림 없음)



- 삼각 함수 그래프 그리기
  - $0^\circ$ 에서  $360^\circ$ 까지  $10^\circ$  간격으로  $\sin$ ,  $\cos$ 을 그래프로 그려라.
  - x축의 단위는 각도로써 degree 단위로 설정한다.
  - 세 함수의 결과를 하나의 그래프로 그리되, 선 모양을 다르게 설정한다.
- 지난 시간에 생성한 삼각함수 테이블을 읽어서 그래프로 나타내어라

# 2차원 그래프 #2

# 2차원 그래프

- plot() 함수의 속성 설정

- 속성(property) : 그래프의 다양한 속성들을 지정할 수 있음
- 속성 + 속성 값의 형식으로 인수들을 입력

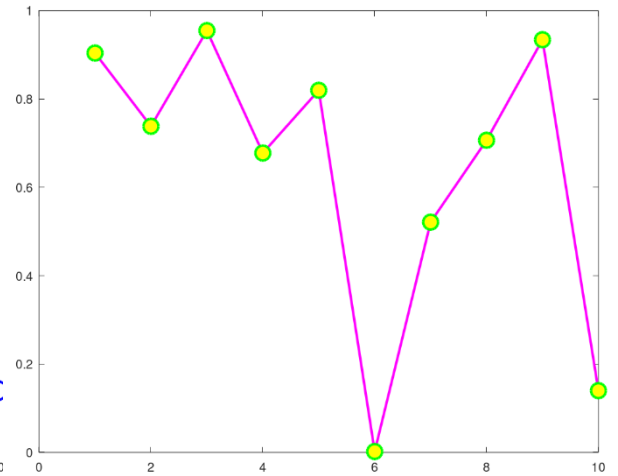
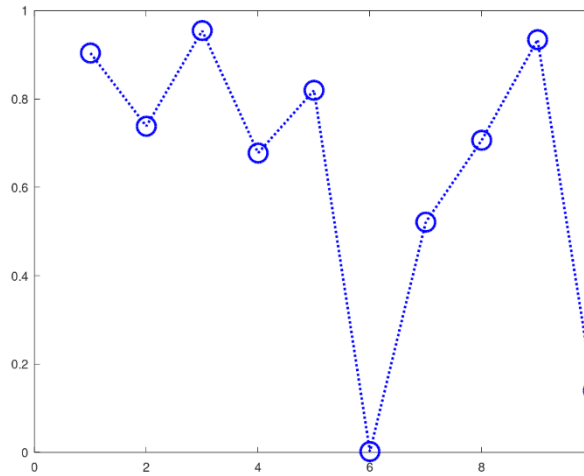
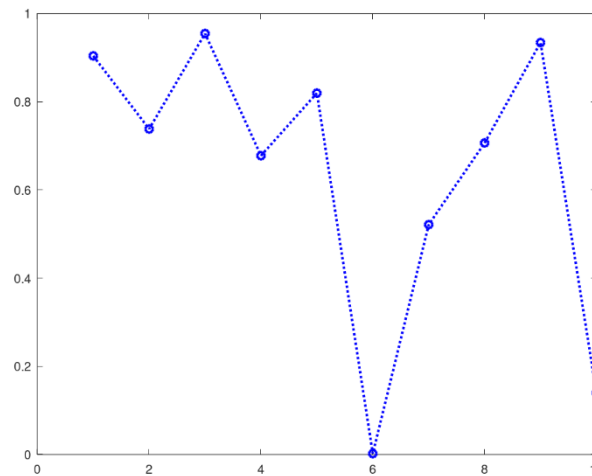
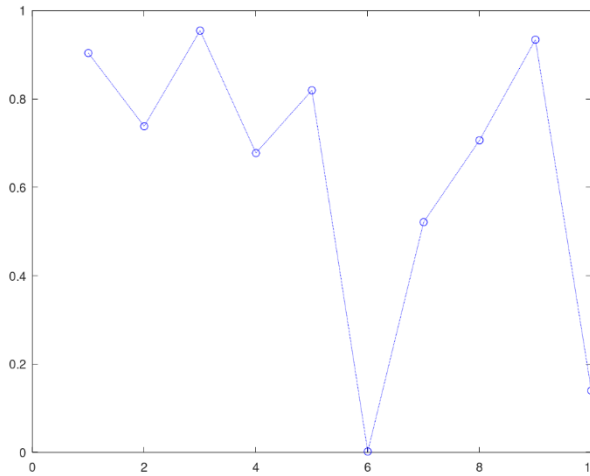
➤ 예] `plot(x, y, 'LineWidth', 2);` ⇒ 선의 굵기를 2로 지정 (기본 : 0.5)

Property	설명	속성값
LineWidth	선의 굵기	포인트 단위(0.5)
MarkerSize	표식의 크기	포인트 단위
MarkerEdgeColor	표식의 색 (테두리)	문자열 표시 (rgb..)
MarkerFaceColor	표식의 배경색	문자열 표시 (rgb..)

# 2차원 그래프

- plot() 함수의 속성 설정

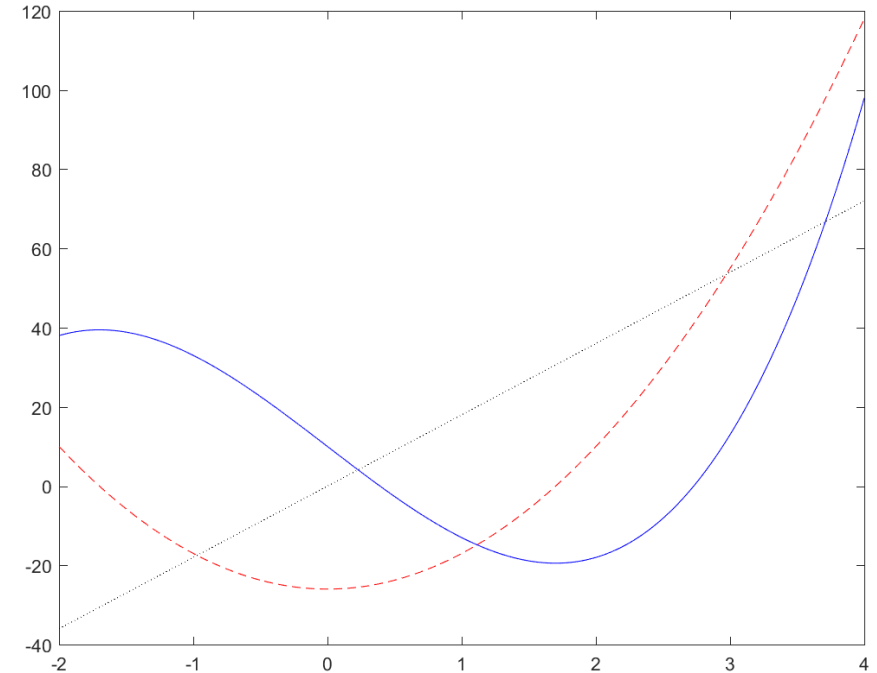
```
>> plot(x, y, 'bo:');  
>> plot(x, y, 'bo:', 'LineWidth', 2);  
>> plot(x, y, 'bo:', 'LineWidth', 2, 'MarkerSize', 5);  
>> plot(x, y, 'bo:', 'LineWidth', 2, 'MarkerSize', 15);  
>> plot(x, y, '-mo', 'LineWidth', 2, 'MarkerSize', 12, 'MarkerEdgeColor', 'g',  
'MarkerFaceColor', 'y')
```



# 2차원 그래프

- 2개 이상의 그래프 그리기
  - `plot(x1, y1, x2, y2);`
  - `plot(x1, y1, s1, x2, y2, s2);`
  - $y = 3x^3 - 26x + 10$ 의 함수와  $y'$ ,  $y''$ 의 함수를 같이 그려라.
    - $y' = 9x^2 - 26$ ,  $y'' = 18x$

```
>> x = -2:0.01:4;  
>> y = 3 * x.^3 - 26 * x + 10;  
>> y1 = 9 * x.^2 - 26;  
>> y2 = 18 * x;  
>> plot(x, y, x, y1);  
>> plot(x, y, '--', x, y1, 'g:');  
>> plot(x, y, '--', x, y1, 'm:');  
>> plot(x, y, 'b-', x, y1, 'r--', x, y2, 'k:');
```



# 2차원 그래프

- 2개 이상의 그래프 그리기

- hold 명령어 (hold on, hold off, hold : 토글방식)

