

# Plotting in 2-Dimensional Space

**黃世強 (Sai-Keung Wong)**

*National Chiao Tung University, Taiwan*



# Plotting data

- Large sets of data are not easily to interpret as tables of numbers.
- Displaying data as graphs and animation help gain insights.
- Insightful trends and possible errors can be identified.

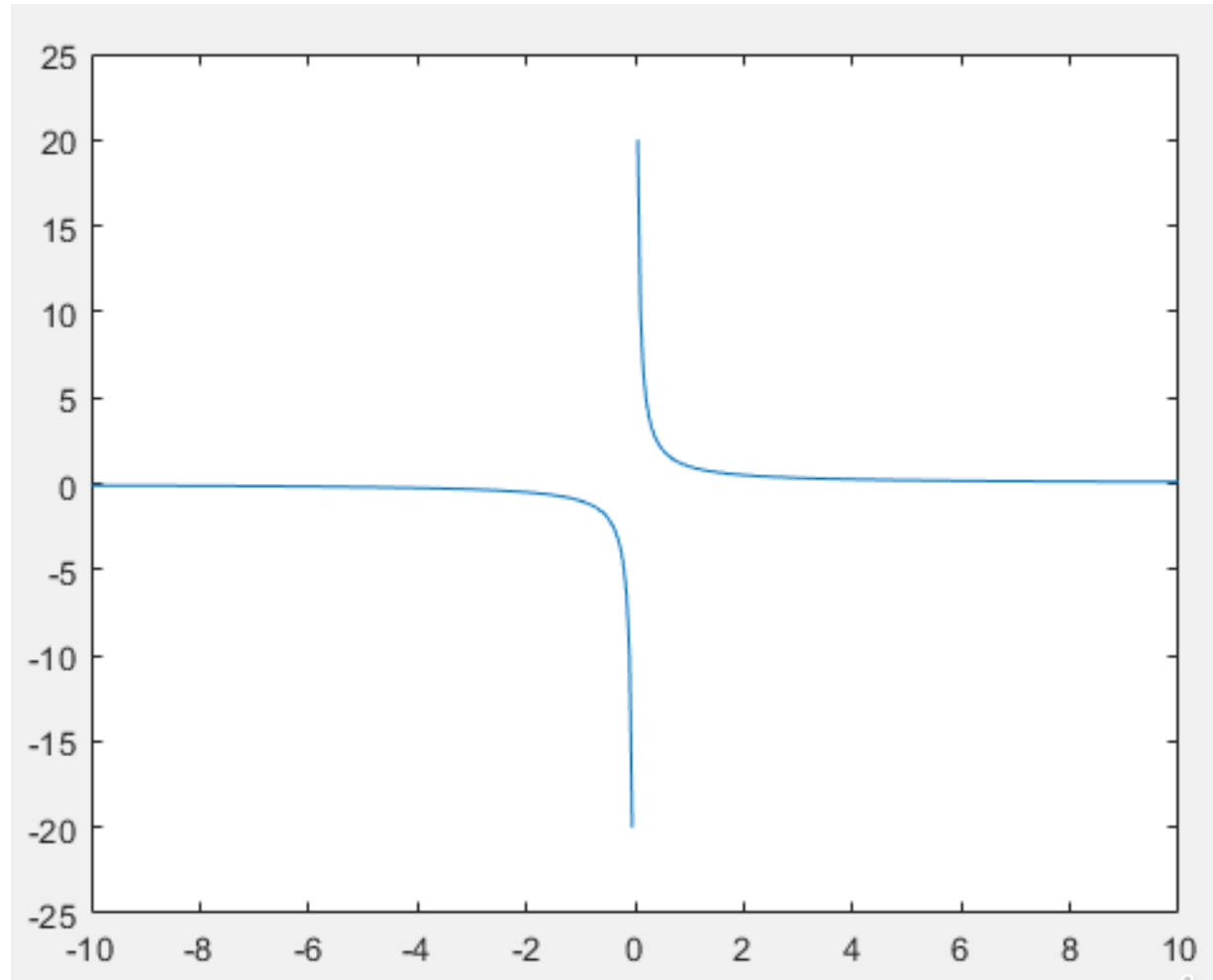


# plot

```
x = [-10:0.05:10]
```

```
y = 1./x;
```

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



# plot

`plot(X,Y)` plots vector `Y` versus vector `X`. If `X` or `Y` is a matrix, then the vector is plotted versus the rows or columns of the matrix, whichever line up.

If `X` is a scalar and `Y` is a vector, disconnected line objects are created and plotted as discrete points vertically at `X`.

See also `plottools`, `semilogx`, `semilogy`, `loglog`, `plotyy`, `plot3`, `grid`, `title`, `xlabel`, `ylabel`, `axis`, `axes`, `hold`, `legend`, `subplot`, `scatter`.

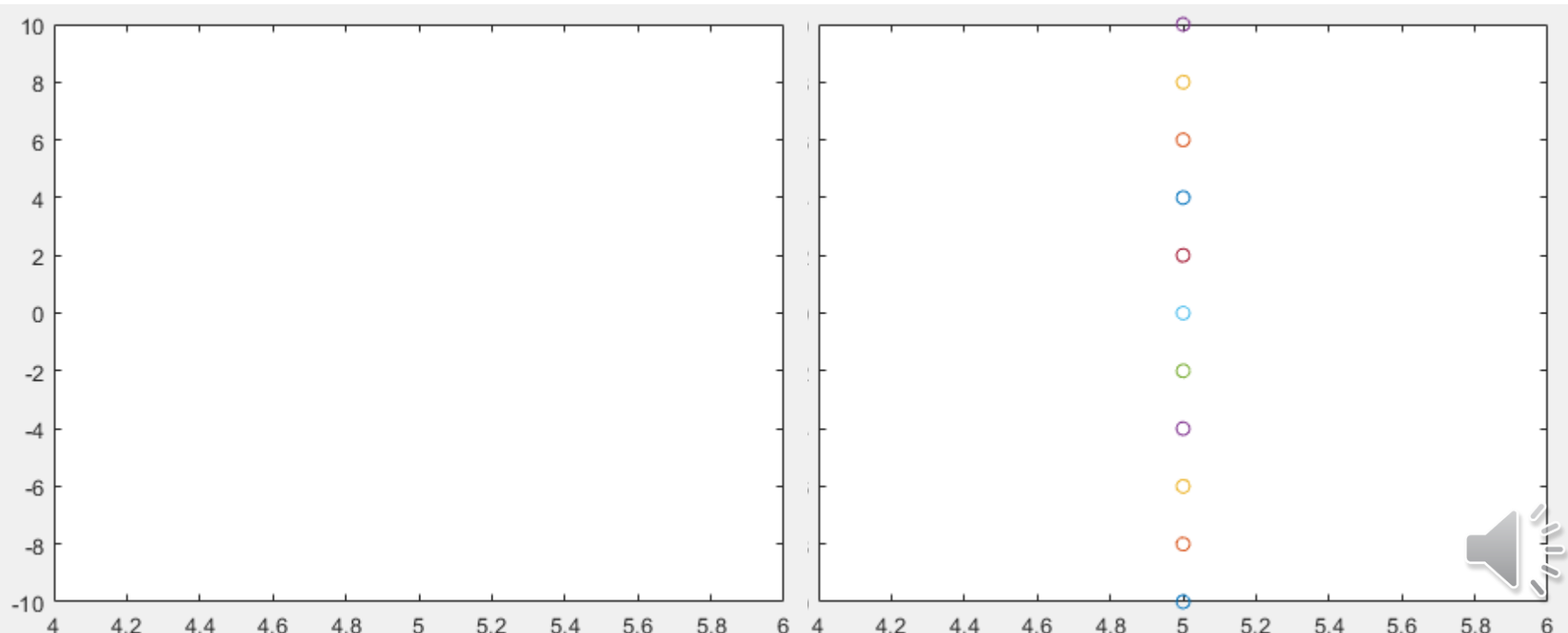


# plot

`plot(X,Y)`. If  $X$  is a scalar and  $Y$  is a vector, disconnected line objects are created and plotted as discrete points vertically at  $X$ .

```
clf
y = [-10:2:10];
x = 5;
plot(x,y)
```

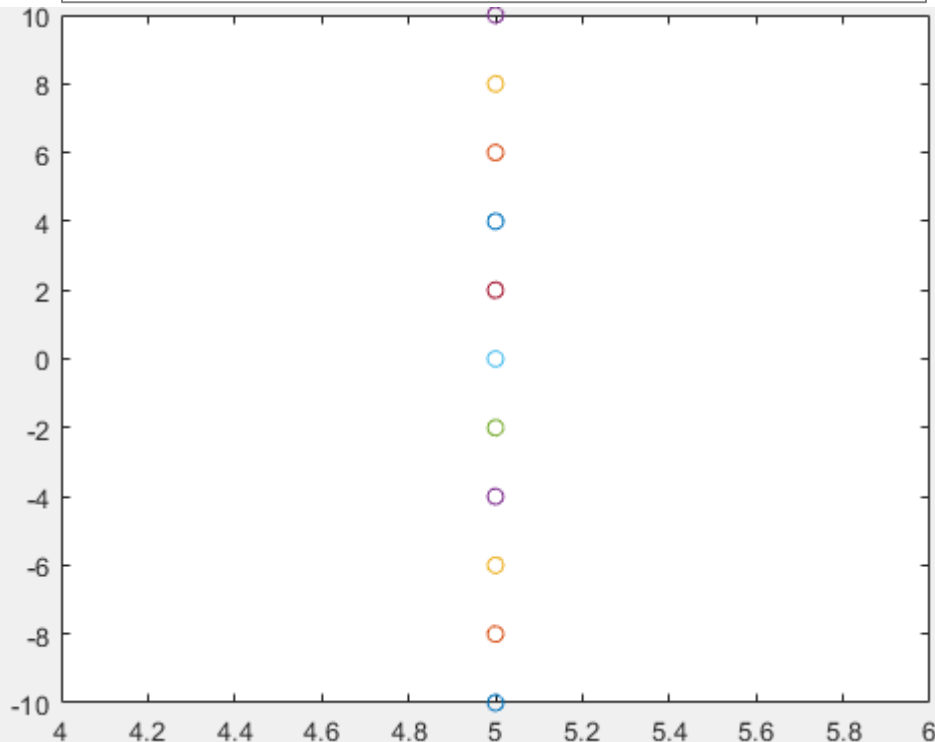
```
clf
y = [-10:2:10];
x = 5;
plot(x,y, 'o')
```



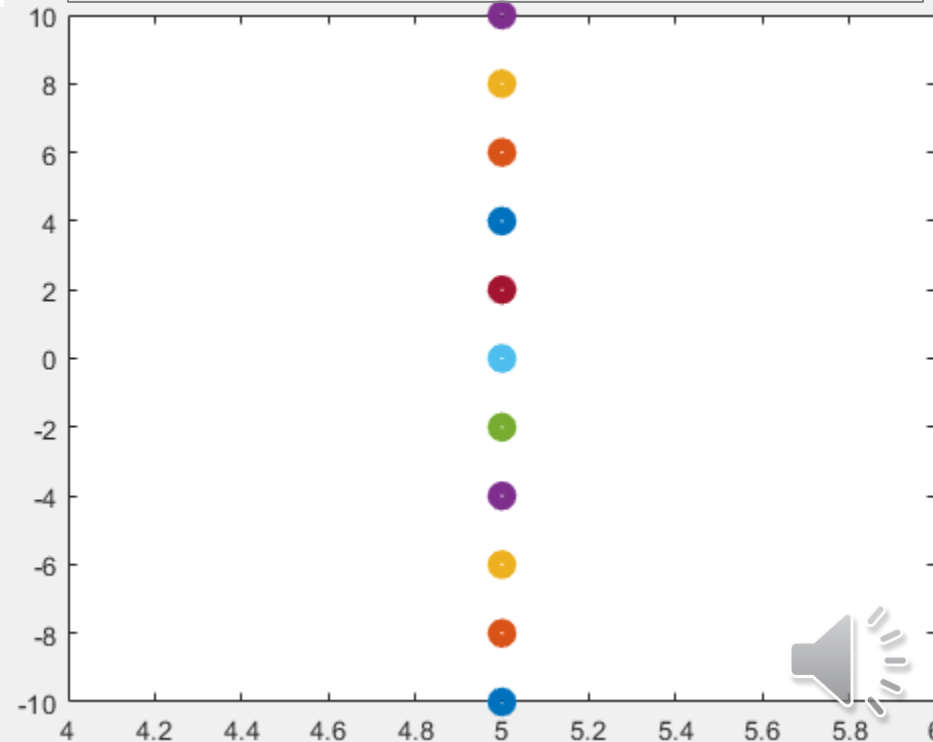
# plot

`plot(X,Y)`. If `X` is a scalar and `Y` is a vector, disconnected line objects are created and plotted as discrete points vertically at `X`.

```
clf
y = [-10:2:10];
x = 5;
plot(x,y, 'o')
```



```
clf
y = [-10:2:10];
x = 5;
plot(x,y, 'o', ...
      'linewidth', 5)
```



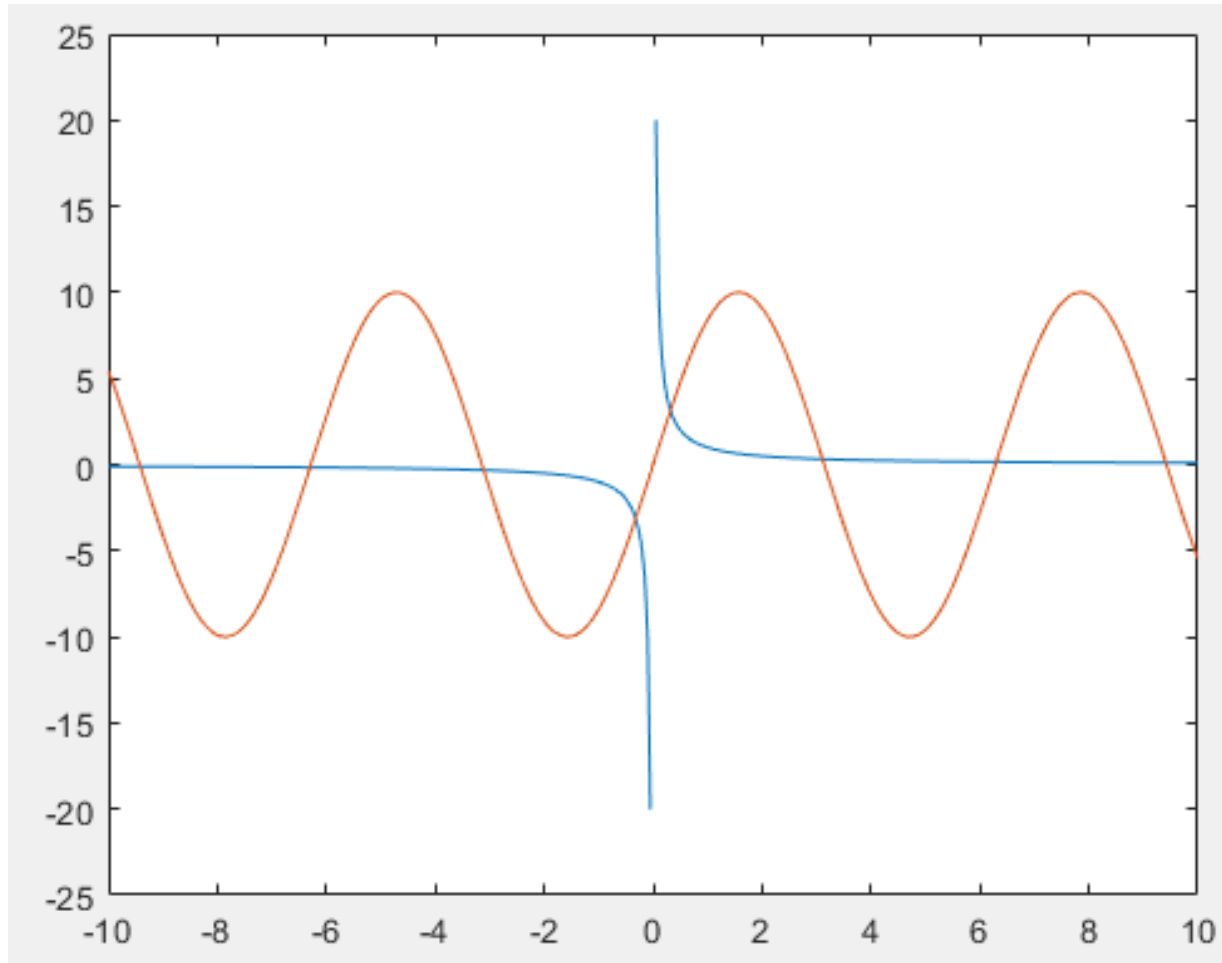
# plot

clf

```
x = [-10:0.05:10];
```

```
y = [1./x; 10.*sin(x)];
```

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



# plot

```
x = [-10:0.05:10];
```

```
y = 1./x;
```

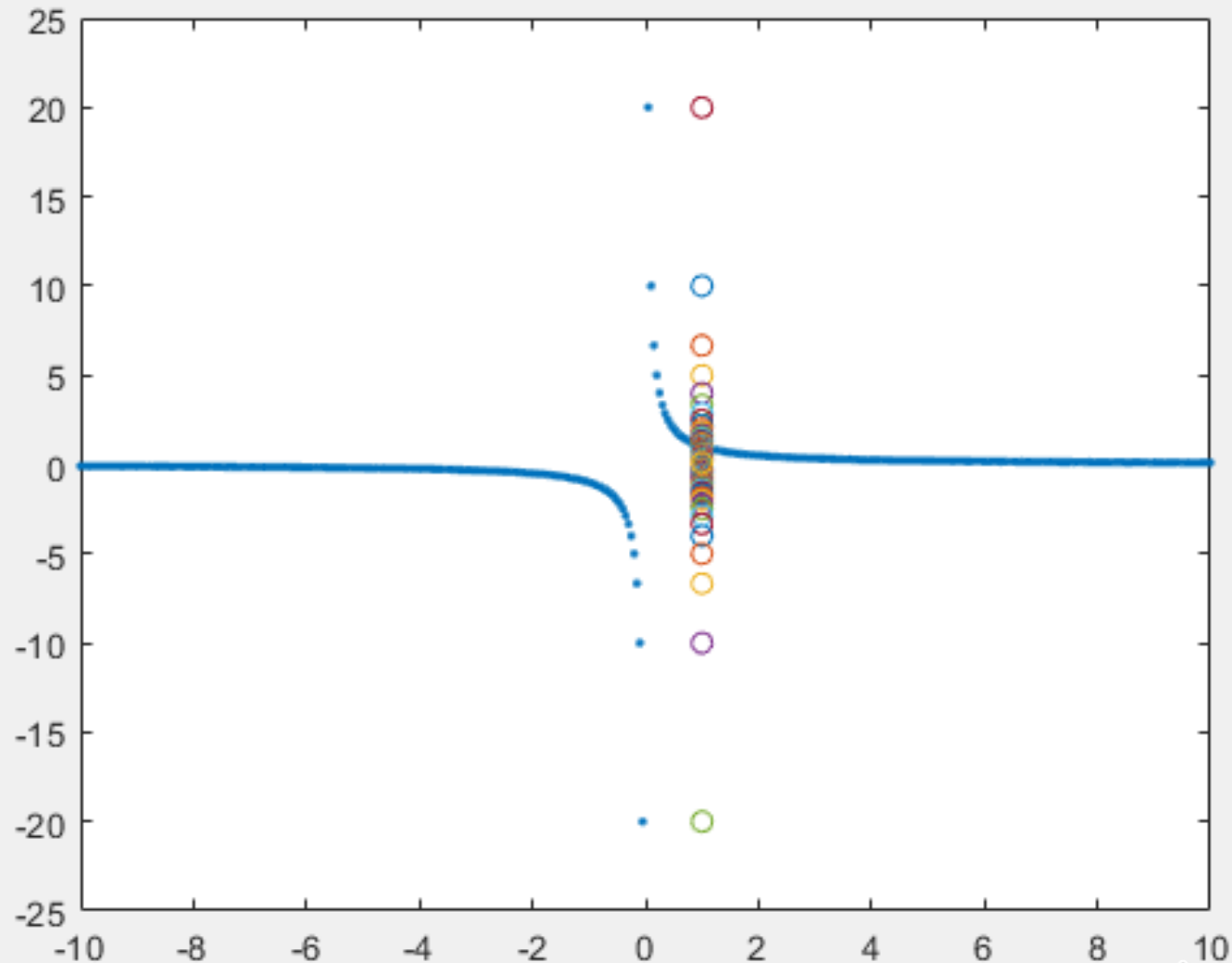
```
x0 = 1;
```

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

```
hold on
```

```
plot(x0,y, 'o');
```

If X is a scalar and Y is a vector, disconnected line objects are created and plotted as discrete points vertically at X.





# plot

```
x = [-10:0.05:10];
```

```
y = 1./x;
```

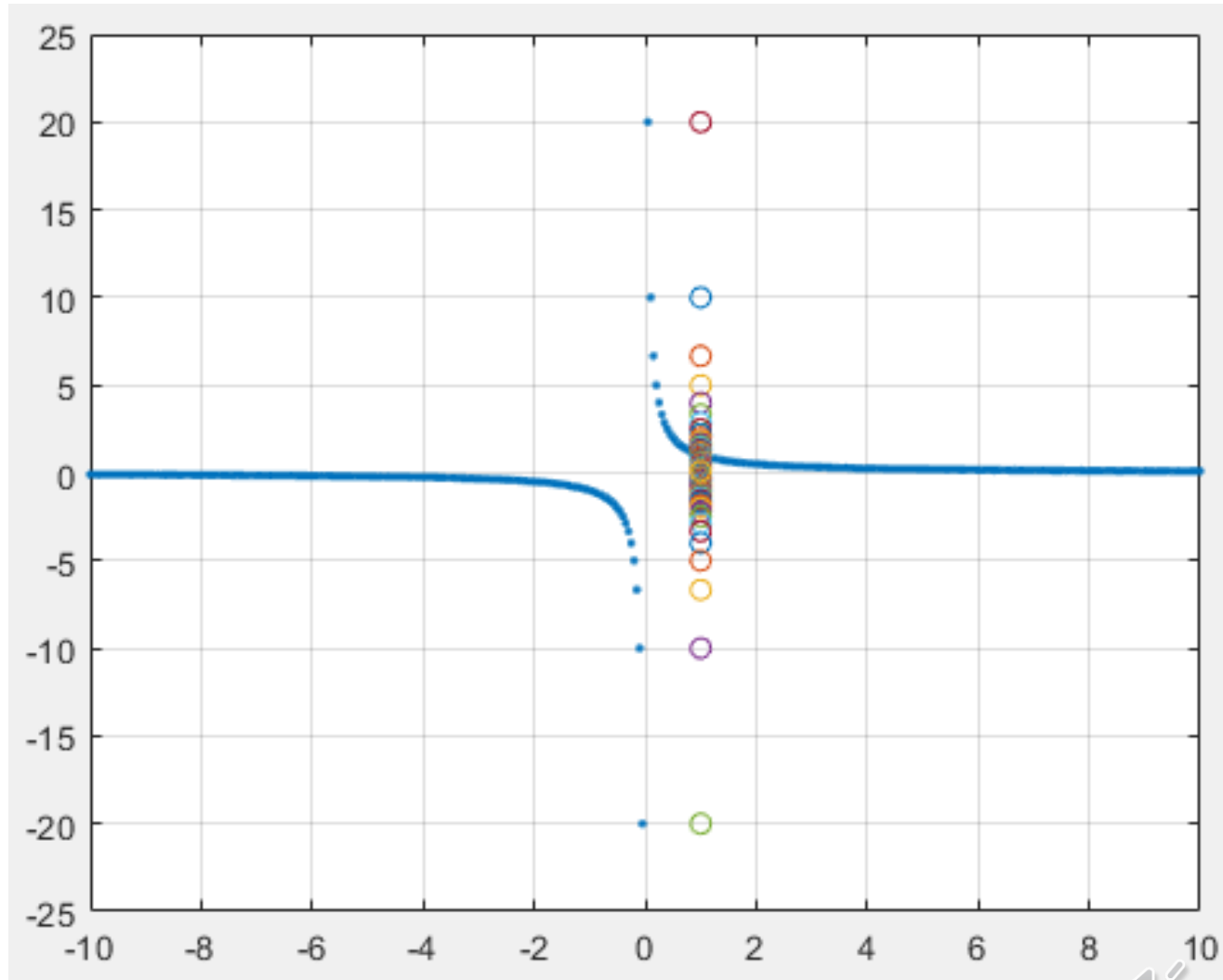
```
x0 = 1;
```

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

```
hold on
```

```
plot(x0,y, 'o');
```

```
grid on;
```



# grid

grid Grid lines.

grid ON adds major grid lines to the current axes.

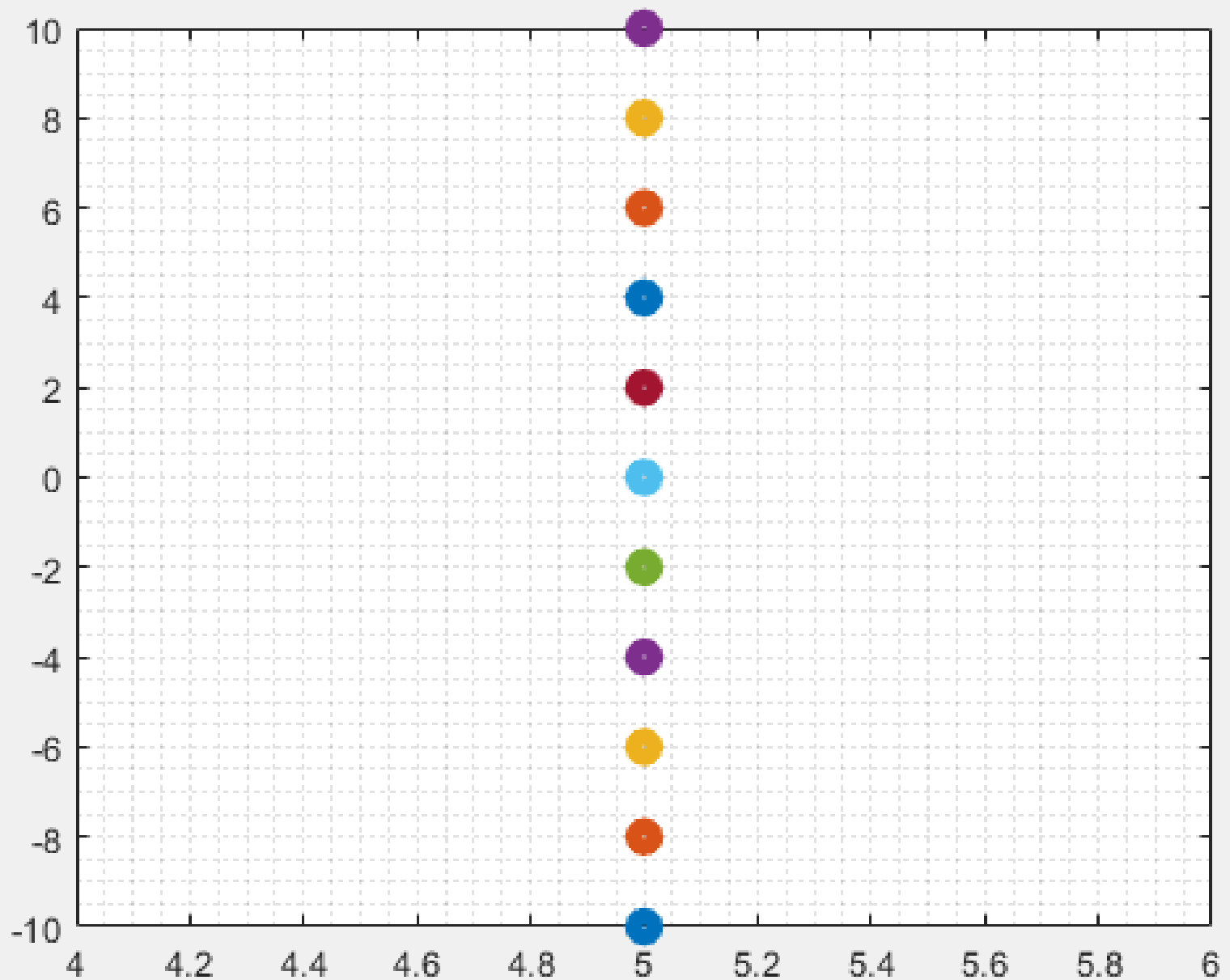
grid OFF removes major and minor grid lines from the current axes.

grid MINOR toggles the minor grid lines of the current axes.

grid, by itself, toggles the major grid lines of the current axes.