- 여러 개의 그래프를 한 그림에 나타낼 때 사용

- hold on이 되어 있는 경우에 새로운 그래프를 그려도 이전의 그래프가 남아있음

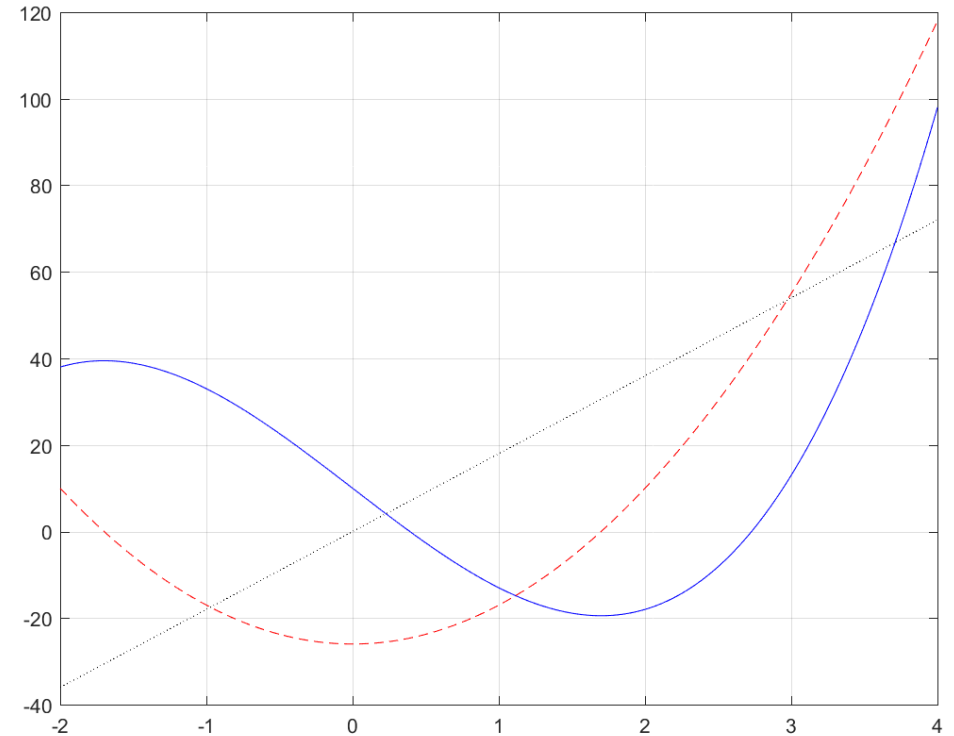
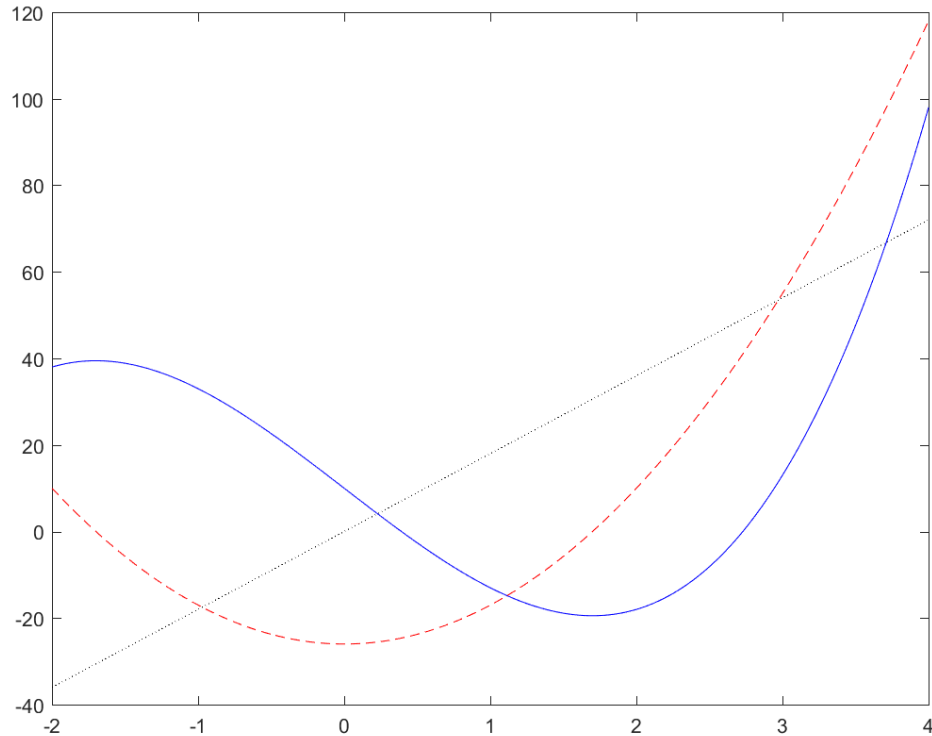
- hold off 되어 있는 경우, 새로운 그래프를 그리면 이전의 그래프는 지워짐

```
>> x = -2:0.01:4;  
>> y = 3 * x.^3 - 26 * x + 10;  
>> y1 = 9 * x.^2 - 26;  
>> y2 = 18 * x;  
>> plot(x, y, x, y1);  
>> plot(x, y, '--', x, y1, 'g:');  
  
>> plot(x, y, '--');  
>> hold on;  
>> plot(x, y1, 'g:');
```



# 2차원 그래프

- grid : 격자선 출력 (grid on, grid off, grid)



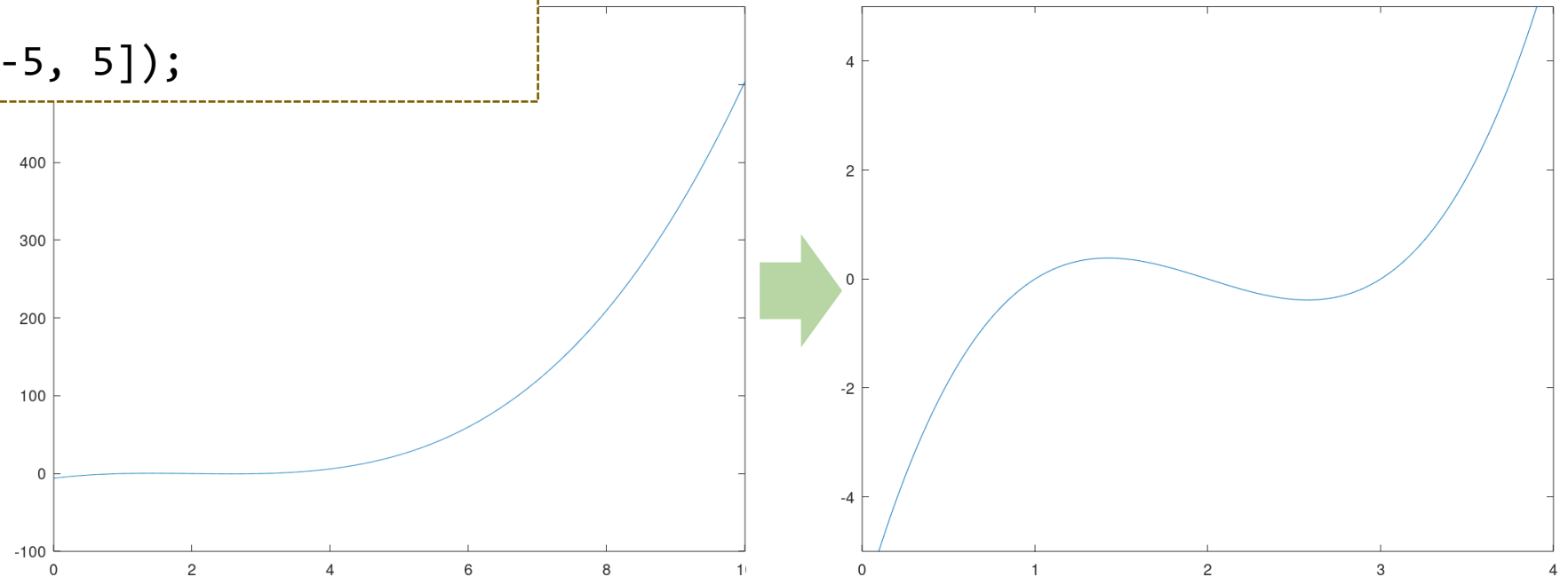
# 2차원 그래프

- axis 함수

- 축의 범위와 모양을 변경할 때 사용

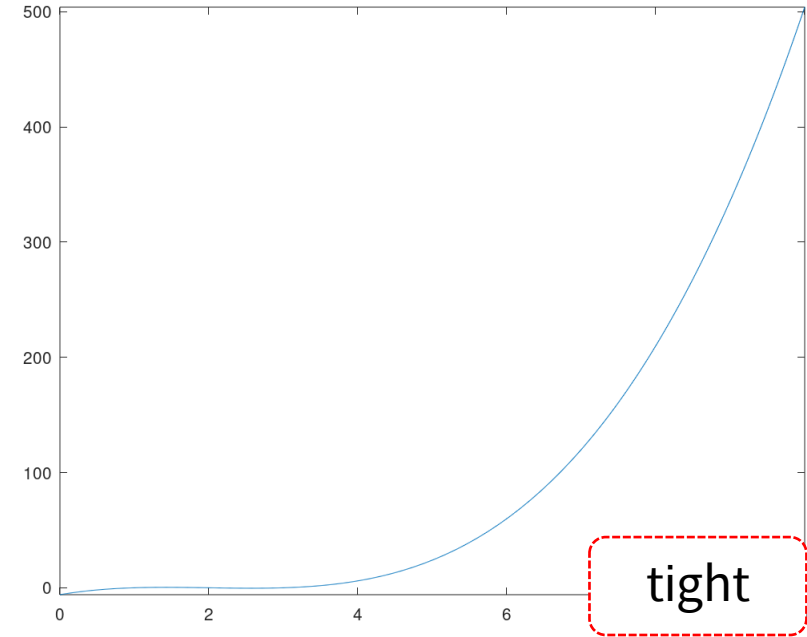
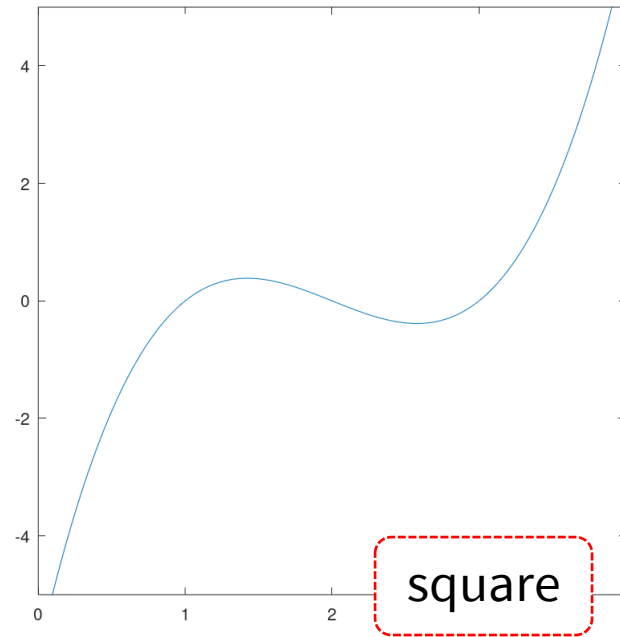
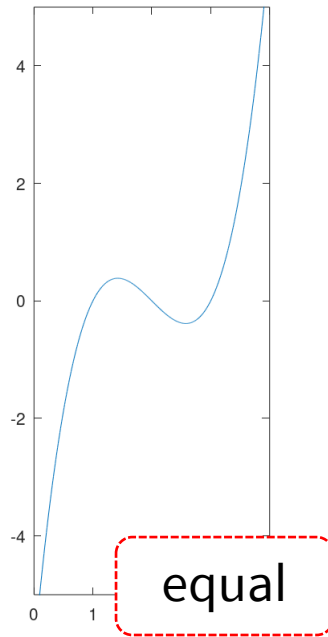
- `axis([xmin xmax ymin ymax]);` ⇒ 그래프의 범위를 재지정

```
>> x = 0:0.01:10;  
>> y = x.^3 - 6 * x.^2 + 11 * x - 6;  
>> plot(x, y);  
>> axis([0, 4, -5, 5]);
```



# 2차원 그래프

- axis 함수
  - axis equal : x축과 y축의 배율을 동일하게 설정
  - axis square : 좌표축의 영역을 정사각형으로 지정
  - axis tight : 좌표축의 범위를 데이터가 있는 영역으로 지정 (여백, 잘림 없음)



# 그래프 저장

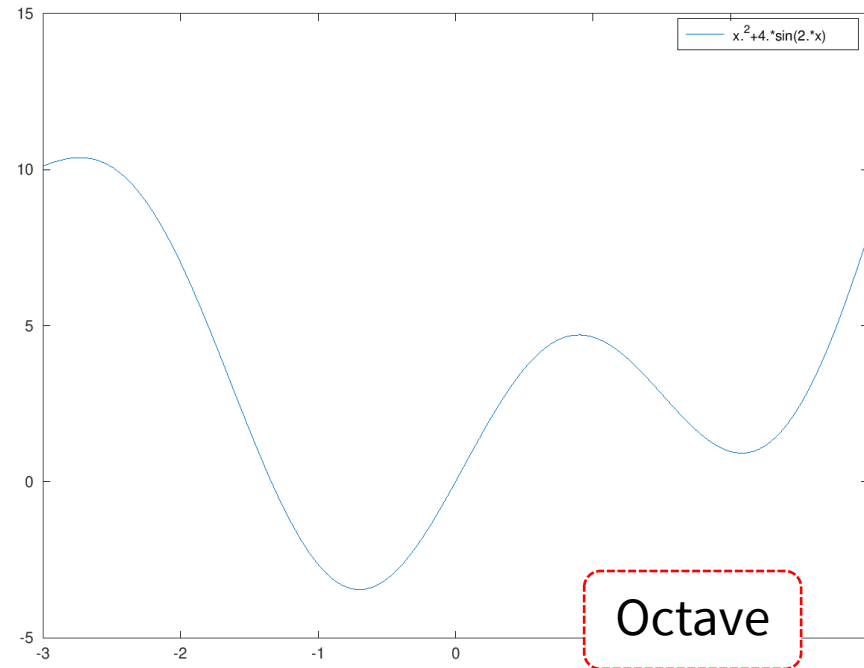
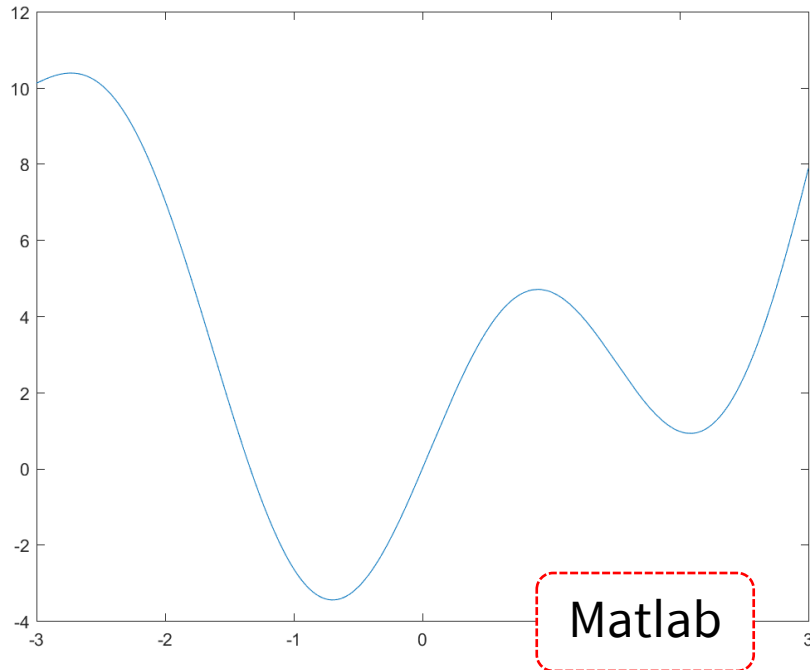
- print() 함수 : 이미지 파일로 저장
  - print -d<형식> <파일이름>
  - print -djpeg ex.jpg ⇒ jpg 파일 형식으로 저장
  - print -dpng ex.png ⇒ png 파일 형식으로 저장
  - jpg 파일 : 손실 저장, 사진 등에 권장
  - png 파일 : 무손실 저장, 그래프 도표 등에 권장
  - bmp 파일 : 무손실, 무압축, 권장하지 않음
- Matlab 함수의 인수 처리 방식
  - 함수명 인수1 인수2 ... ⇒ 함수명('인수1', '인수2', ...)
    - 공백 후의 문장은 단어 단위로 문자열로 취급하여 인수로 변환
    - print -dpng ex.png ⇒ print('-dpng', 'ex.png')

# 그래프 저장

- print() 함수 : 이미지 파일로 저장
  - print -d<형식> <파일이름>
  - 가능 형식
    - Matlab
      - ✓ Windows 전용 : win, winc, meta, bitmap
      - ✓ 범용 : ps, psc, ps2, psc2, eps, epsc, eps2, epsc2, pdf svg, jpg, jpeg<nn>, tiff, tiffnocompression, png, bmpmono, bmp256, bmp16m, pcxmono, pcx16, pcx256, pcx24b, bpm, bpmraw, pgm, pgmraw, ppm, ppmraw
    - Octave
      - ✓ pdf, pdfcrop, ps, ps2, psc, psc2, eps, eps2, epsc, epsc2, pslatex, epslatex, pdflatex, pslatexstandalone, epslatexstandalone, pdflatexstandalone, epscairo\*, pdfcario\*, epscairolatex\*, pdfcairolatex\*, epscairolatexstandalone\*, pdfcairolatexstandalone\*, svg, canvas\*, cdr\*, corel\*, cgm\*, dxf, emf, meta, fig, hpgl, ill, aifm, latex\*, eepic\*, mf\*, tokz, tokzstandalone, png, jpg, jpeg, tif, tiff, tiffn, gif, pbm, dumb\*

# 함수 그리기

- fplot : 주어진 함수를 주어진 정의역에서 그리기
  - fplot('함수', [정의역의 범위]);
  - fplot('x^2+4\*sin(2\*x)', [-3 3])
  - print -dpng ex.png



# 그래프 꾸미기

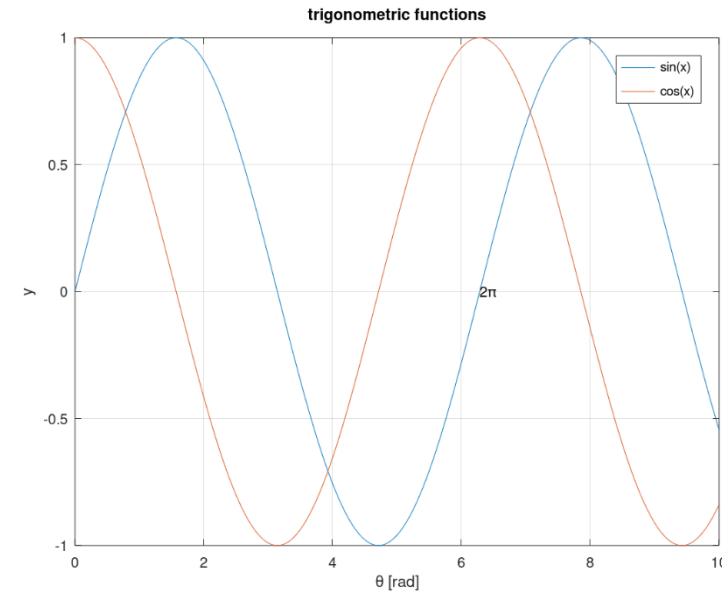
- 그래프 꾸미기
  - title('제목')
  - xlabel('x 축 제목')
  - ylabel('y 축 제목')
  - legend('제목1', '제목2', '제목3', ...): 범례
  - text(x, y, '그래프 위의 문자열')

```
plot(x, ys, x, yc);  
grid on;  
title('trigonometric functions');  
xlabel('\theta [rad]');  
ylabel('y');  
legend('sin(x)', 'cos(x)');  
text(2 * pi, 0, '2\pi');
```