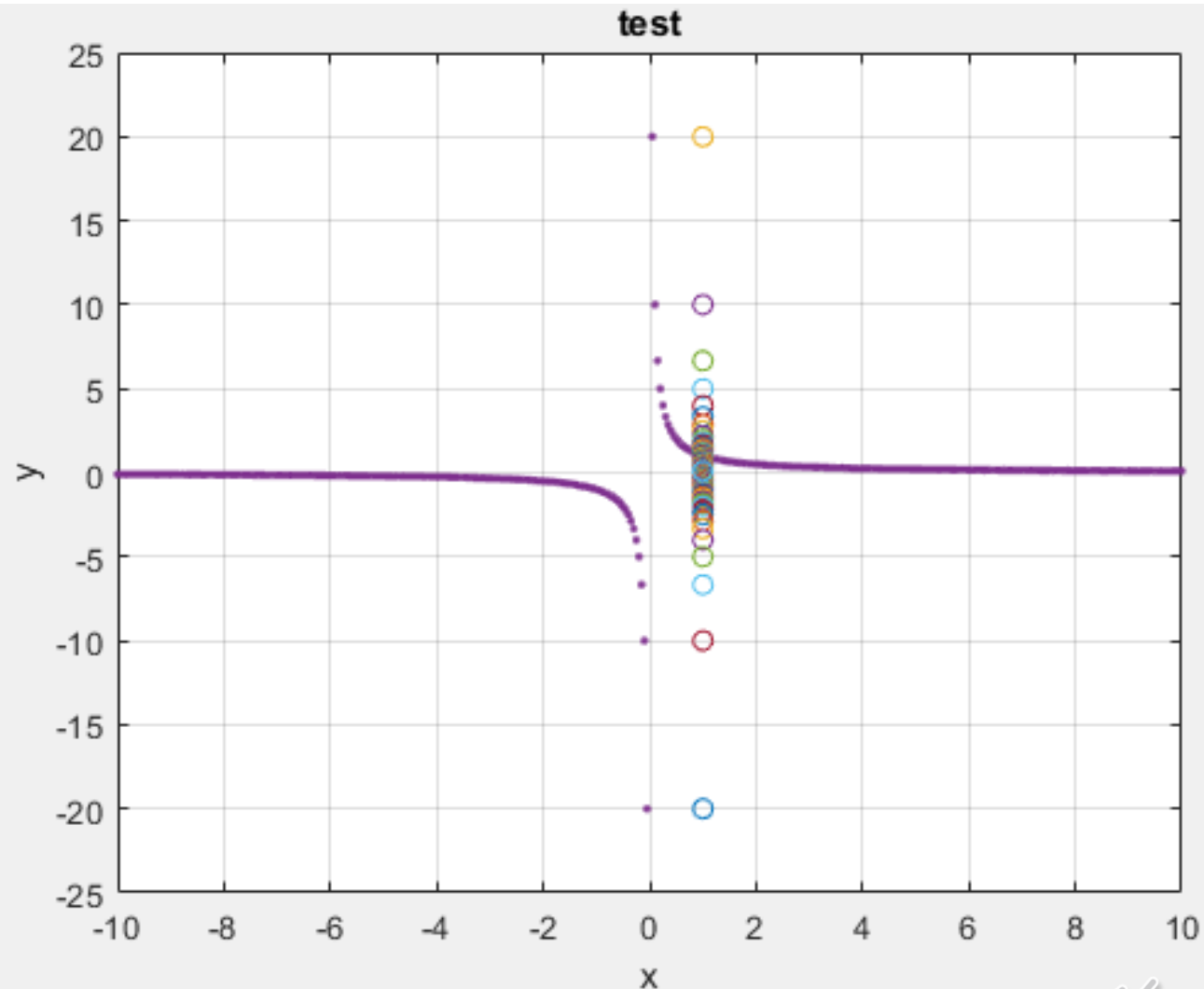


# grid minor



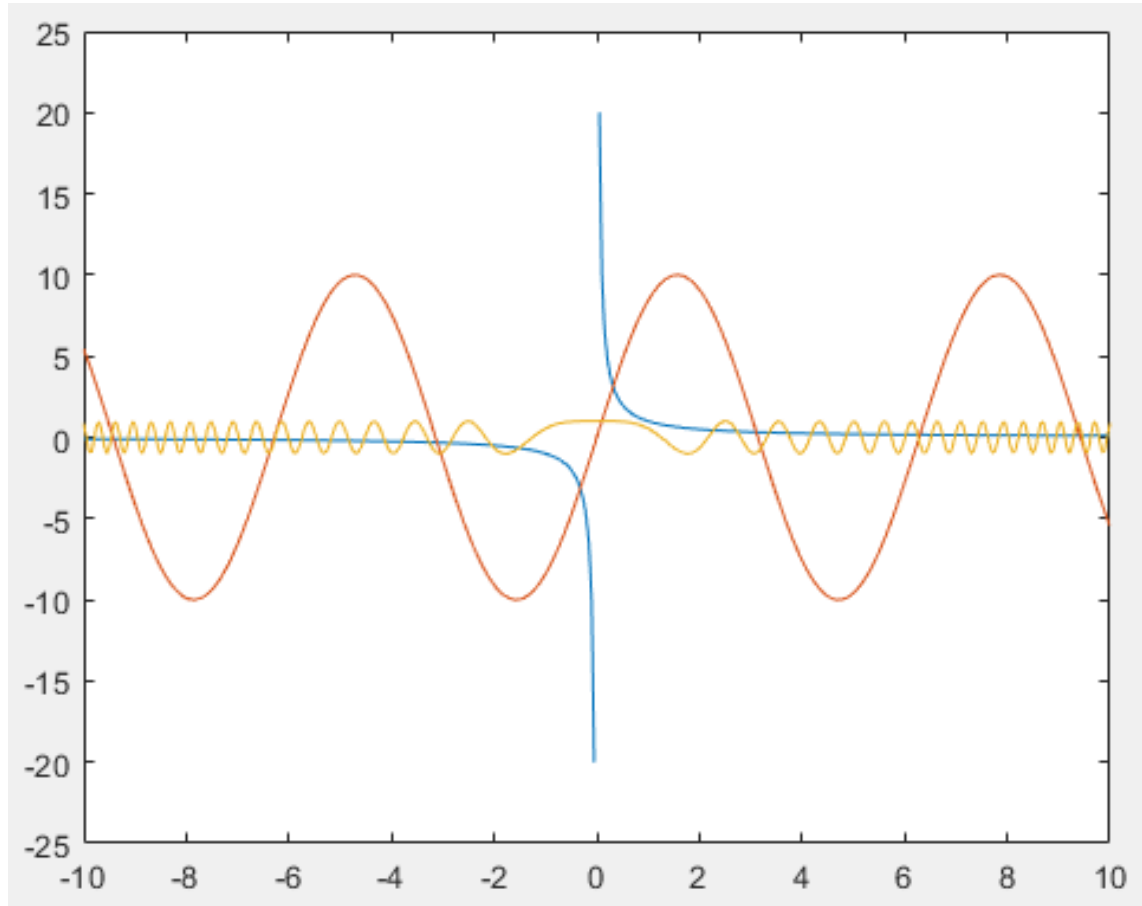
# plot

```
x = [-10:0.05:10];  
y = 1./x;  
x0 = 1;  
plot(x,y,'.');  
hold on  
plot(x0,y, 'o');  
title('test');  
xlabel('x');  
ylabel('y');
```



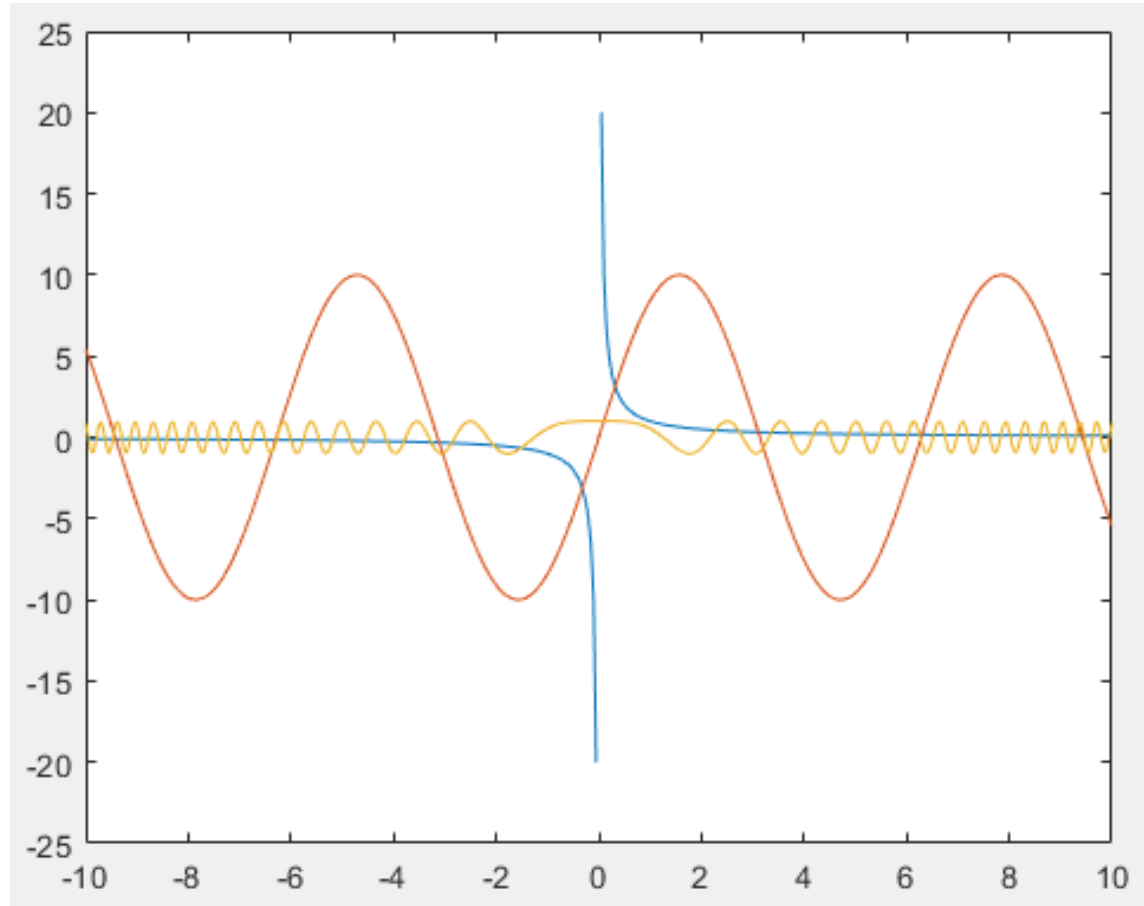
# Multiple curves

```
clf
x = [-10:0.05:10];
y = [1./x;...
     10.*sin(x);...
     cos(x.^2)];
plot(x,y);
```



# Multiple curves

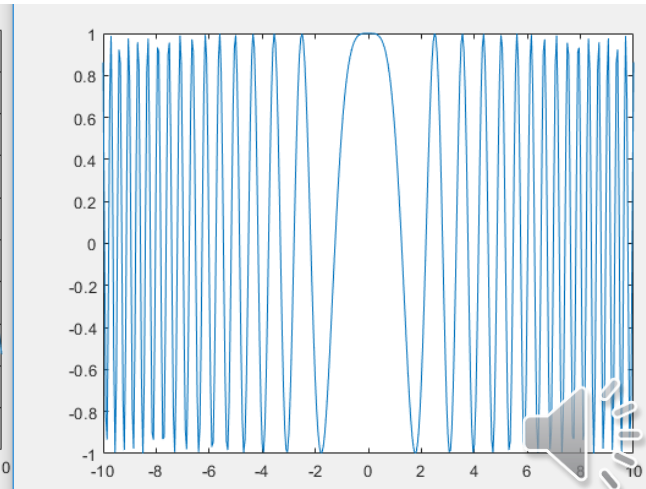
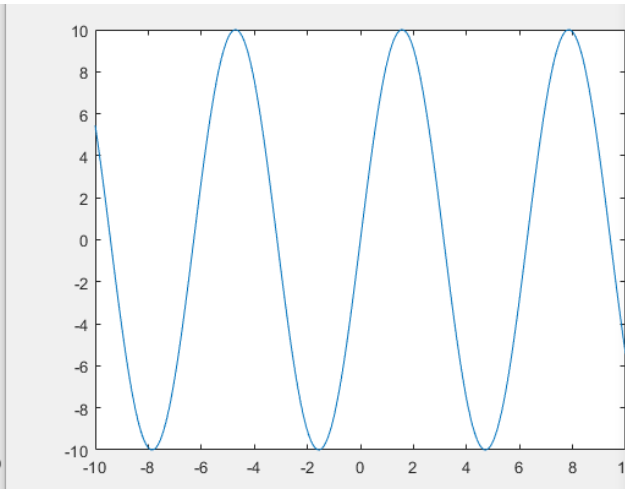
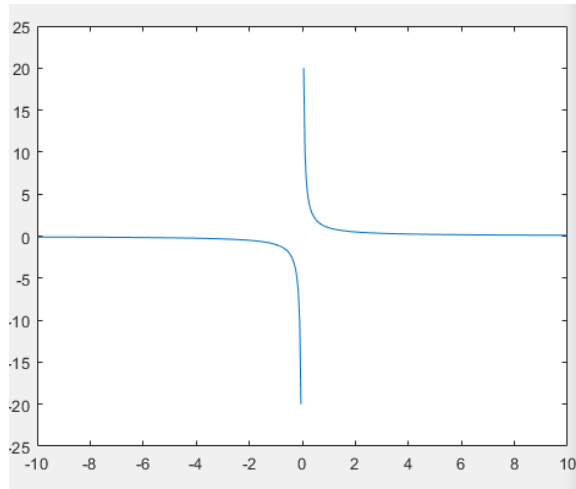
```
clf
x = [-10:0.05:10];
y1 = 1./x;
y2 = 10.*sin(x);
y3 = cos(x.^2);
plot(x,y1);
hold on
plot(x,y2);
hold on
plot(x,y3);
hold off
figure %new figure
```



# Multiple figures

```
clf  
x = [-10:0.05:10];  
y1 = 1./x;  
y2 = 10.*sin(x);  
y3 = cos(x.^2);
```

```
figure(1), plot(x,y1);  
figure(2), plot(x,y2);  
figure(3), plot(x,y3);
```



# Line Styles

Line Style	Description
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dot line





# Markers

Marker	Description
o	Circle
+	Plus sign
*	Asterisk
.	Point
x	Cross
s	Square
d	Diamond
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
p	Pentagram
h	Hexagram



# Colors

Color	Description
y	yellow
m	magenta
c	cyan
r	red
g	green
b	blue
w	white
k	black



# Functions

axis

text

legend

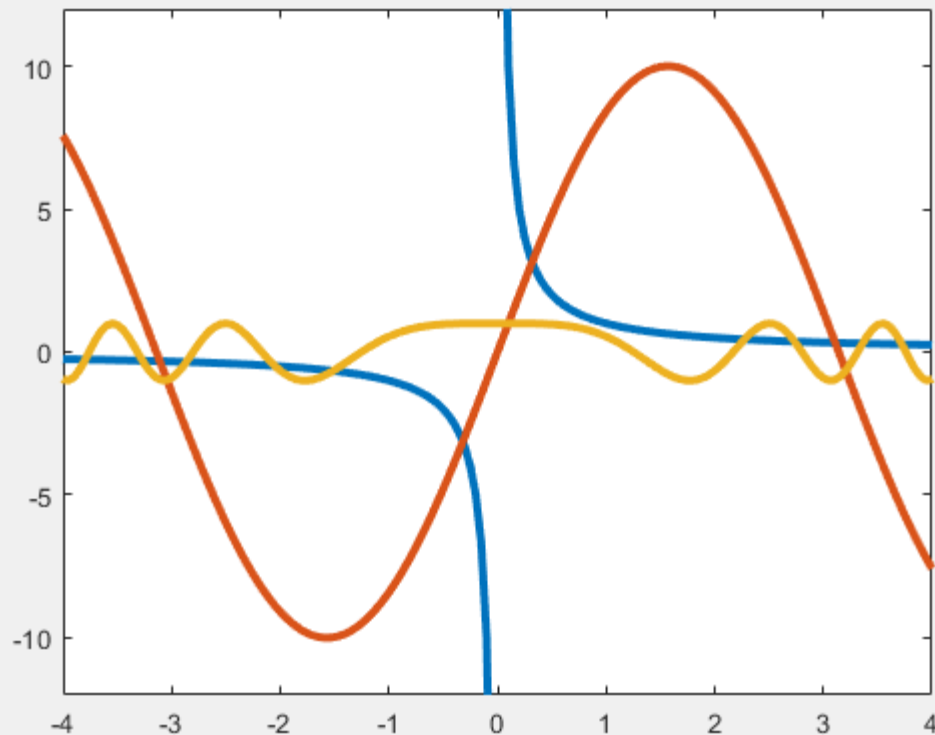
set



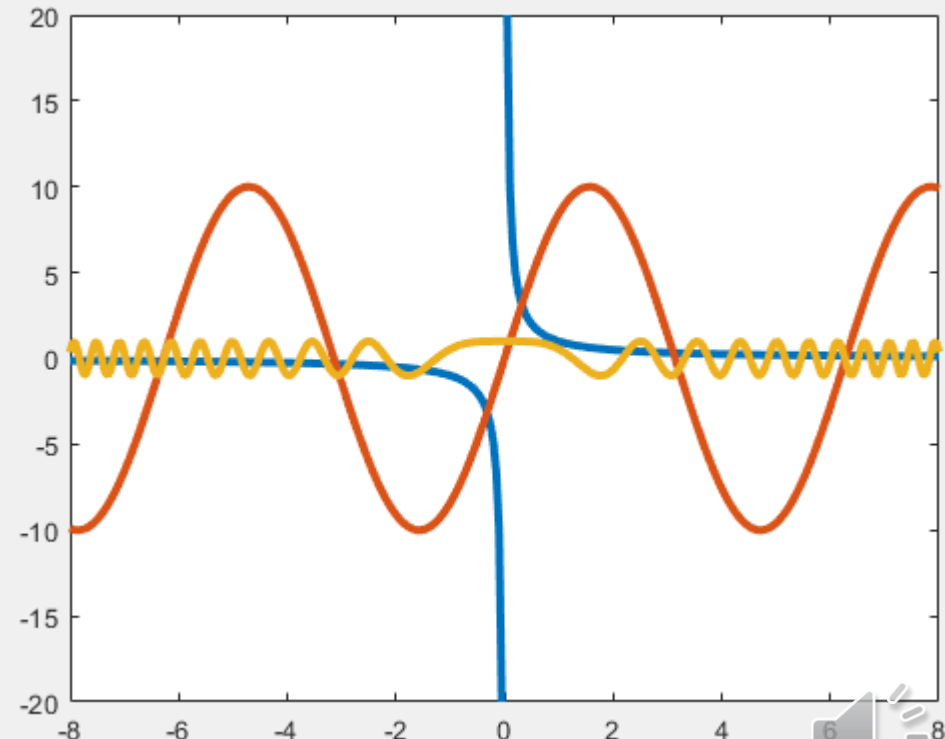
# axis

`axis([XMIN XMAX YMIN YMAX])` sets scaling for the x- and y-axes on the current plot.

`axis( [-4 4 -12 12] );`



`axis( [-8 8 -20 20] );`

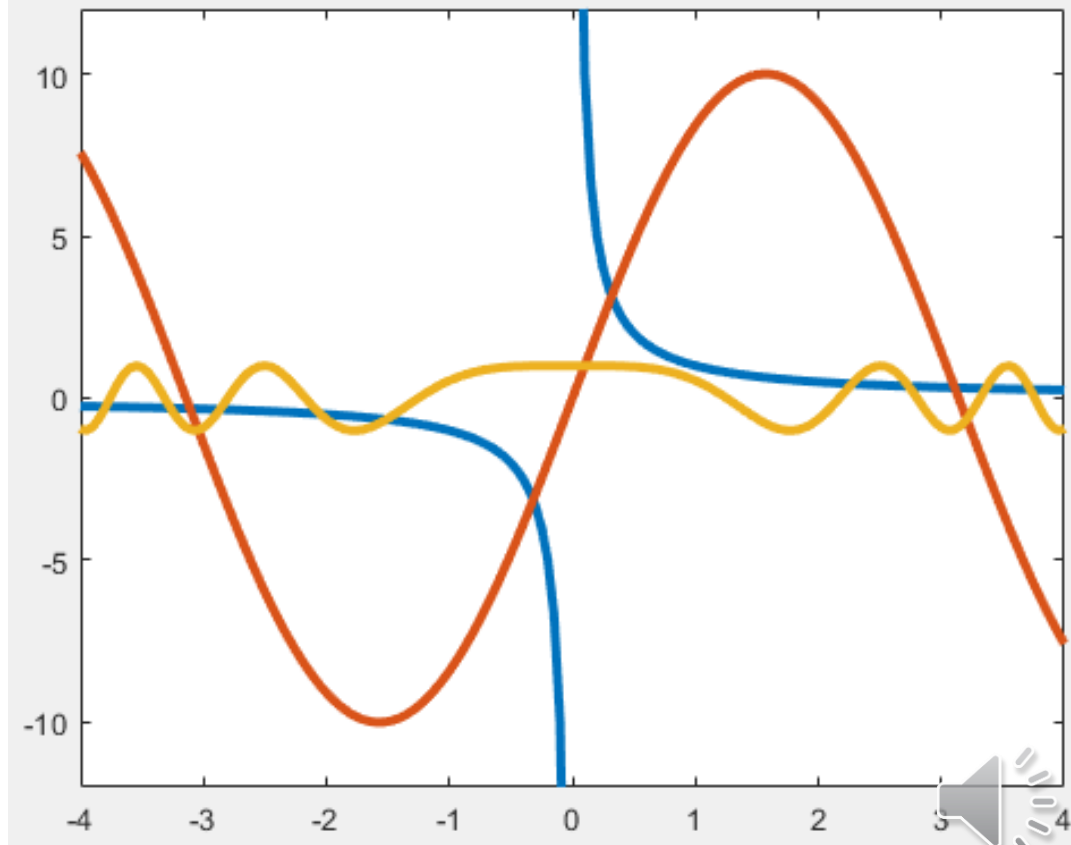


# axis

```
clf
x = [-10:0.05:10];
y1 = 1./x;
y2 = 10.*sin(x);
y3 = cos(x.^2);
plot(x,y1,'LineWidth',3);
hold on
plot(x,y2,'LineWidth',3);
hold on
plot(x,y3,'LineWidth',3);
hold off
axis( [-4 4 -12 12] );
```

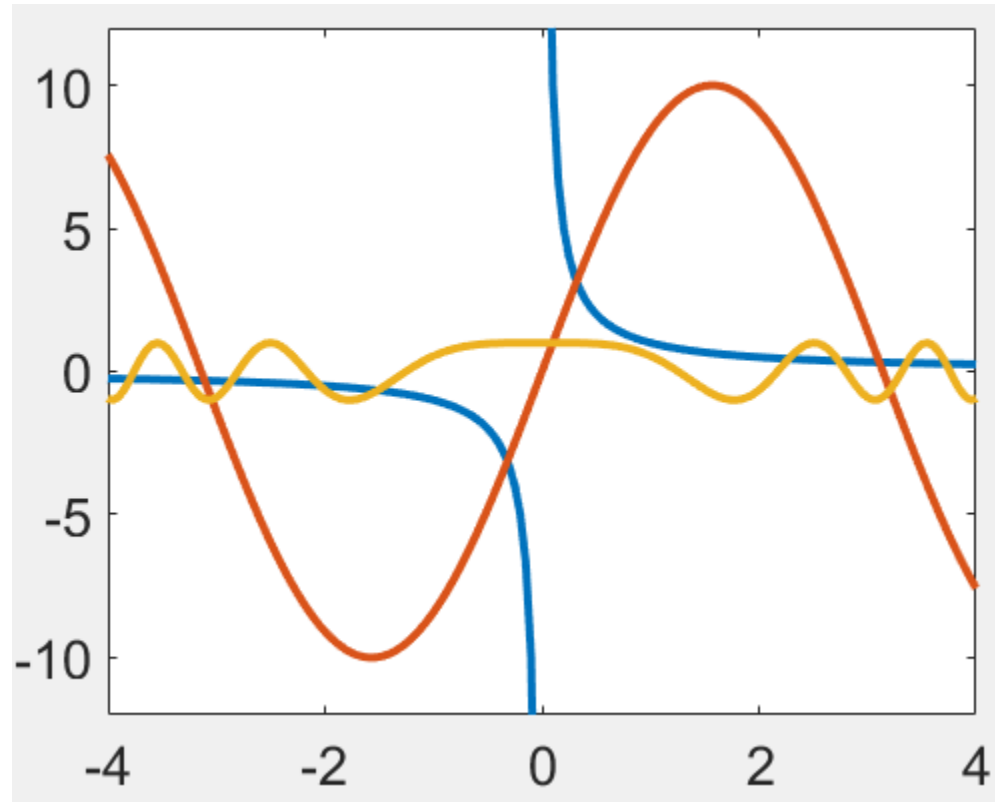
Main process:

1. Set the range for the independent variable.
2. Define the functions
3. Plot the curves for the functions
4. Set range for the plot



# set

```
clf
x = [-10:0.05:10];
y1 = 1./x;
y2 = 10.*sin(x);
y3 = cos(x.^2);
plot(x,y1,'LineWidth',3);
hold on
plot(x,y2,'LineWidth',3);
hold on
plot(x,y3,'LineWidth',3);
hold off
axis( [-4 4 -12 12] );
set(gca,'FontSize',20);
```

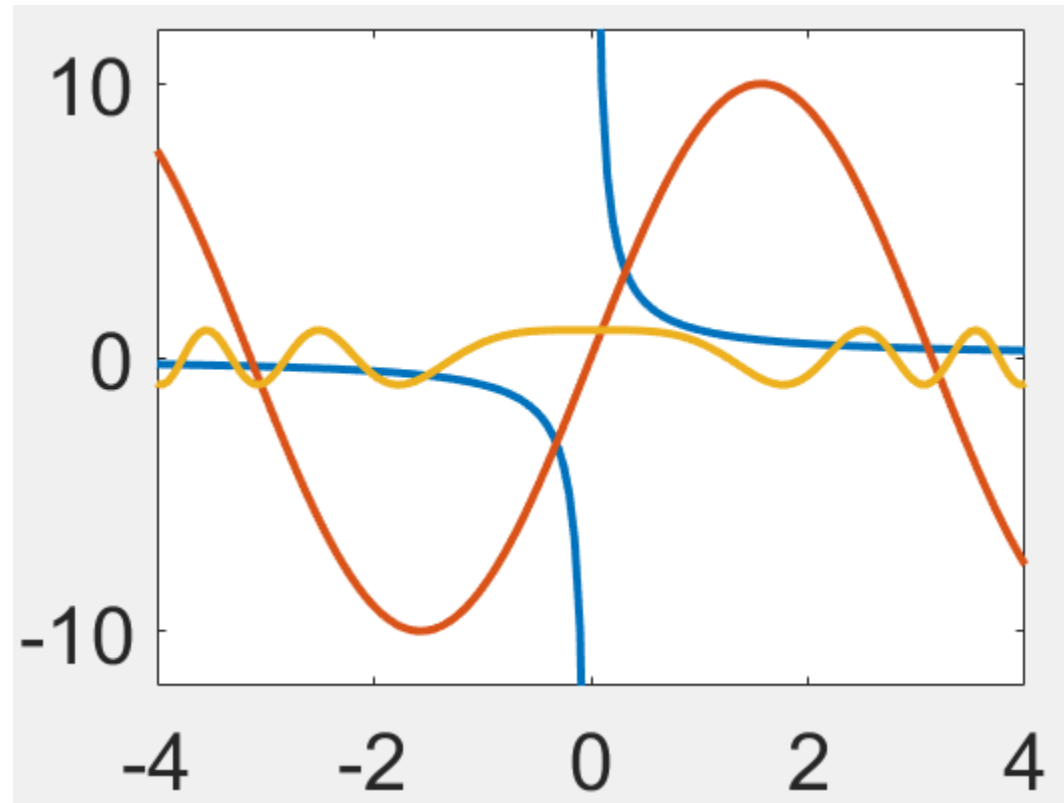


gca: the "handle" of the  
currently active axes object



# set

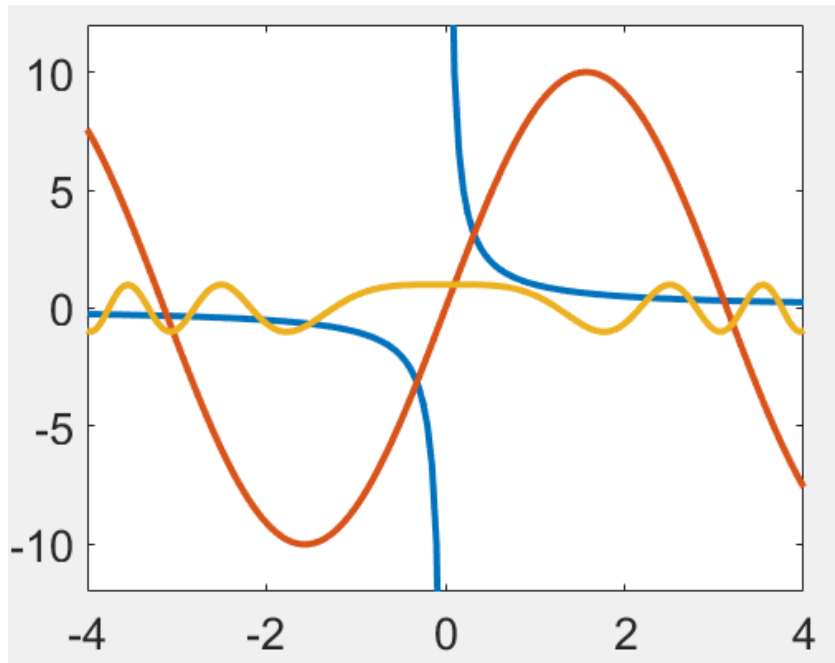
```
clf
x = [-10:0.05:10];
y1 = 1./x;
y2 = 10.*sin(x);
y3 = cos(x.^2);
plot(x,y1,'LineWidth',3);
hold on
plot(x,y2,'LineWidth',3);
hold on
plot(x,y3,'LineWidth',3);
hold off
axis( [-4 4 -12 12] );
set(gca,'FontSize',30);
```



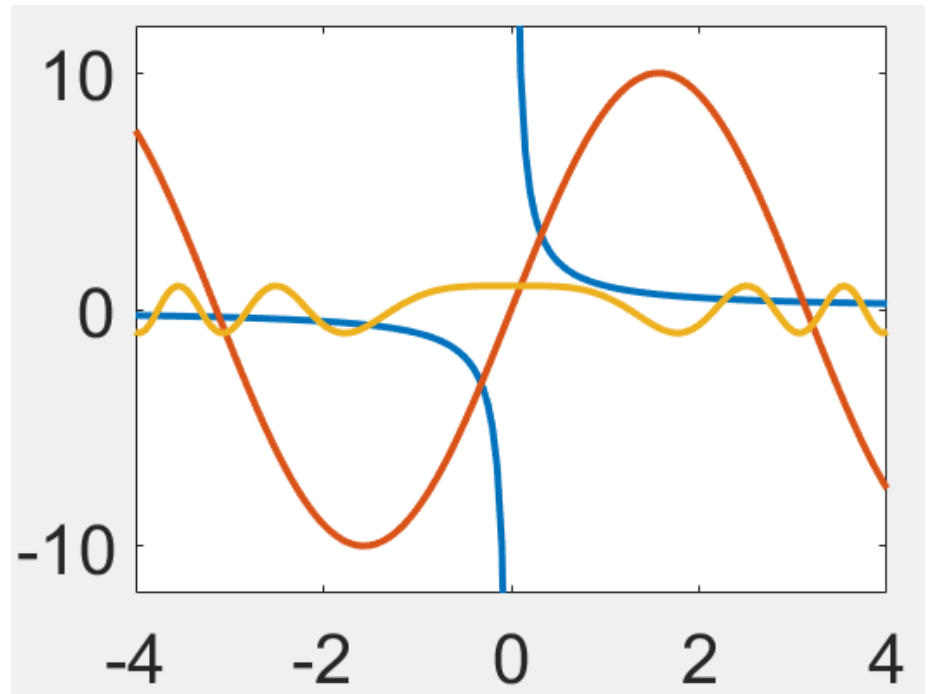
gca: the "handle" of the  
currently active axes object



# set



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



**`set(gca,'FontSize',30);`**





# text

`text(x,y,str)` adds a text description to one or more data points in the current axes using the text specified by `str`. To add text to one point, specify `x` and `y` as scalars in data units. To add text to multiple points, specify `x` and `y` as vectors with equal length.



# text

Plot a curve  $y = \sqrt{x}$ , for  $x$  in  $[-10:0.05:10]$ .

Mark a curve point at  $x = 5$ , i.e., mark  $(5, \sqrt{5})$ ; and show the point coordinates with 2 significant digits for the  $y$ -coordinate. Beautify the figure, i.e., make it readable, and clear.



# text

Plot a curve  $y = \sqrt{x}$ , for  $x$  in  $[-10:0.05:10]$ .

Mark a curve point at  $x = 5$ , i.e., mark  $(5, \sqrt{5})$ ; and show the point coordinates with 2 significant digits for the y-coordinate. Beautify the figure, i.e., make it readable, and clear.

## The Approach?

1. Define the range of  $x$
2. Compute values for the function
3. Use a method to compute a string for the coordinates
4. Display the string at  $(5, \sqrt{5})$
5. Set the fontsize properly



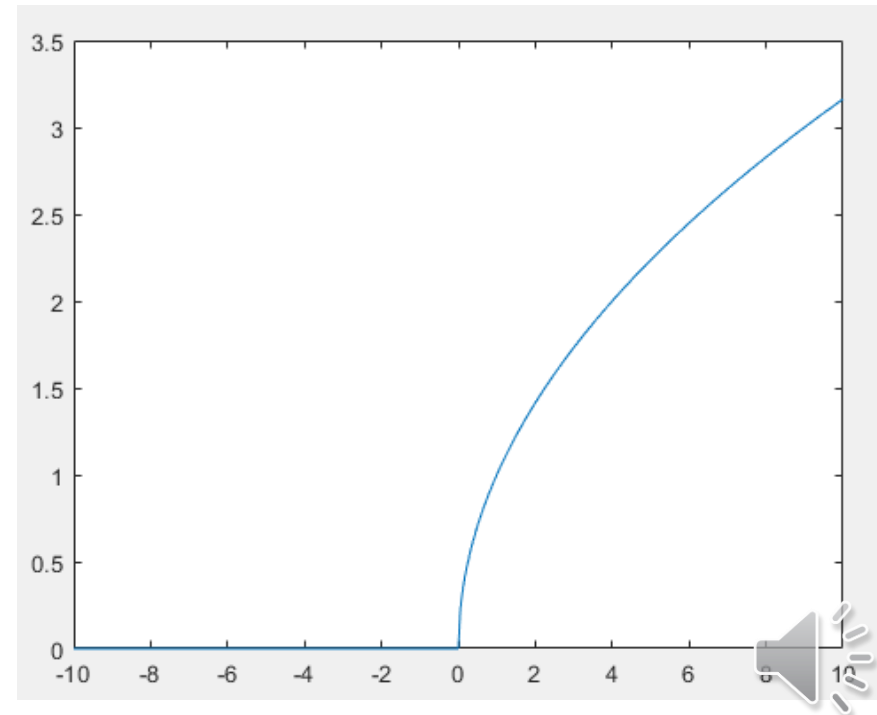
# text

Plot a curve  $y = \sqrt{x}$ , for  $x$  in  $[-10:0.05:10]$ .

Mark a curve point at  $x = 5$ , i.e., mark  $(5, \sqrt{5})$ ; and show the point coordinates with 2 significant digits for the  $y$ -coordinate. Beautify the figure, i.e., make it readable, and clear.

```
clf
x = [-10:0.05:10];
y = sqrt(x);
plot(x,y);

x0 = 5; y0 = sqrt(x0);
```



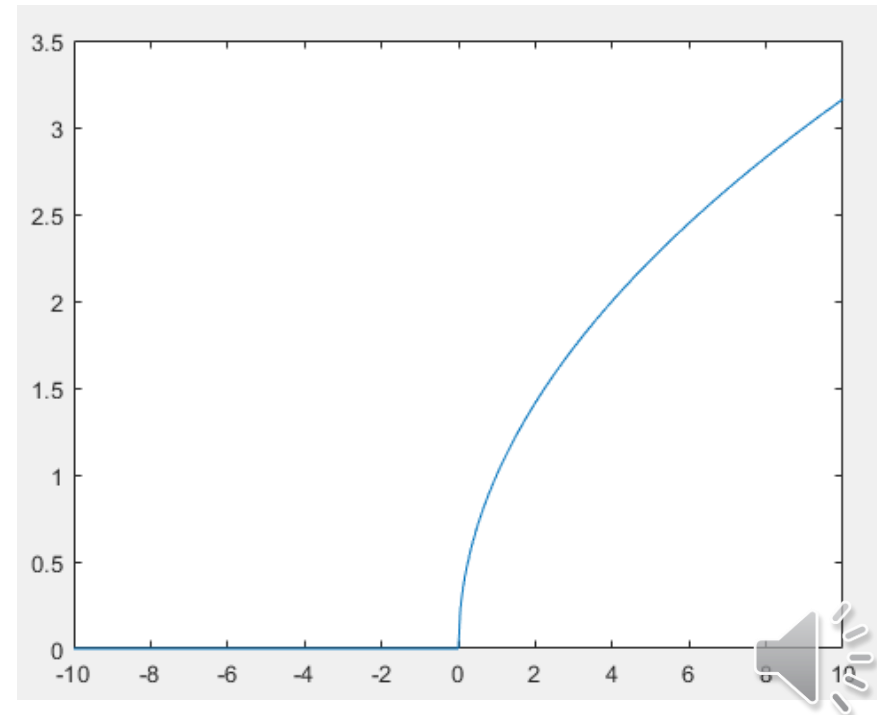
# text

Plot a curve  $y = \sqrt{x}$ , for  $x$  in  $[-10:0.05:10]$ .