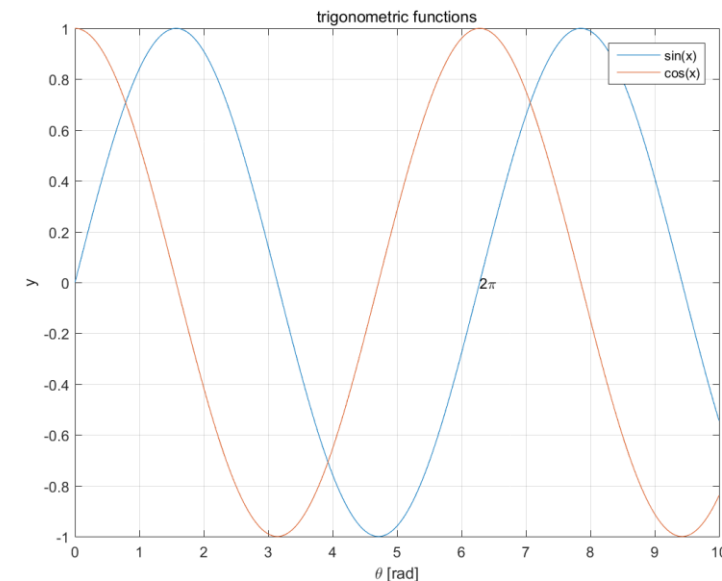
# 그래프 꾸미기

- 꾸미기 예제

```
x = 0:0.01:10;  
ys = sin(x);  
yc = cos(x);  
  
plot(x, ys, x, yc);  
grid on;  
xlabel('\theta [rad]');  
ylabel('y');  
title('trigonometric functions');  
legend('sin(x)', 'cos(x)');  
text(2 * pi, 0, '2\pi');  
print -dpng sincos.png
```



Octave



Matlab



# 그래프 꾸미기

- 텍스트 형식 (LaTeX-style)

- title, xlabel, ylabel, legend, text 등에서 텍스트 내부에 사용되는 형식지정
- 그리스 문자, 대문자로 시작하는 경우 그리스어 대문자 (예: `\Gamma` ⇒  $\Gamma$ )
- 함수의 추가 인수로 속성(property)를 지정

➤ `text(2*pi, 0, '2\pi', 'fontsize', 20)`

형식지정	내용
<code>\bf</code>	bold face
<code>\it</code>	Italic
<code>\rm</code>	Roman
<code>_</code>	Subscript
<code>^</code>	Superscript
속성	내용
fontname	폰트 설정
fontsize	글자 크기
color	글자 색상

형식지정	내용	형식지정	내용	형식지정	내용
<code>\alpha</code>	$\alpha$	<code>\vartheta</code>	$\vartheta$	<code>\sigma</code>	$\sigma$
<code>\beta</code>	$\beta$	<code>\iota</code>	$\iota$	<code>\varsigma</code>	$\varsigma$
<code>\gamma</code>	$\gamma$	<code>\kappa</code>	$\kappa$	<code>\tau</code>	$\tau$
<code>\delta</code>	$\delta$	<code>\mu</code>	$\mu$	<code>\upsilon</code>	$\upsilon$
<code>\epsilon</code>	$\epsilon$	<code>\nu</code>	$\nu$	<code>\chi</code>	$\chi$
<code>\zeta</code>	$\zeta$	<code>\xi</code>	$\xi$	<code>\psi</code>	$\psi$
<code>\eta</code>	$\eta$	<code>\pi</code>	$\pi$	<code>\omega</code>	$\omega$
<code>\theta</code>	$\theta$	<code>\rho</code>	$\rho$	<code>\varphi</code>	$\varphi$
<code>\varpi</code>	$\varpi$	<code>\lambda</code>	$\lambda$	<code>\varepsilon</code>	$\varepsilon$

# 실습 과제

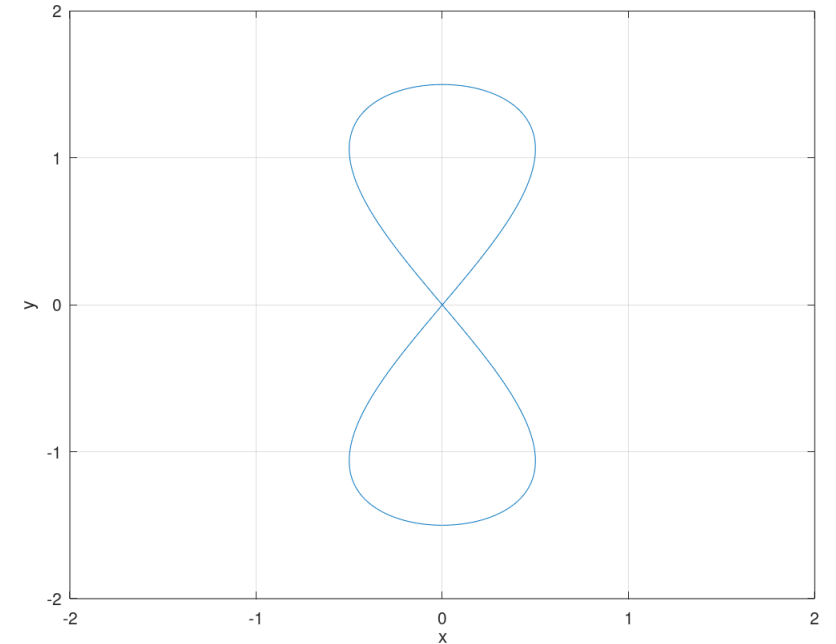
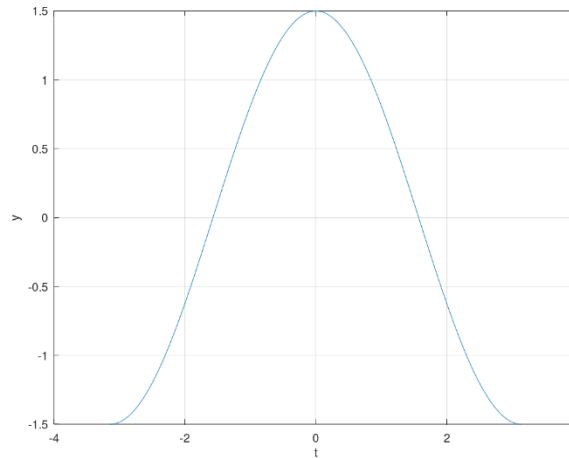
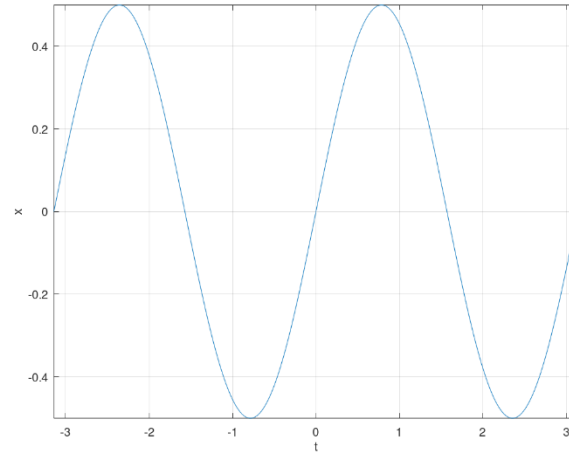
- 수업시간에 실습한 내용을 캡처해서 제출하시오. 그래프의 속성을 다양하게 변화시킨 경우에는 대표적인 그림만 2~3장 캡처해서 제출하면 되며, 그래프는 print 함수를 사용하여 저장한 후 제출하기 바람.
- 다음의 그래프를 그리는 스크립트 파일을 만들어서 수행하고, 스크립트 파일과 결과 그래프를 제출하시오.
  - $f(x) = (3 \cos x - \sin x)e^{-0.1x}$ ,  $-4 \leq x \leq 9$ , 그래프 그리기
  - $f(x) = x^3 - 2x^2 - 10 \sin^2 x - e^{0.9x}$ ,  $-2 \leq x \leq 4$ ,  $f(x)$ 와  $f'(x)$ 를 하나의 그래프에 그리기.