Mark a curve point at  $x = 5$ , i.e., mark  $(5, \sqrt{5})$ ; and show the point coordinates with 2 significant digits for the  $y$ -coordinate. **Beautify the figure, i.e., make it readable, and clear.**

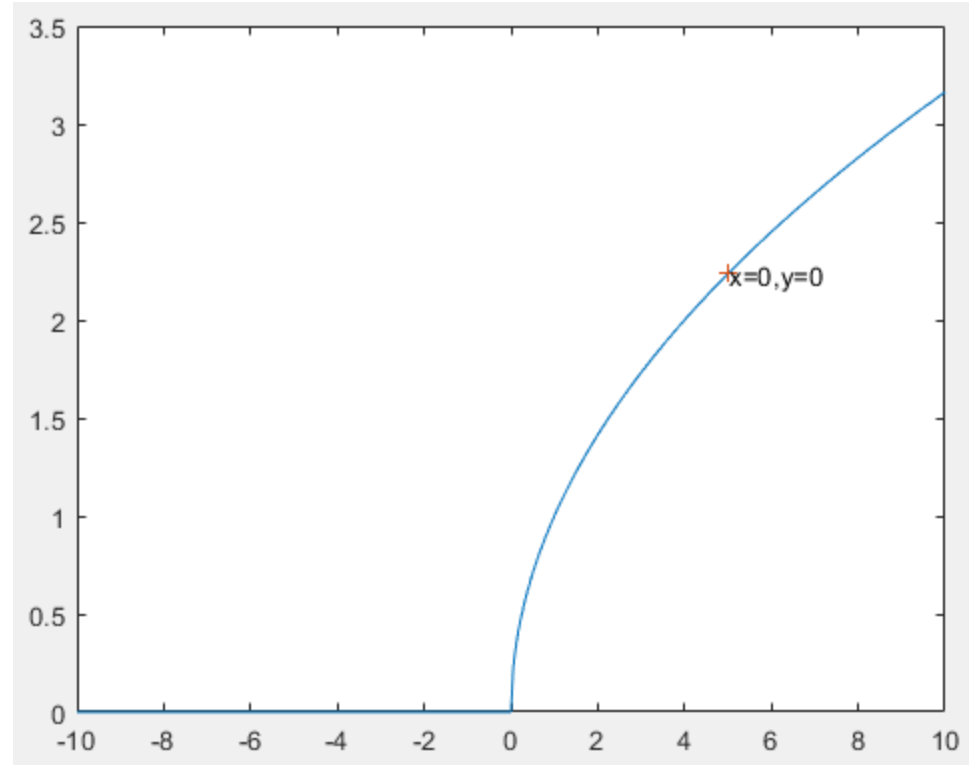
```
clf
x = [-10:0.05:10];
y = sqrt(x);
plot(x,y);

x0 = 5; y0 = sqrt(x0);
```



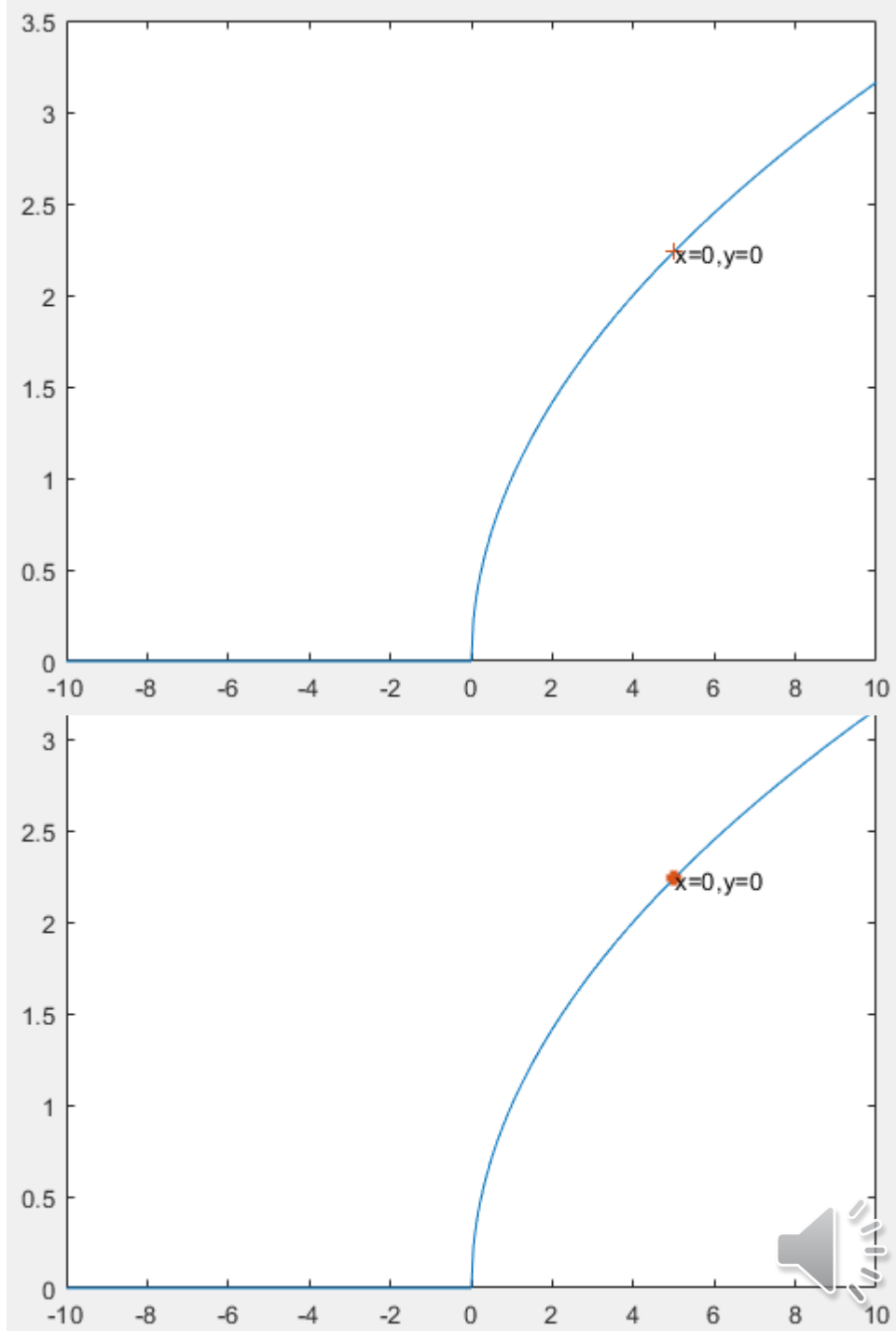
# text

```
clf
x = [-10:0.05:10];
y = sqrt(x); plot(x,y);
x0 = 5; y0 = sqrt(x0);
hold on,...
plot(x0,y0, '+');
text(x0, y0, ...
     'x=0,y=0');
```



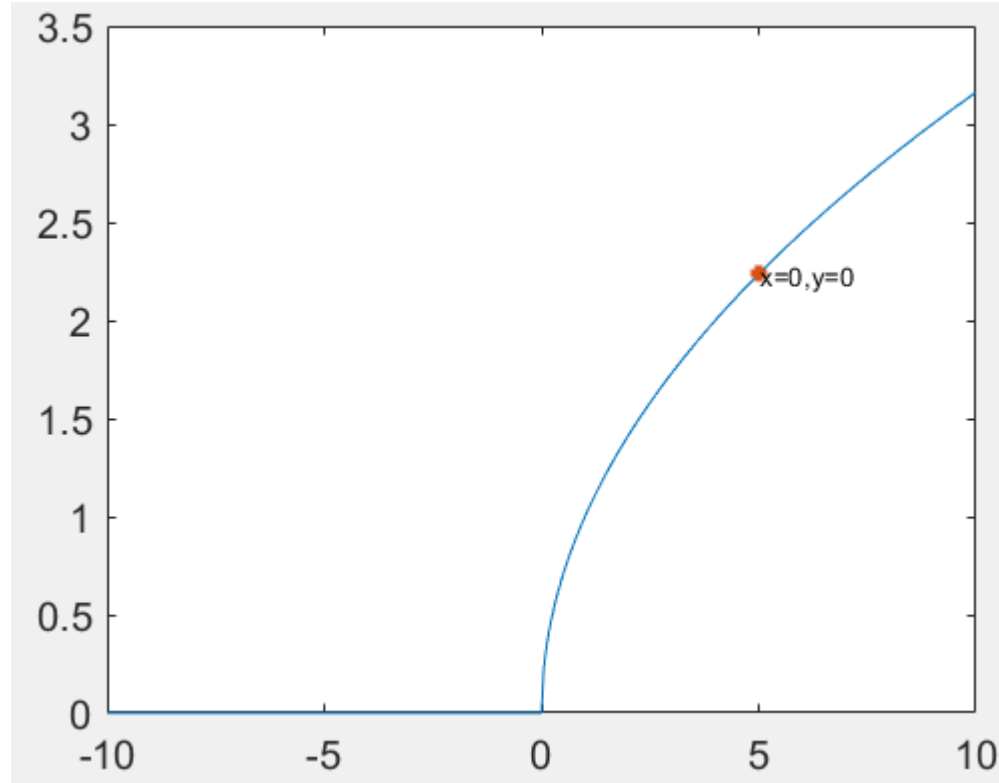
# text

```
clf
x = [-10:0.05:10];
y = sqrt(x); plot(x,y);
x0 = 5; y0 = sqrt(x0);
hold on, ...
plot(x0,y0,...
    '+', 'LineWidth', 3);
text(x0, y0, ...
    'x=0,y=0');
```



# text

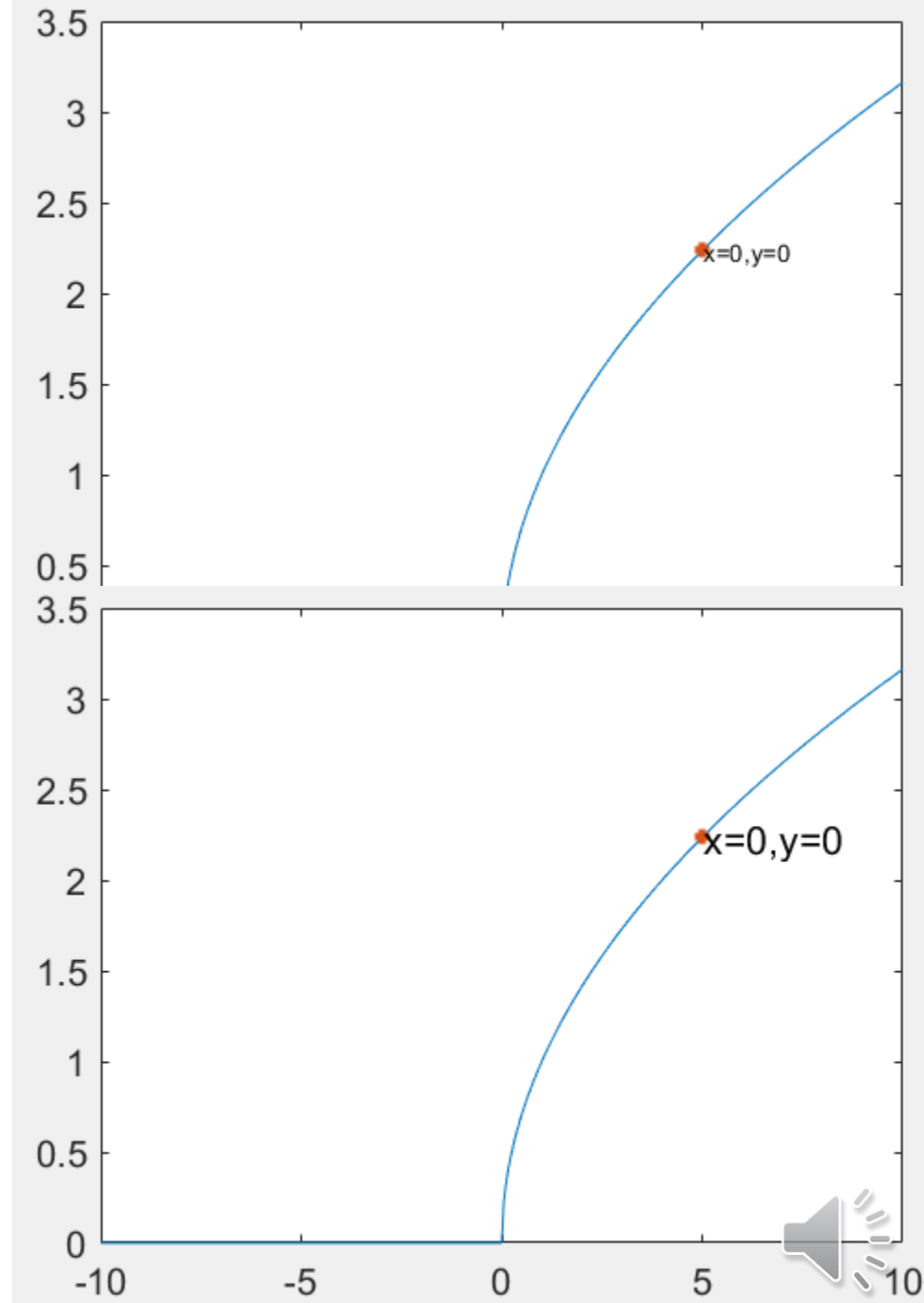
```
clf
x = [-10:0.05:10];
y = sqrt(x); plot(x,y);
x0 = 5; y0 = sqrt(x0);
hold on, ...
plot(x0,y0,...
    '+', 'LineWidth', 3);
text(x0, y0, ...
    'x=0,y=0');
set(gca,'FontSize',15);
```





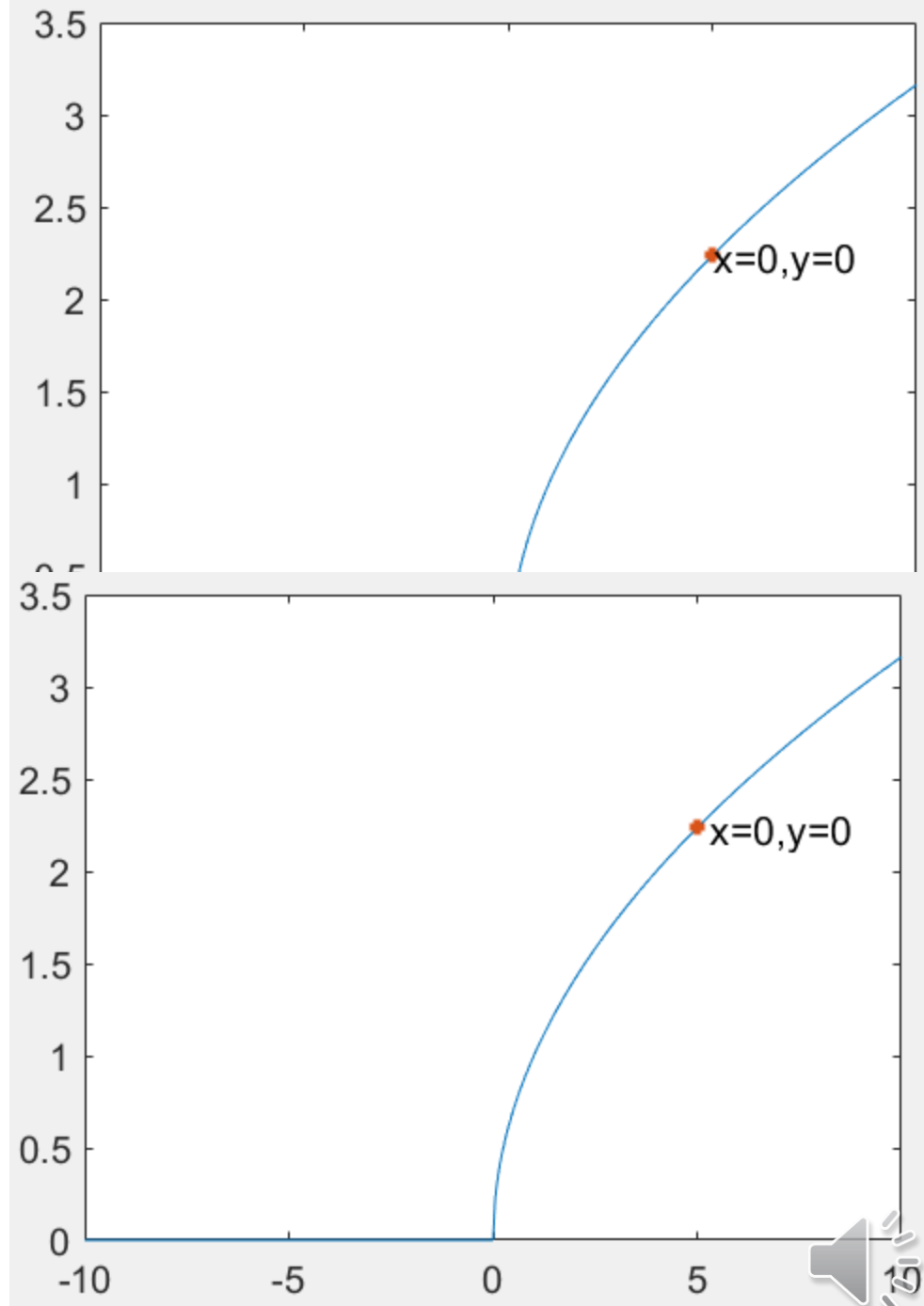
# text

```
clf
x = [-10:0.05:10];
y = sqrt(x); plot(x,y);
x0 = 5; y0 = sqrt(x0);
hold on, ...
plot(x0,y0,...
    '+', 'LineWidth', 3);
text(x0, y0, ...
    'x=0,y=0', ...
    'FontSize',16);
set(gca,'FontSize',15);
```