# 매개변수 방정식

- $x$ 와  $y$ 가 다른 변수의 함수로 구성되는 경우

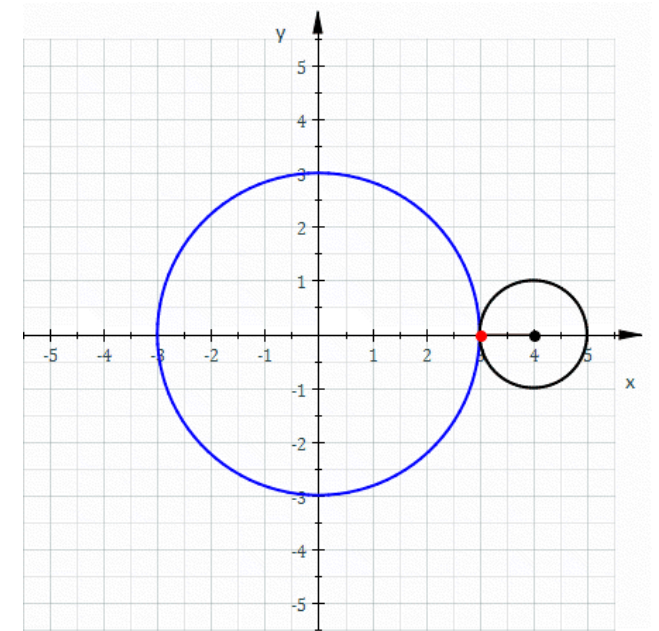
예]  $x = \sin t \cdot \cos t$ ,  $y = 1.5 \cos t$ ,  $-\pi \leq t \leq \pi$

```
>> figure;  
>> plot(t, x);  
>> xlabel('t');  
>> ylabel('x');  
>> axis tight;  
  
>> figure;  
>> plot(t, y);  
>> xlabel('t');  
>> ylabel('y');  
  
>> figure;  
>> plot(x, y);  
>> xlabel('x');  
>> ylabel('y');  
>> axis([-2 2 -2 2]);  
>> grid;
```



# 매개변수 방정식의 예

- 에피사이클로이드(Epicycloid)
  - 서로 외접하는 원이 있을 때 외부의 원이 내부의 원과 접하면서 구르는 경우, 외부 원의 정점의 궤적
  - 참고 : <https://suhak.tistory.com/687>
- 사이클로이드(Cycloid)
  - 직선 위로 원을 굴렸을 때 원의 정점이 그리는 곡선
  - 참고 : <https://ko.wikipedia.org/wiki/사이클로이드>



# 매개변수 방정식의 예

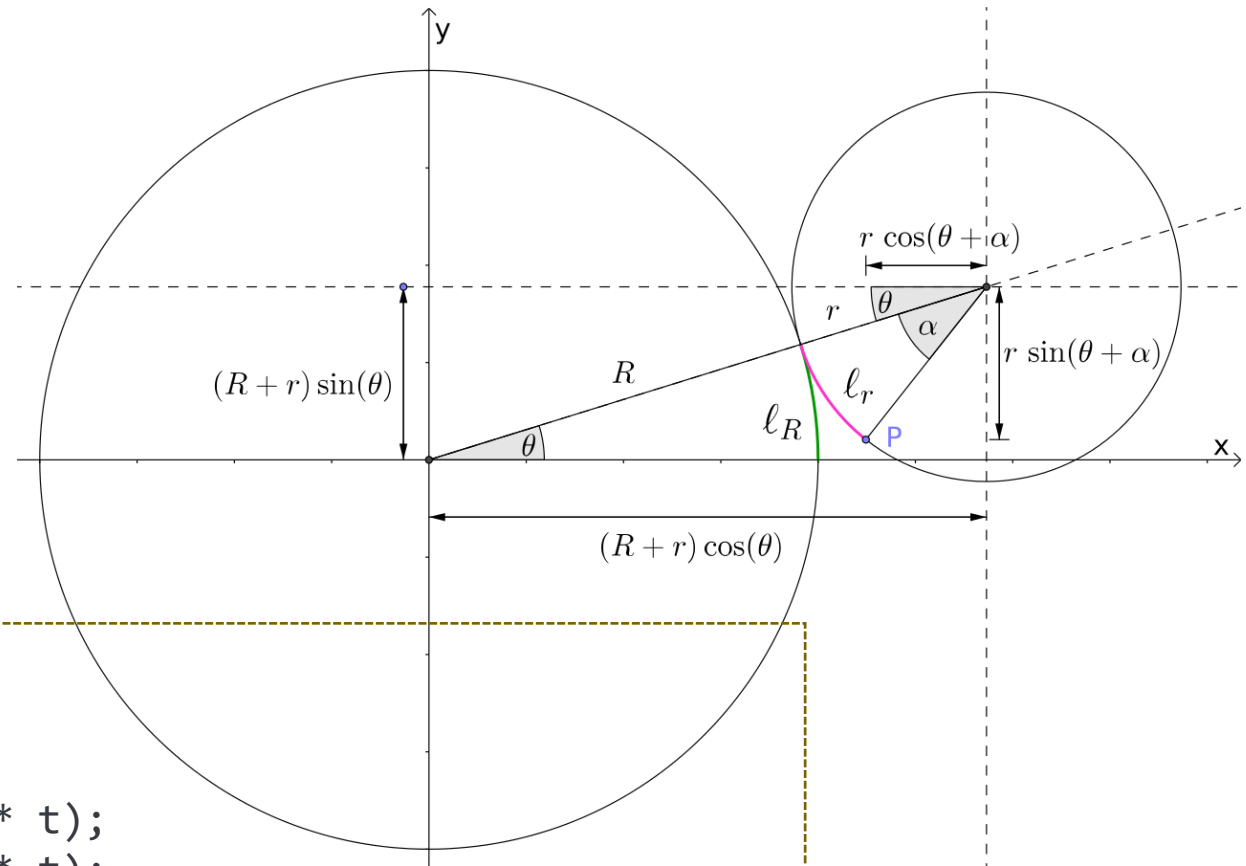
## • 에피사이클로이드(Epicycloid)

• 참고: <https://suhak.tistory.com/687>

$$\triangleright l_R = R\theta = r\alpha = l_r, \quad \alpha = \frac{R\theta}{r}$$

$$\begin{aligned} \checkmark x(\theta) &= (R + r) \cos \theta - r \cos(\theta + \alpha) \\ &= (R + r) \cos \theta - r \cos\left(\frac{R+r}{r} \theta\right) \end{aligned}$$

$$\begin{aligned} \checkmark y(\theta) &= (R + r) \sin \theta - r \sin(\theta + \alpha) \\ &= (R + r) \sin \theta - r \sin\left(\frac{R+r}{r} \theta\right) \end{aligned}$$



```
R = 11;
r = 2;
t = 0:0.01:4*pi;
x = (R + r) * cos(t) - r * cos((R + r) / r * t);
y = (R + r) * sin(t) - r * sin((R + r) / r * t);

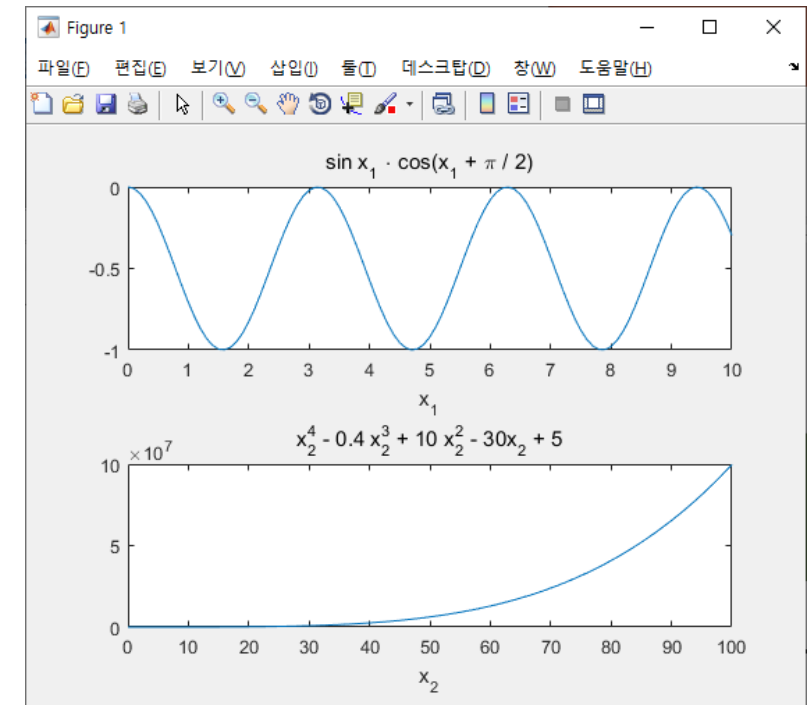
plot(x, y);
axis equal;
grid;
title(sprintf('Epicycloid, {\it R} = %g, {\it r} = %g', R, r));
```

# 2차원 그래프

- subplot : 여러 개의 그래프를 하나의 윈도우에 그리기

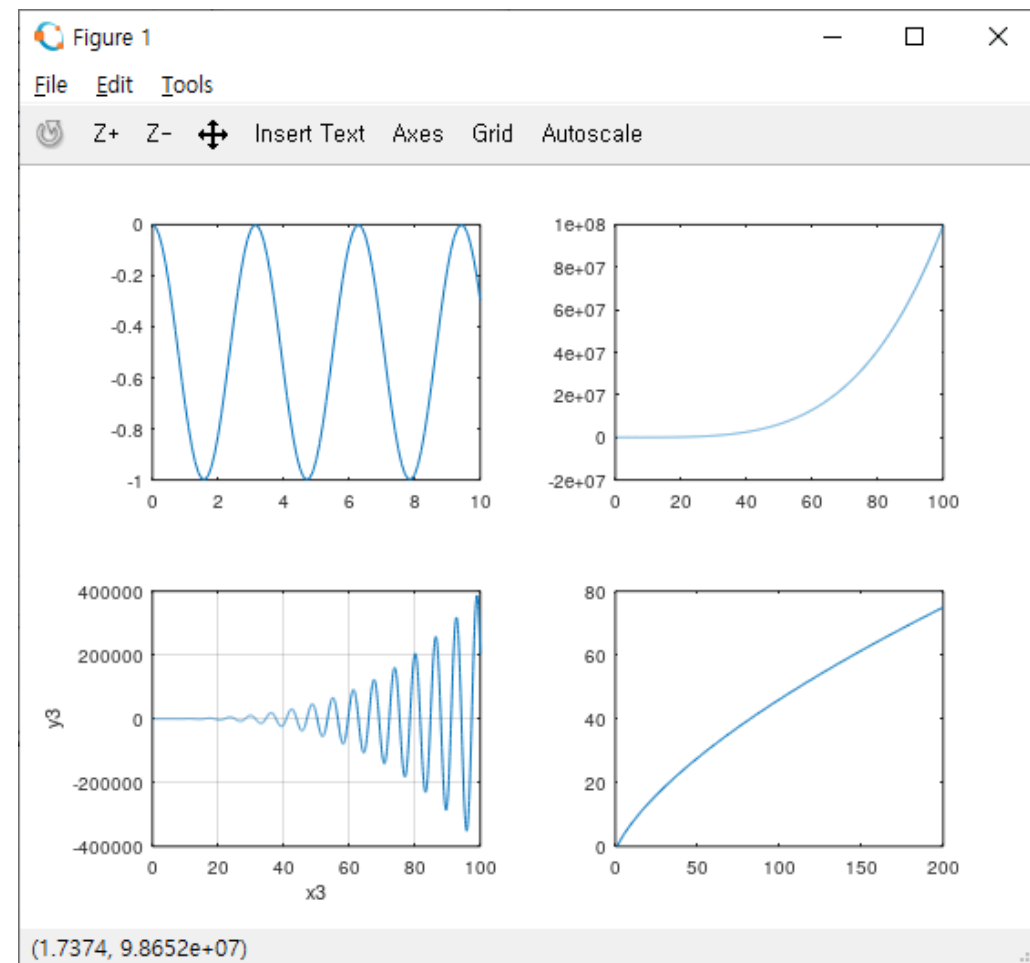
Form	Meaning
subplot(m, n, i)	Breaks the Figure window into an m X n matrix of small axis and selects the i'th axis for the current plot
subplot(mni)	

```
x1 = 0:0.1:10;  
f1 = sin(x1) .* cos(x1 + pi / 2);  
subplot(221);  
plot(x1, f1);  
xlabel('x_1');  
title('sin x_1 \cdot cos(x_1 + \pi / 2)');  
  
x2 = 0:0.1:100;  
f2 = x2.^4 - 0.4*x2.^3 + 10*x2.^2 - 30*x2 + 5;  
subplot(222);  
plot(x2, f2);  
xlabel('x_2');  
title('x_2^4 - 0.4 x_2^3 + 10 x_2^2 - 30x_2 + 5');
```



# 2차원 그래프

```
>> x1 = 0:0.1:10;  
>> f1 = sin(x1) .* cos(x1 + pi / 2);  
>> subplot(221);  
>> plot(x1, f1);  
  
>> x2 = 0:0.1:100;  
>> f2 = x2.^4 - 0.4*x2.^3 + 10*x2.^2 - 30*x2 + 5;  
>> subplot(222);  
>> plot(x2, f2);  
  
>> x3 = 0:0.1:100;  
>> f3 = -(0.4*x3.^3).*sin(x3);  
>> subplot(223);  
>> plot(x3, f3);  
>> grid on;  
>> xlabel('x3');  
>> ylabel('y3');  
  
>> x4 = 1:1:200;  
>> f4 = sqrt(x4).*log(x4);  
>> subplot(224);  
>> plot(x4, f4);
```



## 2차원 그래프 연습

- 2x2의 subplot을 만들어서, 각각 sin, cos, tan 함수의 그래프를 그리고, 마지막 그래프에는 세 함수를 한번에 나타내는 스크립트를 완성하라.



# 로그 축 및 기타 형식의 그래프

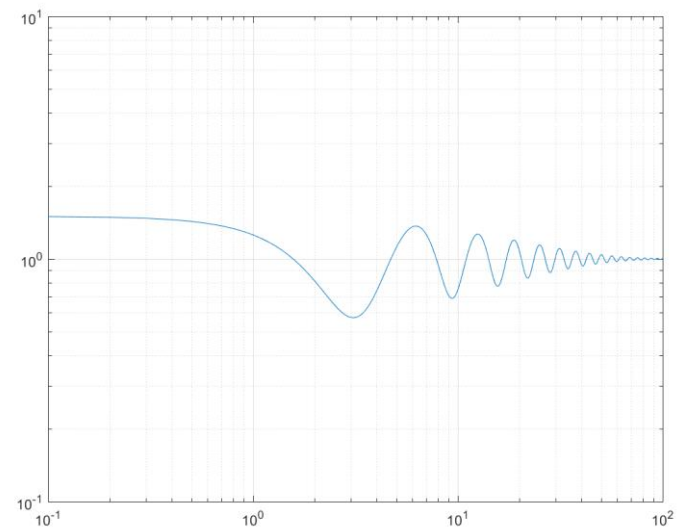
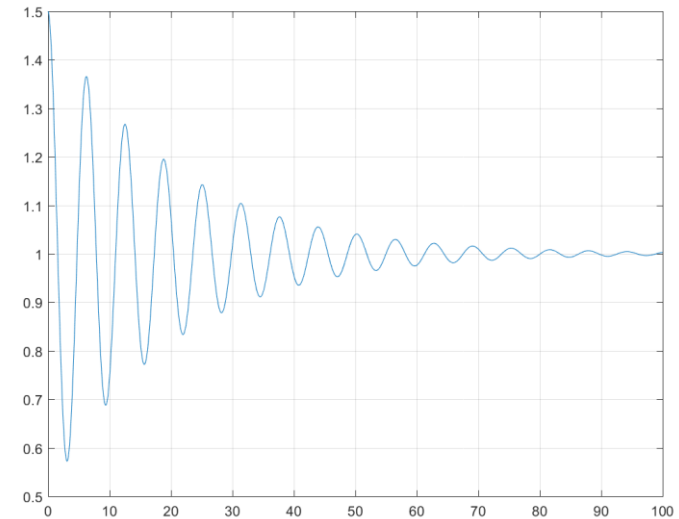
Function	Description
<code>Plot(x1, y1, s1, x2, y2, s2, ... )</code>	Combines the plots defined by the (x, y, s) triples
<code>loglog(x1, y1, s1, x2, y2, s2, ... )</code>	<code>loglog(...)</code> is the same as <code>plot(...)</code> except a logarithmic (base 10) scale is used for the X-axis and the Y-axis
<code>semilogx(x1, y1, s1, x2, y2, s2, ...)</code>	<code>semilogx(...)</code> is the same as <code>plot(...)</code> except a logarithmic (base 10) scale is used for the X-axis
<code>semilogy(x1, y1, s1, x2, y2, s2, ...)</code>	<code>semilogy(...)</code> is the same as <code>plot(...)</code> except a logarithmic (base 10) scale is used for the Y-axis
<code>plotyy(x1, y1, x2, y2)</code>	Plots Y1 versus X1 with y-axis labeling on the left and plots Y2 versus X2 with y-axis labeling on the right

# 로그 축 및 기타 형식

- $x = 1 + \frac{1}{2}e^{-0.05t} \sin\left(t + \frac{\pi}{2}\right), 0 \leq t \leq 100, 0.1 \leq x \leq 10$

```
>> t=0:0.1:100;  
>> x=1 + 0.5*exp(-0.05*t) .* sin(t + pi/2);  
>> plot(t, x);  
>> grid on
```

```
>> loglog(t, x);  
>> axis([0, 100, 0.1, 10]);  
>> grid on
```

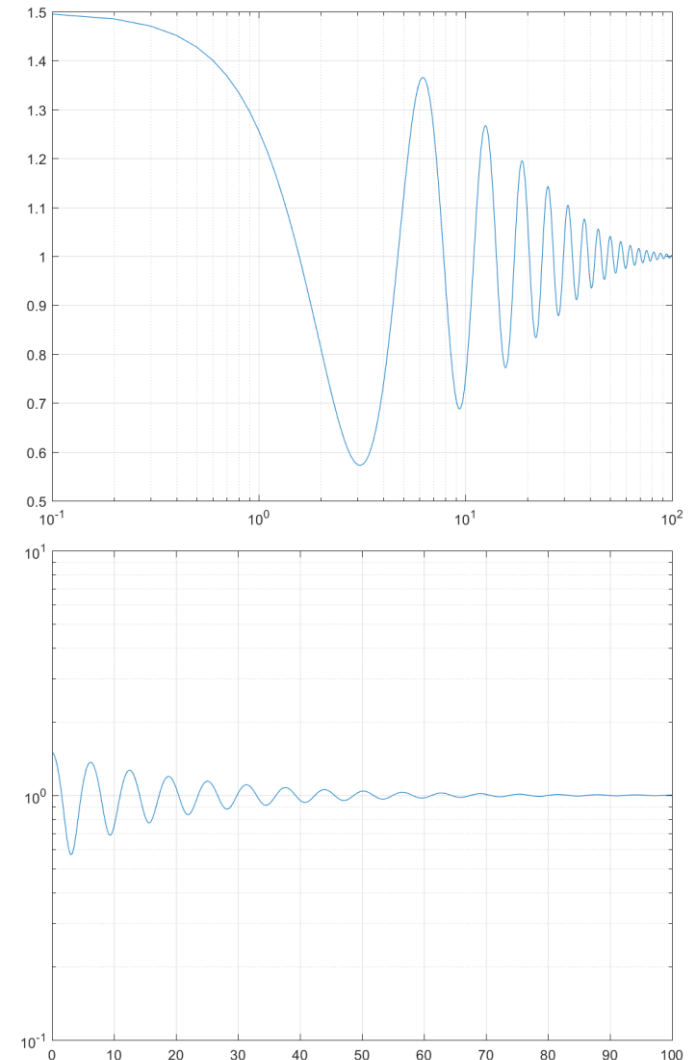


# 로그 축 및 기타 형식

- $x = 1 + \frac{1}{2}e^{-0.05t} \sin\left(t + \frac{\pi}{2}\right), 0 \leq t \leq 100, 0.1 \leq x \leq 10$

```
>> t=0:0.1:100;  
>> x=1 + 0.5*exp(-0.05*t) .* sin(t + pi/2);  
>> semilogx(t, x);  
>> grid on
```

```
>> semilogy(t, x);  
>> axis([0, 100, 0.1, 10]);  
>> grid on
```