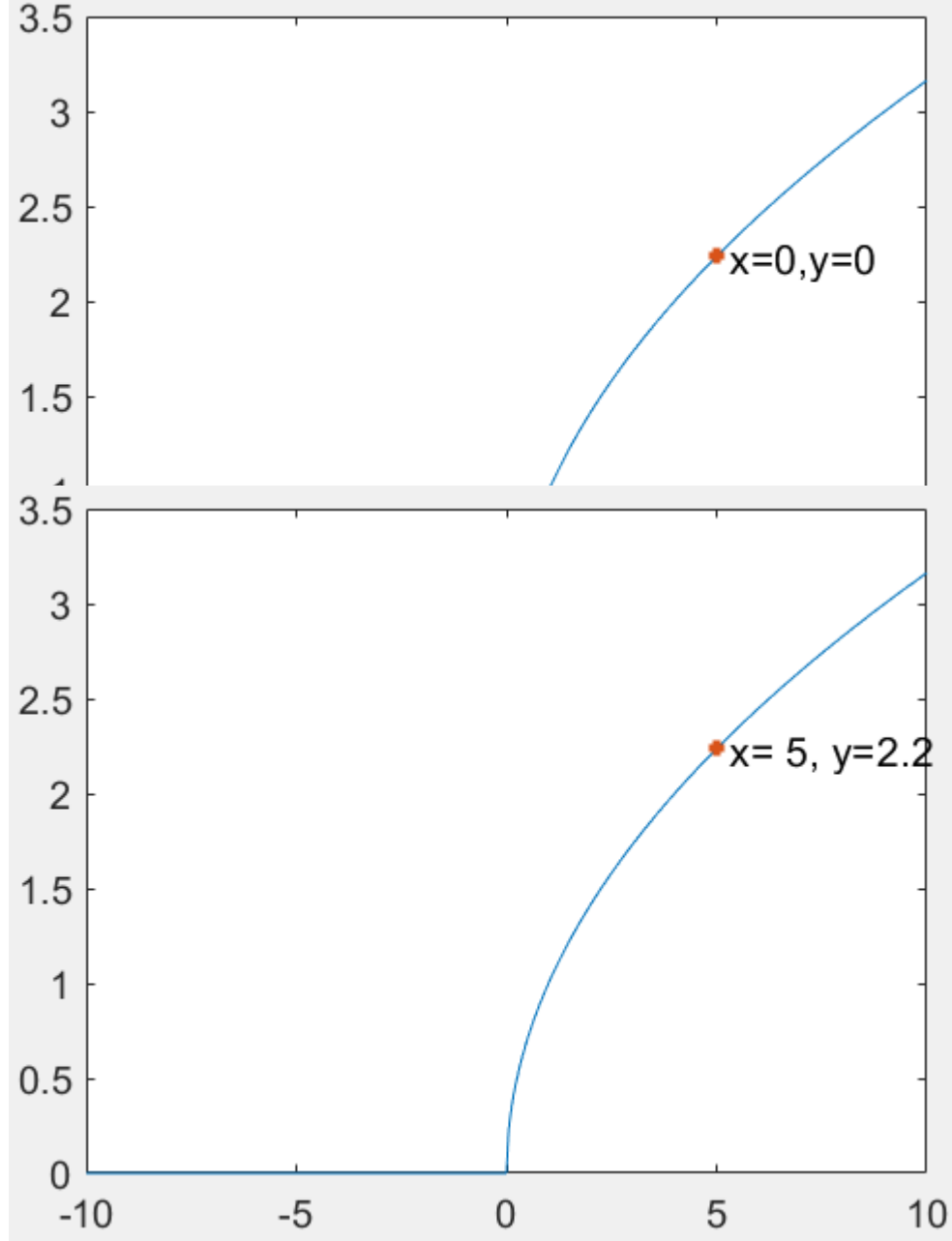
# text

```
clf
x = [-10:0.05:10];
y = sqrt(x); plot(x,y);
x0 = 5; y0 = sqrt(x0);
hold on, ...
plot(x0,y0,...
    '+', 'LineWidth', 3);
text(x0+0.3, y0, ...
    'x=0,y=0', ...
    'FontSize',16);
set(gca,'FontSize',15);
```



# text

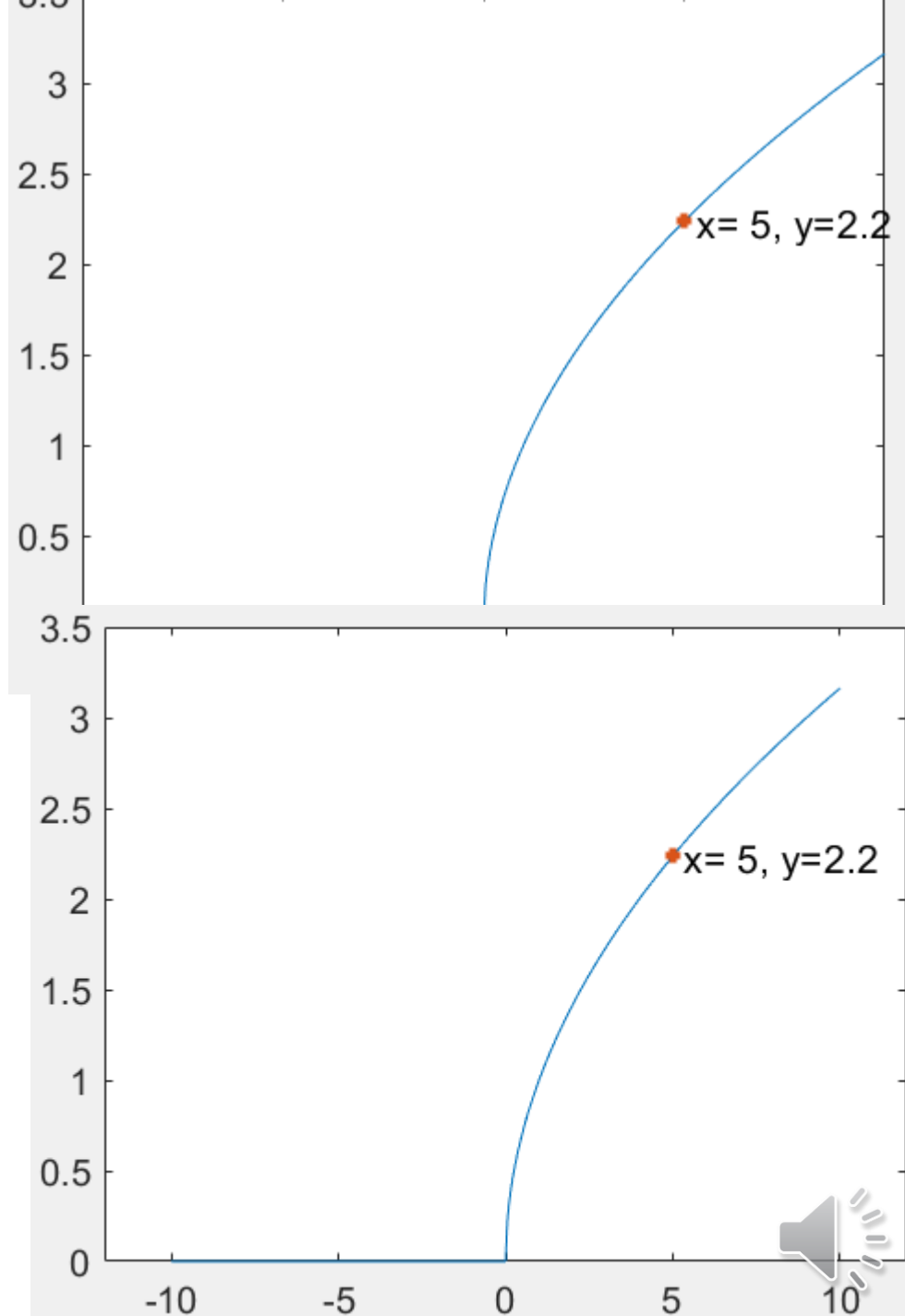
```
clf
x = [-10:0.05:10];
y = sqrt(x);
plot(x,y);
x0 = 5; y0 = sqrt(x0);
hold on,
plot(x0,y0,'+', 'LineWidth',
3);
s = sprintf('x=%2.0f,
y=%2.1f',x0, y0);
text(x0+0.3, y0, ...
    s, 'FontSize',16);
set(gca,'FontSize',15);
```



```
s = sprintf('x=%2.0f, y=%2.1f',x0, y0);
```

# text

```
clf
x = [-10:0.05:10];
y = sqrt(x); plot(x,y);
x0 = 5; y0 = sqrt(x0);
hold on,
plot(x0,y0,'+', 'LineWidth',
3);
s = sprintf('x=%2.0f,
y=%2.1f',x0, y0);
text(x0+0.3, y0, ...
    s, 'FontSize',16);
set(gca,'FontSize',15);
axis([-12 12 0 3.5]);
```



# The legend function

*legend*: create a legend with descriptive labels for each plotted data series.

Plot a set of curves:

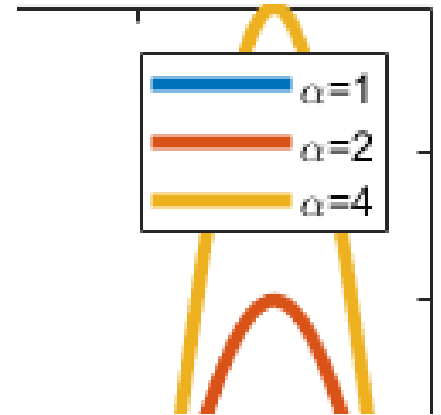
$$y = \alpha \sin(x) \text{ for } x \text{ in } [-10:0.05:10].$$

The values of  $\alpha$  are set to 1, 2 and 4.

Mark each curve with the  $\alpha$  value. Also, show the legend for all the curves.

Set the linewidth to 3.

Beautify the figure.



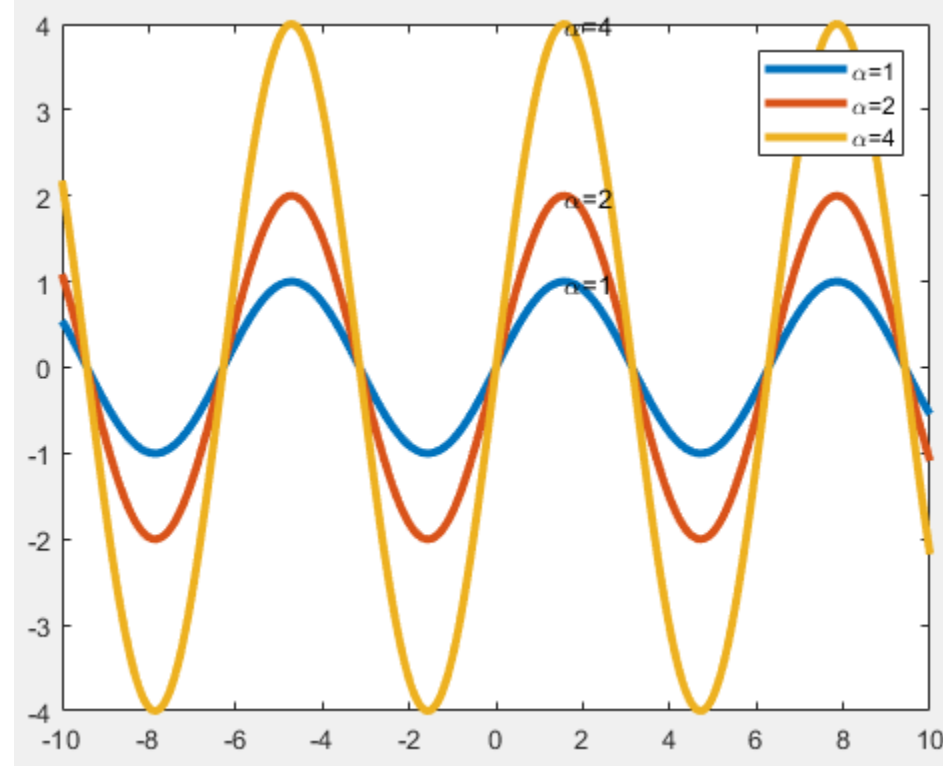
What's the  
structure  
plan?



# legend

## show legends

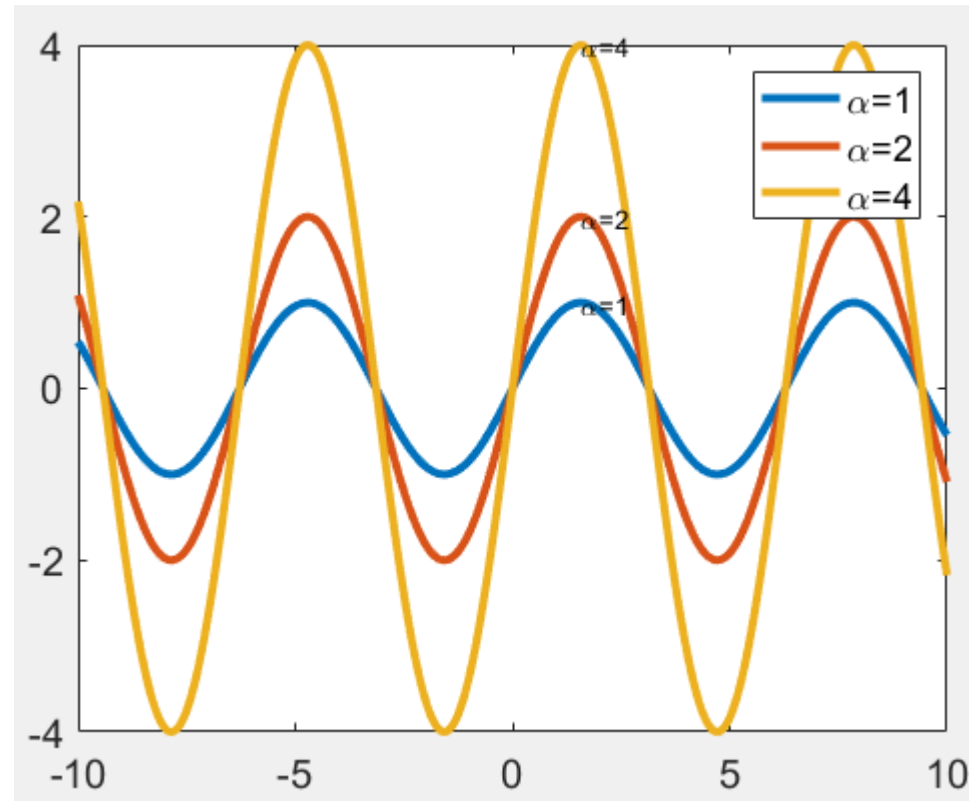
```
clf;clear
for a = [ 1 2 4]
    x = [-10:0.05:10];
    y = a.*sin(x);
    plot(x,y, 'LineWidth', 3);
    hold on
end
legend('\alpha=1', ...
       '\alpha=2', ...
       '\alpha=4');
hold off
text(pi/2, 1, '\alpha=1');
text(pi/2, 2, '\alpha=2');
text(pi/2, 4, '\alpha=4');
```