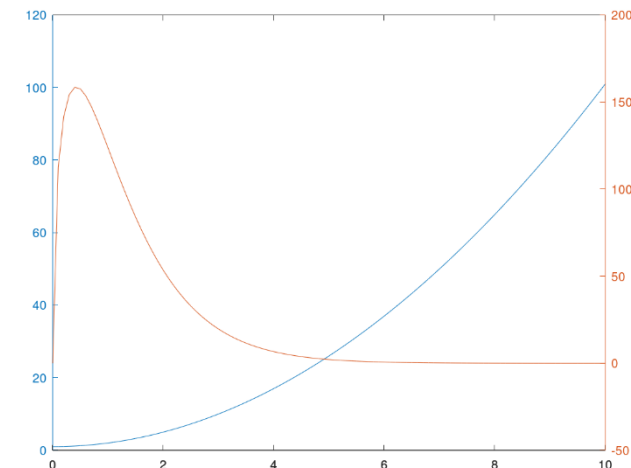
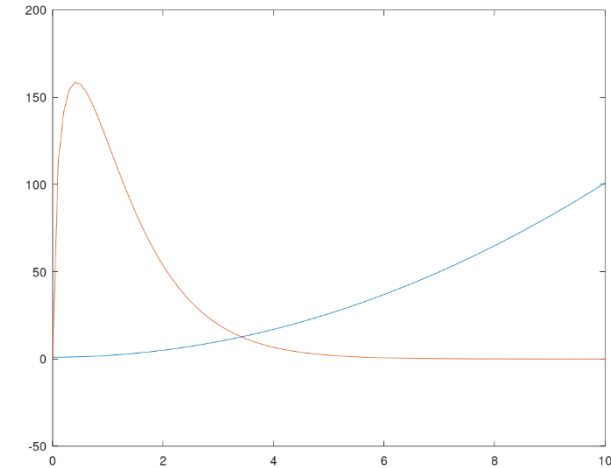
# 로그 축 및 기타 형식

- $y = x^2 + 1$ ,  $y = 400 \frac{\sin \sqrt{x}}{e^x}$ , where  $0 \leq x \leq 10$

```
>> x = 0:0.1:10;  
>> y1 = x.^2 + 1;  
>> y2 = 400 * sin(sqrt(x)) ./ exp(x);  
>> plot(x, y1, x, y2);
```

- plotyy() : 두 개의 그래프, y축 별도

```
>> x = 0:0.1:10;  
>> y1 = x.^2 + 1;  
>> y2 = 400 * sin(sqrt(x)) ./ exp(x);  
>> plotyy(x, y1, x, y2);
```



# 좌표 축 속성 설정하기

- `set(gca, 'property', 'value', 'property', 'value', ...)`
  - `set()` 함수 : 주어진 개체(첫번째 인수)의 속성들을 설정
    - 개체 : 그래프 윈도우, 좌표축, 하나의 데이터 선(line) 등
    - `F = gcf();` ⇒ 현재 그래프 윈도우 가져오기(get current figure)
    - `A = gca();` ⇒ 현재 그래프 좌표축 가져오기 (get current axes) → 가장 빈번히 사용됨
    - `L = plot(x, y);` ⇒ 그래프에 선을 추가하고 그 개체를 가져오기
  - `Get()` 함수 : 현재의 속성을 가져 오, 속성이 없는 경우 전체 속성 표시
    - 속성 참고
      - ✓ Axes 속성 : <https://kr.mathworks.com/help/matlab/ref/matlab.graphics.axis.axes-properties.html>
      - ✓ Figure 속성 : <https://kr.mathworks.com/help/matlab/ref/matlab.ui.figure-properties.html>
      - ✓ Line 속성 : <https://kr.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.line-properties.html>

# 좌표 축 속성 설정하기 (Matlab Axes 속성예)

```
>> get(gca)
```

```
    ALim: [0 1]
    ALimMode: 'auto'
    ActivePositionProperty: 'outerposition'
    AmbientLightColor: [1 1 1]
    BeingDeleted: 'off'
    Box: 'on'
    BoxStyle: 'back'
    BusyAction: 'queue'
    ButtonDownFcn: ''
    CLim: [0 1]
    CLimMode: 'auto'
    CameraPosition: [0.1638 -4.7007e-04 241.0890]
    CameraPositionMode: 'auto'
    CameraTarget: [0.1638 -4.7007e-04 0]
    CameraTargetMode: 'auto'
    CameraUpVector: [0 1 0]
    CameraUpVectorMode: 'auto'
    CameraViewAngle: 8.9714
    CameraViewAngleMode: 'auto'
    Children: [1x1 Line]
    Clipping: 'on'
    ClippingStyle: '3dbox'
    Color: [1 1 1]
    ColorOrder: [7x3 double]
    ColorOrderIndex: 2
    CreateFcn: ''
    CurrentPoint: [2x3 double]
    DataAspectRatio: [1 1 1]
    DataAspectRatioMode: 'manual'
    DeleteFcn: ''
    FontAngle: 'normal'
    FontName: 'Helvetica'
    FontSize: 10
    FontSmoothing: 'on'
    FontUnits: 'points'
    FontWeight: 'normal'
    GridAlpha: 0.1500
    GridAlphaMode: 'auto'
    GridColor: [0.1500 0.1500 0.1500]
    GridColorMode: 'auto'
    GridLineStyle: '-'
    HandleVisibility: 'on'
    HitTest: 'on'
```

```
    Interruptible: 'on'
    LabelFontSizeMultiplier: 1.1000
    Layer: 'bottom'
    LineStyleOrder: '-'
    LineStyleOrderIndex: 1
    LineWidth: 0.5000
    MinorGridAlpha: 0.2500
    MinorGridAlphaMode: 'auto'
    MinorGridColor: [0.1000 0.1000 0.1000]
    MinorGridColorMode: 'auto'
    MinorGridLineStyle: ':'
    NextPlot: 'replace'
    OuterPosition: [0 0 1 1]
    Parent: [1x1 Figure]
    PickableParts: 'visible'
    PlotBoxAspectRatio: [434 342.3000 22.8200]
    PlotBoxAspectRatioMode: 'manual'
    Position: [0.1300 0.1100 0.7750 0.8150]
    Projection: 'orthographic'
    Selected: 'off'
    SelectionHighlight: 'on'
    SortMethod: 'childorder'
    Tag: ''
    TickDir: 'in'
    TickDirMode: 'auto'
    TickLabelInterpreter: 'tex'
    TickLength: [0.0100 0.0250]
    TightInset: [0.0435 0.0523 0 0.0555]
    Title: [1x1 Text]
    TitleFontSizeMultiplier: 1.1000
    TitleFontWeight: 'normal'
    Type: 'axes'
    UIContextMenu: [0x0 GraphicsPlaceholder]
    Units: 'normalized'
    UserData: []
    View: [0 90]
    Visible: 'on'
    XAxis: [1x1 NumericRuler]
    XAxisLocation: 'bottom'
    XColor: [0.1500 0.1500 0.1500]
    XColorMode: 'auto'
    XDir: 'normal'
    XGrid: 'on'
    XLabel: [1x1 Text]
```

```
    XLim: [-18.7498 19.0773]
    XLimMode: 'auto'
    XMinorGrid: 'off'
    XMinorTick: 'off'
    XScale: 'linear'
    XTick: [-15 -10 -5 0 5 10 15]
    XTickLabel: {7x1 cell}
    XTickLabelMode: 'auto'
    XTickLabelRotation: 0
    XTickMode: 'auto'
    YAxis: [1x1 NumericRuler]
    YAxisLocation: 'left'
    YColor: [0.1500 0.1500 0.1500]
    YColorMode: 'auto'
    YDir: 'normal'
    YGrid: 'on'
    YLabel: [1x1 Text]
    YLim: [-14.9178 14.9169]
    YLimMode: 'auto'
    YMinorGrid: 'off'
    YMinorTick: 'off'
    YScale: 'linear'
    YTick: [-10 -5 0 5 10]
    YTickLabel: {5x1 cell}
    YTickLabelMode: 'auto'
    YTickLabelRotation: 0
    YTickMode: 'auto'
    ZAxis: [1x1 NumericRuler]
    ZColor: [0.1500 0.1500 0.1500]
    ZColorMode: 'auto'
    ZDir: 'normal'
    ZGrid: 'on'
    ZLabel: [1x1 Text]
    ZLim: [-0.9945 0.9945]
    ZLimMode: 'auto'
    ZMinorGrid: 'off'
    ZMinorTick: 'off'
    ZScale: 'linear'
    ZTick: [-0.5000 0 0.5000]
    ZTickLabel: ''
    ZTickLabelMode: 'auto'
    ZTickLabelRotation: 0
    ZTickMode: 'auto'
```

```
>>
```