# legend

set font size in all the elements of figures

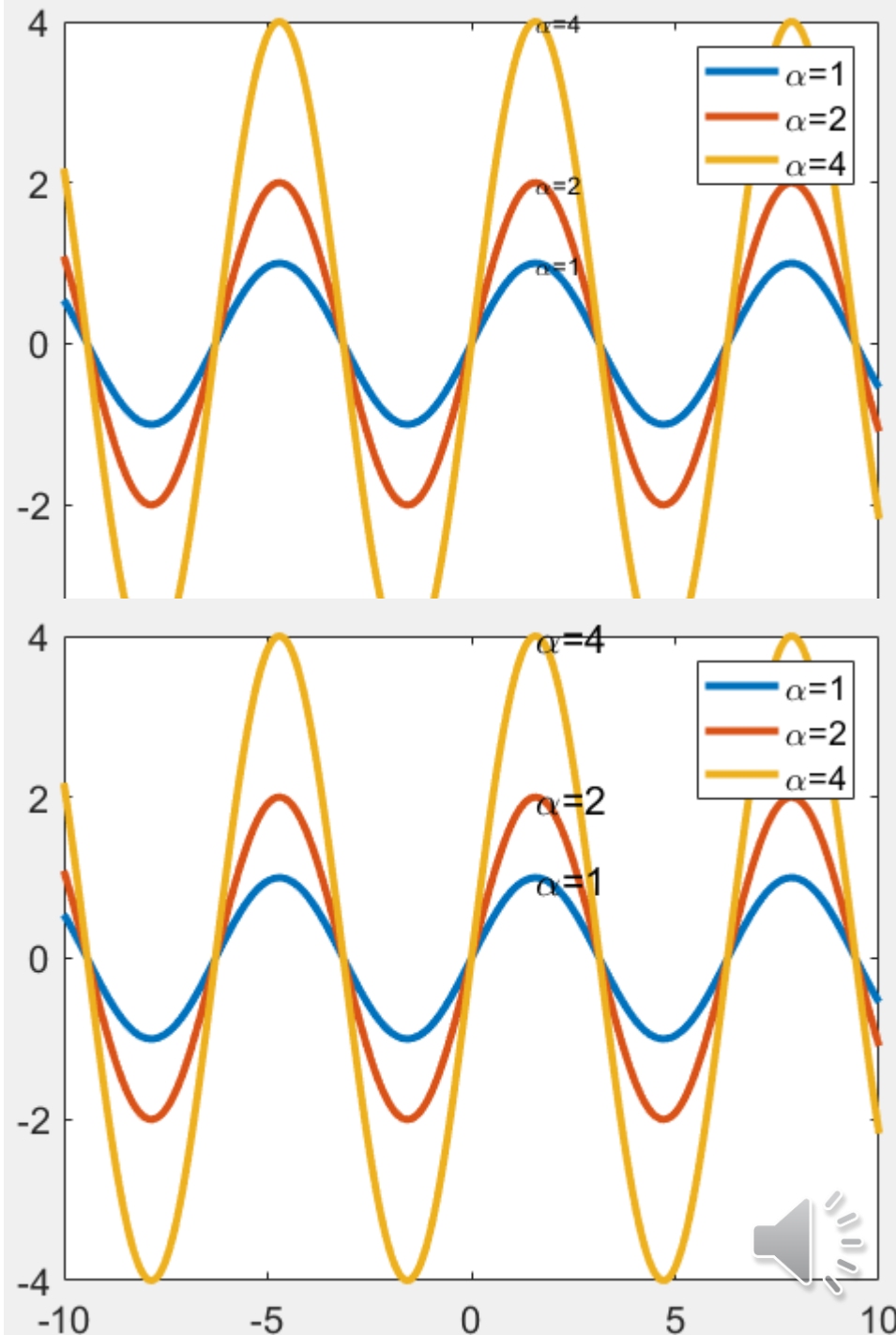
```
clf;clear
for a = [ 1 2 4]
    x = [-10:0.05:10];
    y = a.*sin(x);
    plot(x,y, 'LineWidth', 3);
    hold on
end
legend('\alpha=1', ...
       '\alpha=2', ...
       '\alpha=4');
hold off
text(pi/2, 1, '\alpha=1');
text(pi/2, 2, '\alpha=2');
text(pi/2, 4, '\alpha=4');
set(gca,'FontSize',15);
```



# legend

## set text font size

```
clf; clear
for a = [ 1 2 4]
    x = [-10:0.05:10];
    y = a.*sin(x);
    plot(x,y, 'LineWidth', 3);
    hold on
end
legend('\alpha=1', ...
       '\alpha=2', ...
       '\alpha=4');
hold off
text(pi/2, 1, '\alpha=1', 'FontSize',16);
text(pi/2, 2, '\alpha=2', 'FontSize',16);
text(pi/2, 4, '\alpha=4', 'FontSize',16);
set(gca,'FontSize',15);
```

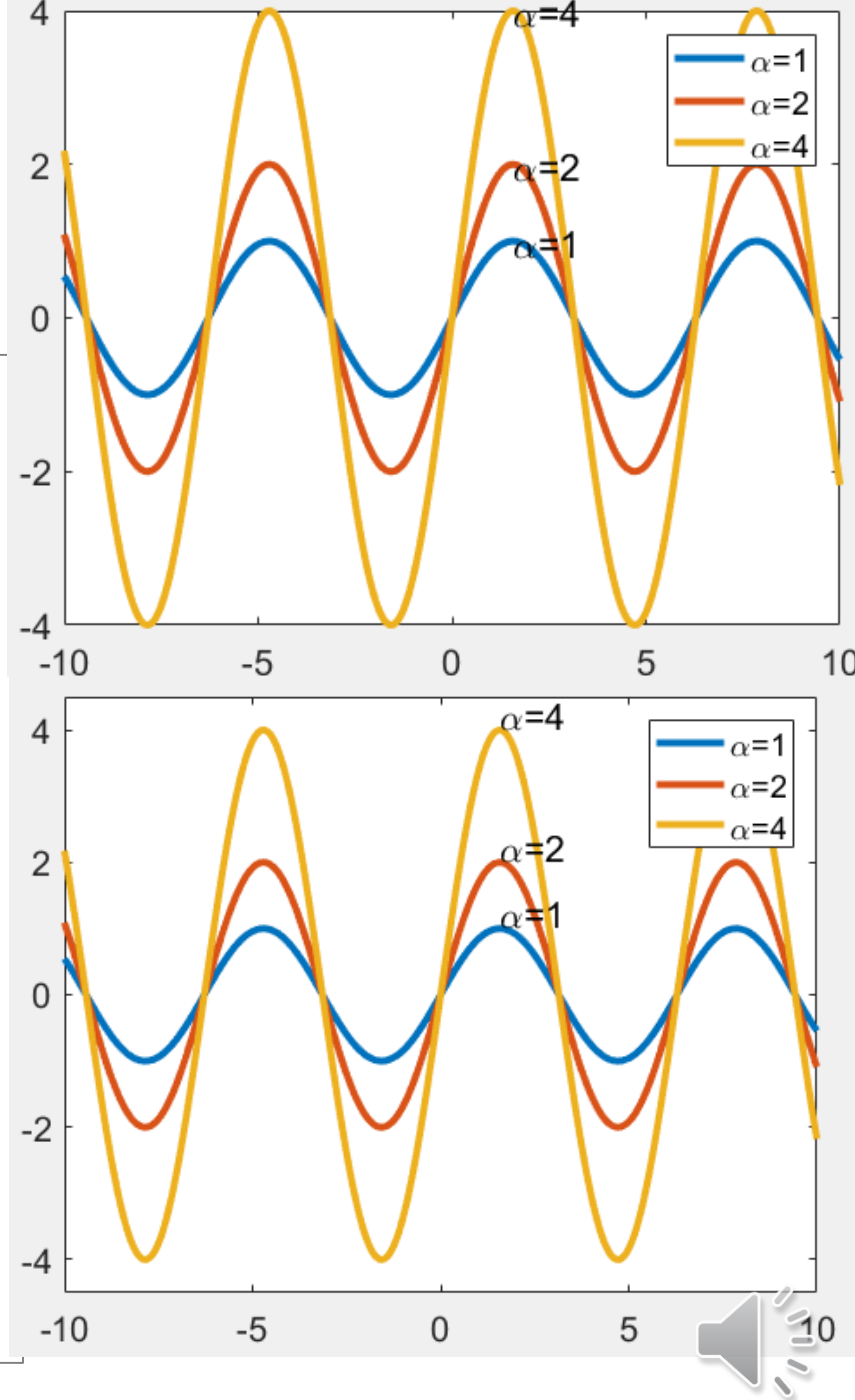




# legend

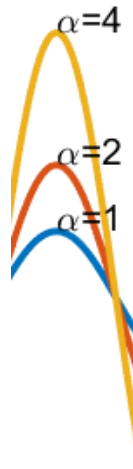
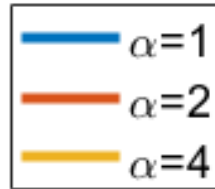
## set the axis range

```
clf; clear; ymin = 0; ymax = 0; %is it good?
for a = [ 1 2 4]
    x = [-10:0.05:10]; y = a.*sin(x);
    ymin = min([y ymin]);
    ymax = max([y ymax]);
    plot(x,y, 'LineWidth', 3);
    hold on
end
legend('\alpha=1', ...
       '\alpha=2', ...
       '\alpha=4');
hold off
text(pi/2, 1+0.25, '\alpha=1', 'FontSize',16);
text(pi/2, 2+0.25, '\alpha=2', 'FontSize',16);
text(pi/2, 4+0.25, '\alpha=4', 'FontSize',16);
set(gca,'FontSize',15);
axis([ min(x) max(x) (ymin -0.5) (ymax +0.5)]);
```



# legend: use string array

```
clf; clear; ymin = 0; ymax = 0;
for a = [ 1 2 4]
    x = [-10:0.05:10]; y = a.*sin(x);
    ymin = min([y ymin]);
    ymax = max([y ymax]);
    plot(x,y, 'LineWidth', 3);
    hold on
end
legend('\alpha=1', ...
    '\alpha=2', ...
    '\alpha=4');
hold off
text(pi/2, 1+0.25, '\alpha=1',
'FontSize',16);
text(pi/2, 2+0.25, '\alpha=2',
'FontSize',16);
text(pi/2, 4+0.25, '\alpha=4',
'FontSize',16);
set(gca,'FontSize',15);
axis([ min(x) max(x) (ymin -0.5)
(ymax +0.5)]);
```

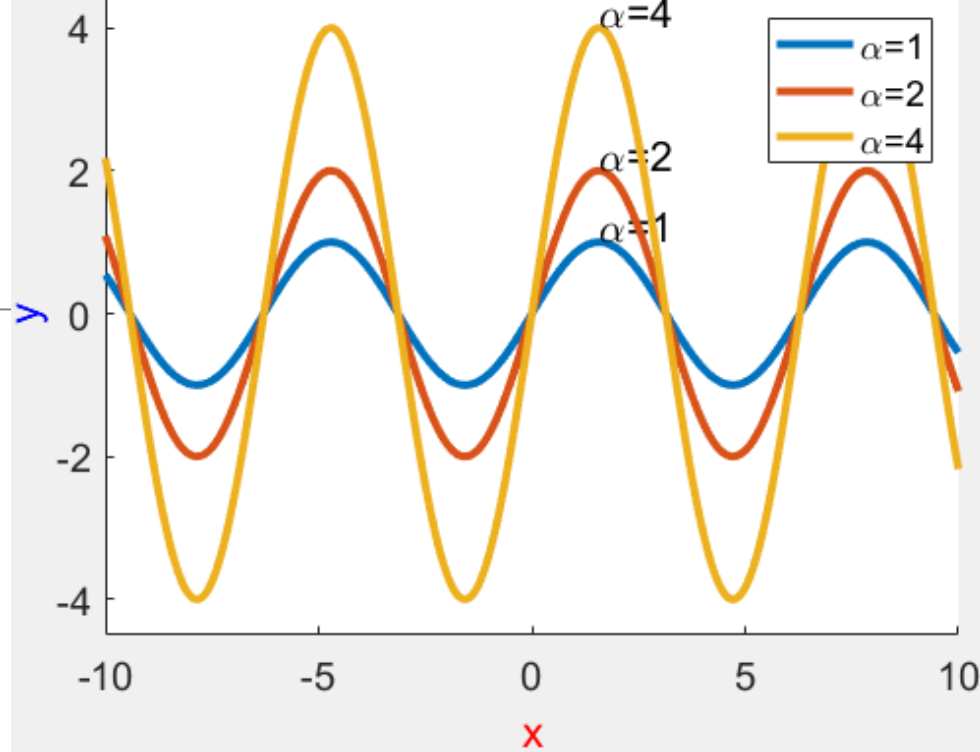


```
clear; clf; msg = [ ];
hold on; ymin = 0; ymax = 0; i =
1;
x = [-10:0.05:10];
for a = [ 1 2 4]
    y = a.*sin(x);
    ymin = min([y ymin]);
    ymax = max([y ymax]);
    plot(x,y, 'LineWidth', 3);
    s = sprintf('\alpha=%d',a);
    i = i + 1;
    text(pi/2, a+0.25, s, 'FontSize',16);
    msg = [msg; s];
end
%legend(msg(1),msg(2), msg(3));
legend(msg); hold off;
set(gca,'FontSize',15);
axis([ min(x) max(x) (ymin -0.5)
(ymax +0.5)]);
```



# legend color

```
clear; clf; msg = [ ];
hold on; ymin = 0; ymax = 0; i = 1;
x = [-10:0.05:10];
for a = [ 1 2 4]
    y = a.*sin(x);
    ymin = min([y ymin]);
    ymax = max([y ymax]);
    plot(x,y, 'LineWidth', 3);
    s = sprintf('\alpha=%d',a); %escape character
    i = i + 1;
    text(pi/2, a+0.25, s, 'FontSize',16);
    msg = [msg; s];
end
%legend(msg(1),msg(2), msg(3));
legend(msg); hold off; set(gca,'FontSize',15);
axis([ min(x) max(x) (ymin -0.5) (ymax +0.5)]);
xlabel('x', 'color', 'red');
ylabel('y', 'color', 'blue');
```



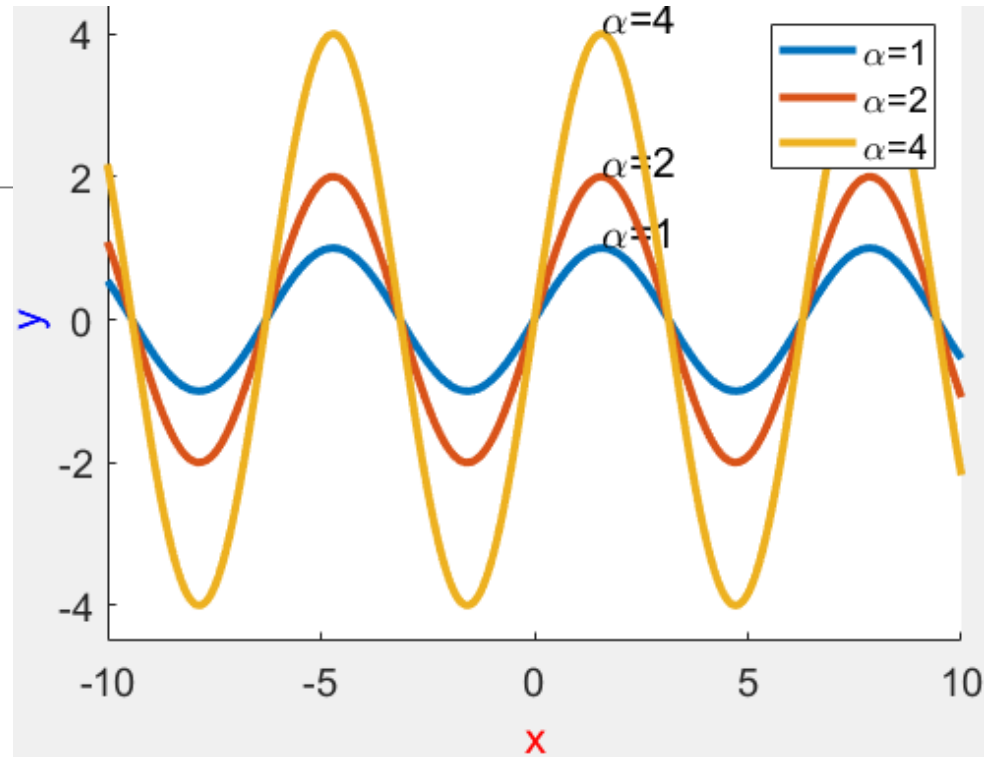
For a = 2,  
s =  
    '\alpha=2'