# 좌표 축 속성 설정하기 (대표적인 속성)

```
CurrentPoint: [2x3 double]
DataAspectRatio: [1 1 1]
DataAspectRatioMode: 'manual'
FontAngle: 'normal'
FontName: 'Helvetica'
FontSize: 10
FontSmoothing: 'on'
FontUnits: 'points'
FontWeight: 'normal'
GridAlpha: 0.1500
GridAlphaMode: 'auto'
GridColor: [0.1500 0.1500 0.1500]
GridColorMode: 'auto'
GridLineStyle: '-'
LineStyleOrder: '-'
LineStyleOrderIndex: 1
LineWidth: 0.5000
MinorGridAlpha: 0.2500
MinorGridAlphaMode: 'auto'
MinorGridColor: [0.1000 0.1000 0.1000]
```

대부분의 속성들은 설정할 필요가 없으며, 세밀한 설정이 필요한 경우에만 사용함. 자주 사용되는 속성들은 다양한 그래픽 함수들(grid, axis, xlabel, ...)을 통해서 설정됨.

```
TickDir: 'in'
TickDirMode: 'auto'
TickLabelInterpreter: 'tex'
TickLength: [0.0100 0.0250]
Title: [1x1 Text]
TitleFontSizeMultiplier: 1.1000
TitleFontWeight: 'normal'
Type: 'axes'
Units: 'normalized'
View: [0 90]
XAxisLocation: 'bottom'
XColor: [0.1500 0.1500 0.1500]
XColorMode: 'auto'
XDir: 'normal'
XGrid: 'on'
XLabel: [1x1 Text]
XLim: [-18.7498 19.0773]
XLimMode: 'auto'
XMinorGrid: 'off'
XMinorTick: 'off'
XScale: 'linear'
XTick: [-15 -10 -5 0 5 10 15]
XTickLabel: {7x1 cell}
XTickLabelMode: 'auto'
XTickLabelRotation: 0
XTickMode: 'auto'
```

# 좌표 축 속성 설정하기

- 좌표 축 속성 변경의 예

```
>> set(gca, 'FontSize', 20);
```

```
>> set(gca, 'GridLineStyle', '--');
```

- 선의 속성 변경의 예

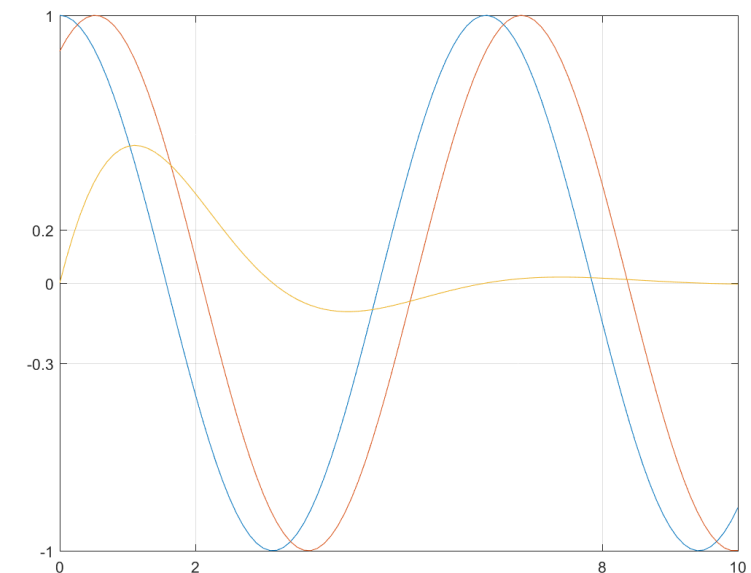
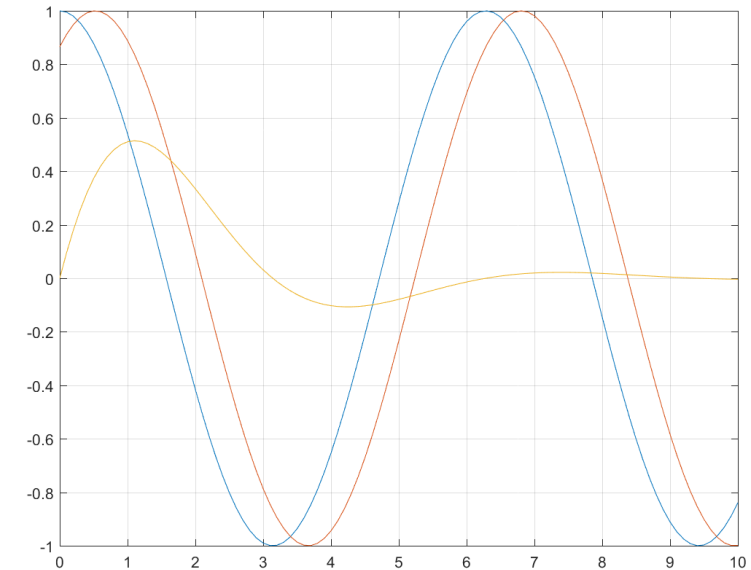
```
>> set(plot(x, y), 'LineStyle', '--');
```



# 좌표 축 속성 설정하기

```
>> x = 0:0.1:10;  
>> y1 = cos(x);  
>> y2 = cos(x - pi / 6);  
>> y3 = exp(-0.5*x) .* sin(x);  
>> plot(x, y1, x, y2, x, y3);  
>> grid on
```

```
>> xt = [0, 2, 4, 6, 8, 10];  
>> yt = [-1, -0.5, 0, 0.5, 1];  
>> set(gca, 'xtick', xt);  
>> set(gca, 'ytick', yt);
```



# 극좌표계 그래프

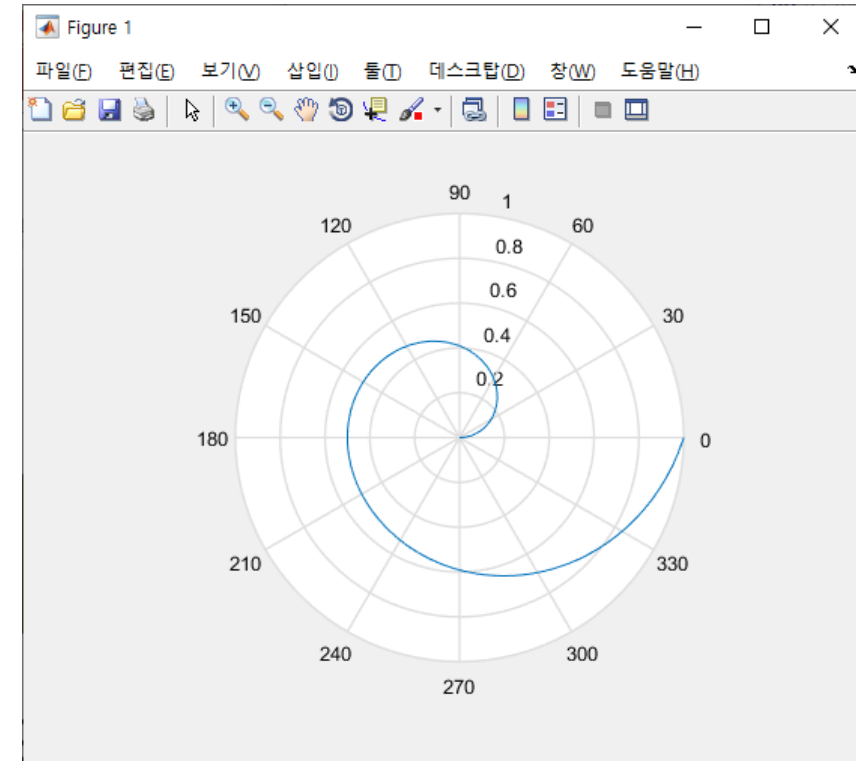
Function	Description
<code>polar(theta, r, s)</code>	Makes a plot using polar coordinates of the angle 'theta', in radians, versus the radius 'r'
<code>[theta, r] = cart2pol(x, y)</code>	Transforms the corresponding elements of data stored in Cartesian coordinates x, y to polar coordinates angle 'theta', radius 'r'
<code>[x, y] = pol2cart(theta, r)</code>	Transforms the corresponding elements of data stored in polar coordinates 'theta', radius 'r' to Cartesian coordinates x, y

# 극 좌표계 그래프

- $r = \frac{\sin \theta + \theta}{2\pi}, \quad 0 \leq \theta \leq 2\pi$

```
>> theta = 0:(2*pi)/100:2*pi;  
>> r = (sin(theta) + theta) / (2*pi);  
>> polar(theta, r);
```

```
>> [x, y] = pol2cart(theta, r);  
>> plot(x, y)  
>> grid  
>> axis equal  
>> axis([-1 1 -1 1]);
```



# 과제

- 수업에 수행한 실습 화면 및 결과 화면을 캡처해서 제출하시오.
- 중심이  $x = 3, y = 2$  이며 주축이  $a = 8, y = 6$ 인 타원의 그래프를 그려라.
- 임의의 그래프를 그리고, 좌표축 속성을 변경하는 5가지 예제를 만들고 결과 화면을 제출하시오.