For a = 4,  
s =  
    '\alpha=4'



# legend color

```
clear; clf; msg = [ ];  
hold on; ymin = 0; ymax = 0; i = 1;  
x = [-10:0.05:10];  
for a = [ 1 2 4]  
    y = a.*sin(x);  
    ymin = min([y ymin]);  
    ymax = max([y ymax]);  
    plot(x,y, 'LineWidth', 3);  
    s = sprintf('\alpha=%d',a);  
    i = i + 1;  
    text(pi/2, a+0.25, s, 'FontSize',16);  
    msg = [msg; s];  
end  
%legend(msg(1),msg(2), msg(3));  
legend(msg); hold off; set(gca,'FontSize',15);  
axis([ min(x) max(x) (ymin -0.5) (ymax +0.5)]);  
xlabel('x', 'color', 'red');  
ylabel('y', 'color', 'blue');
```



msg =

3×8 char array

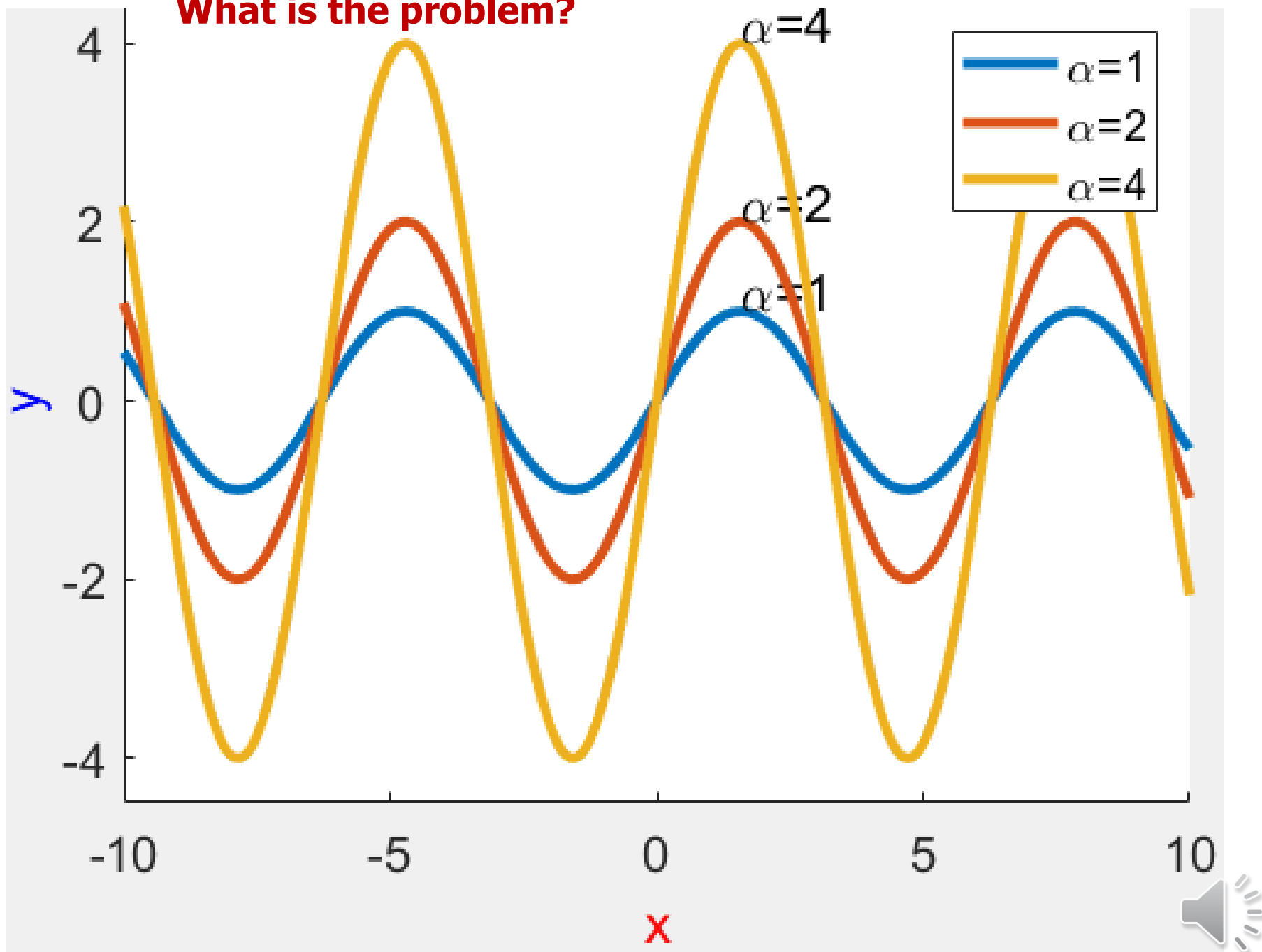
'\alpha=1'

'\alpha=2'

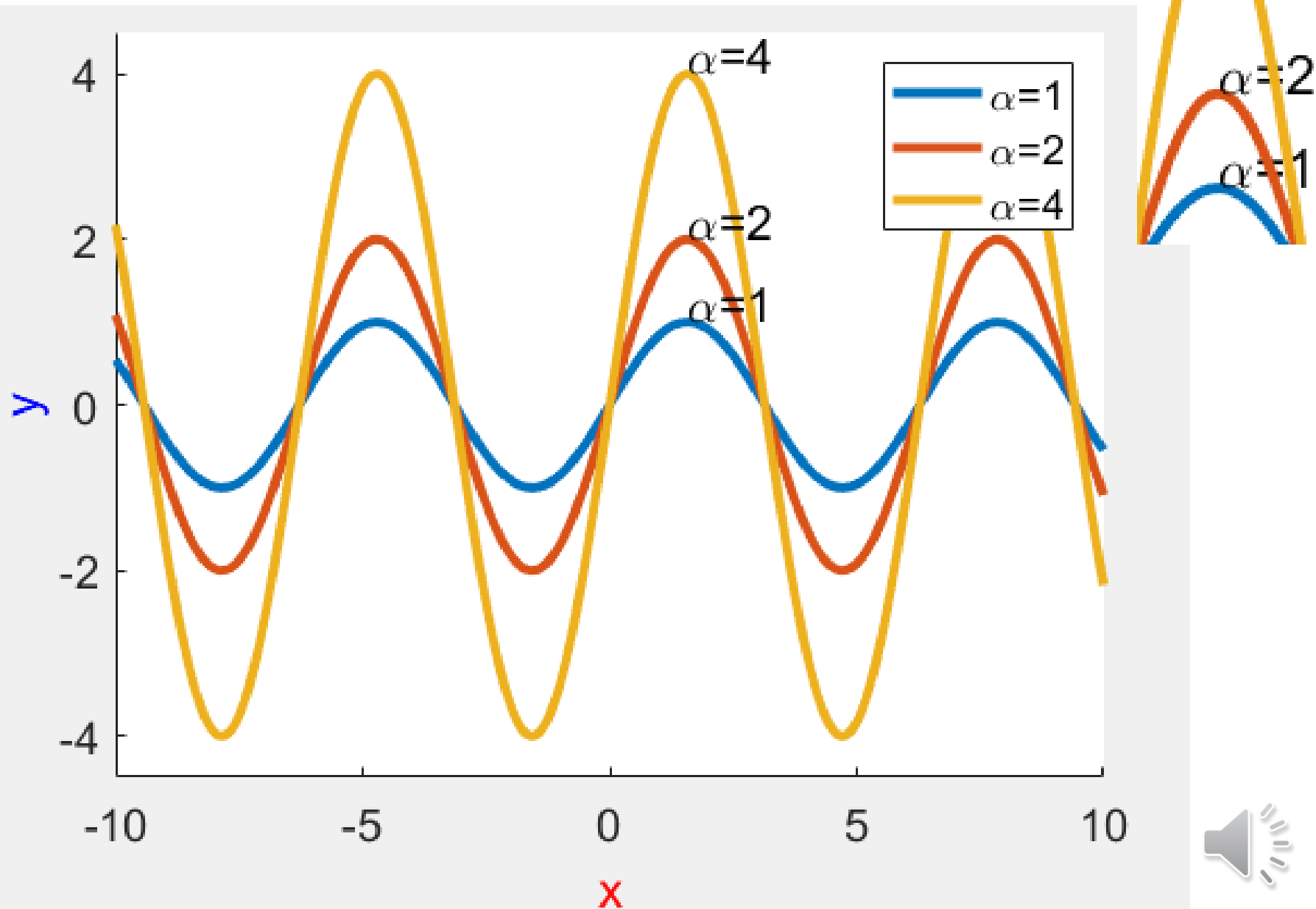
'\alpha=4'



**What is the problem?**



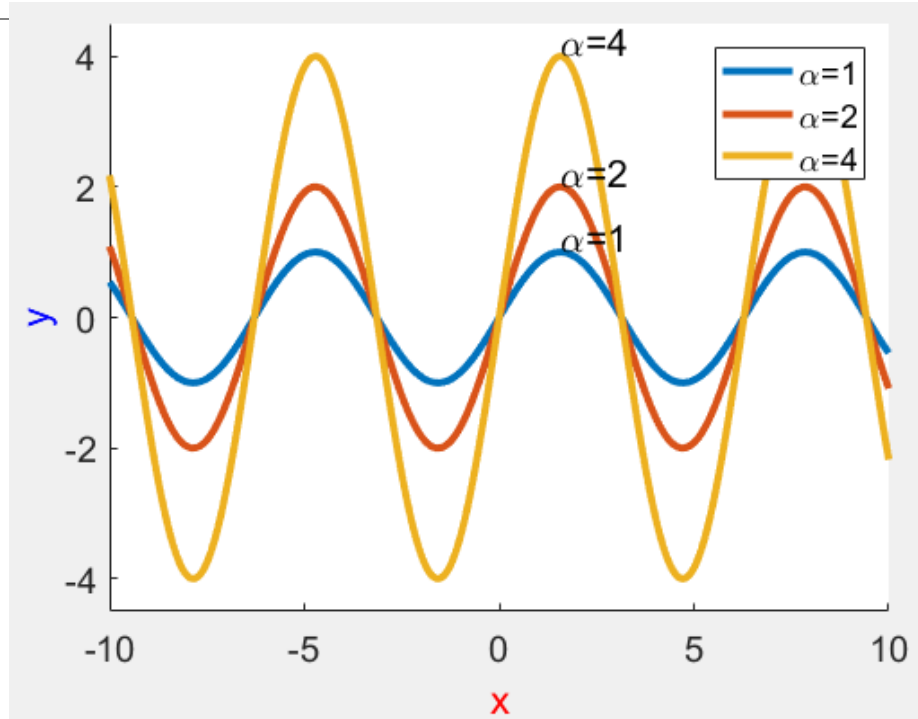
Solve the occlusion problem



# Legend: Solve the occlusion problem

## 1) draw curves; 2) draw labels

```
clear; clf; msg = []; hold on
ymin = 0; ymax = 0;
i = 1; A = [ 1 2 4];
for a = A
    x = [-10:0.05:10]; y = a.*sin(x);
    ymin = min([y ymin]);
    ymax = max([y ymax]);
    plot(x,y, 'LineWidth', 3);
    s = sprintf('\alpha=%d',a);
    i = i + 1; msg = [msg;s];
end
legend(msg);
for i = [1:3]
    text(pi/2, A(i)+0.25, msg(i,:), 'FontSize',16)
end
hold off; set(gca,'FontSize',15);
axis([ min(x) max(x) (ymin -0.5) (ymax +0.5)]);
xlabel('x', 'color', 'red');
ylabel('y', 'color', 'blue');
```



msg = 3×8 char array

'\alpha=1'

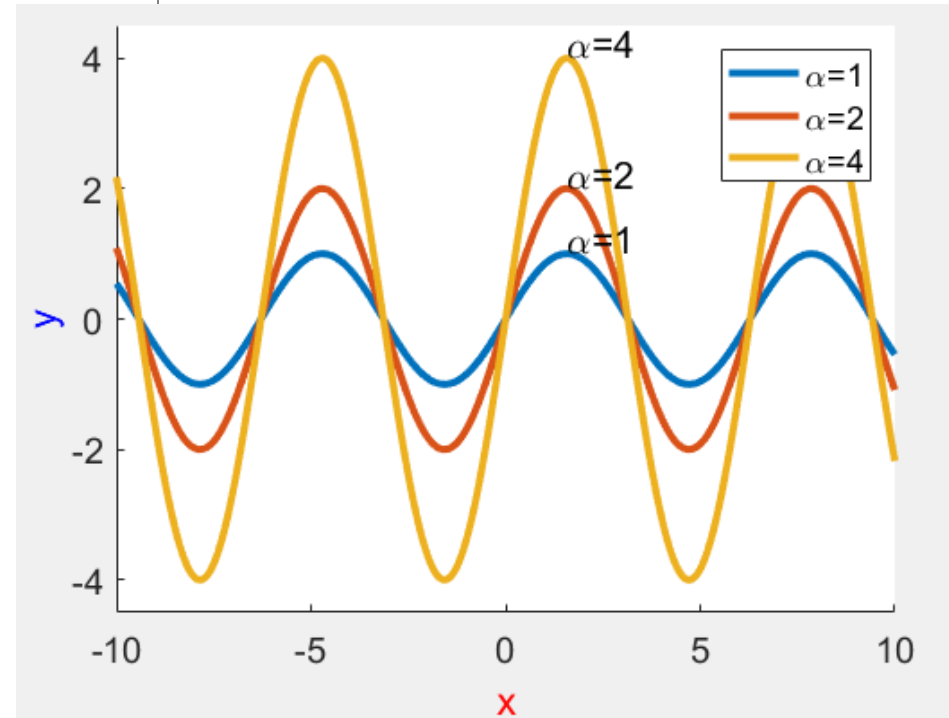
'\alpha=2'

'\alpha=4'



# Initialization problem

```
clear; clf; msg = []; hold on
ymin = -10000; ymax = 1000?
i = 1; A = [ 1 2 4];
for a = A
    x = [-10:0.05:10]; y = a.*sin(x);
    ymin = min([y ymin]);
    ymax = max([y ymax]);
    plot(x,y, 'LineWidth', 3);
    s = sprintf('\\alpha=%d',a);
    i = i + 1; msg = [msg;s];
end
legend(msg);
for i = [1:3]
    text(pi/2, A(i)+0.25, msg(i,:), 'FontSize',16)
end
hold off; set(gca,'FontSize',15);
axis([ min(x) max(x) (ymin -0.5) (ymax +0.5)]);
xlabel('x', 'color', 'red');
ylabel('y', 'color', 'blue');
```





# Symbols

Character Sequence	Symbol	Character Sequence	Symbol	Character Sequence	Symbol
\alpha	$\alpha$	\upsilon	$\upsilon$	\sim	$\sim$
\angle	$\angle$	\phi	$\phi$	\leq	$\leq$
\ast	$*$	\chi	$\chi$	\infty	$\infty$
\beta	$\beta$	\psi	$\psi$	\clubsuit	$\clubsuit$
\gamma	$\gamma$	\omega	$\omega$	\diamondsuit	$\diamondsuit$
\delta	$\delta$	\Gamma	$\Gamma$	\heartsuit	$\heartsuit$
\epsilon	$\epsilon$	\Delta	$\Delta$	\spadesuit	$\spadesuit$
\zeta	$\zeta$	\Theta	$\Theta$	\leftarrow	$\leftarrow$
				\rightarrow	$\rightarrow$



# Exercise

## legend

Plot a set of curves:

$$y = e^{\beta \cos(\omega x)} \text{ for } x \text{ in } [0:0.01:2\pi].$$

The values of  $\beta$  are set to 1, 2.

The values of  $\omega$  are set to 1, 2.

Mark each curve with the  $\beta$  and  $\omega$  values. Also, show the legend of each curve. Set the linewidth to 3. Set the title as  $y = e^{\beta \cos(\omega x)}$ .

Beautify the figure.



# Exercise: legend

Plot a set of curves:

$$y = e^{\beta \cos(\omega x)} \text{ for } x \text{ in } [0:0.01:2\pi].$$

The values of  $\beta$  are set to 1, 2.

The values of  $\omega$  are set to 1, 2.

Mark each curve with the  $\beta$  and  $\omega$  values. Also, show the legend of each curve. Set the linewidth to 3. Set the title as  $y = e^{\beta \cos(\omega x)}$ .

Beautify the figure.

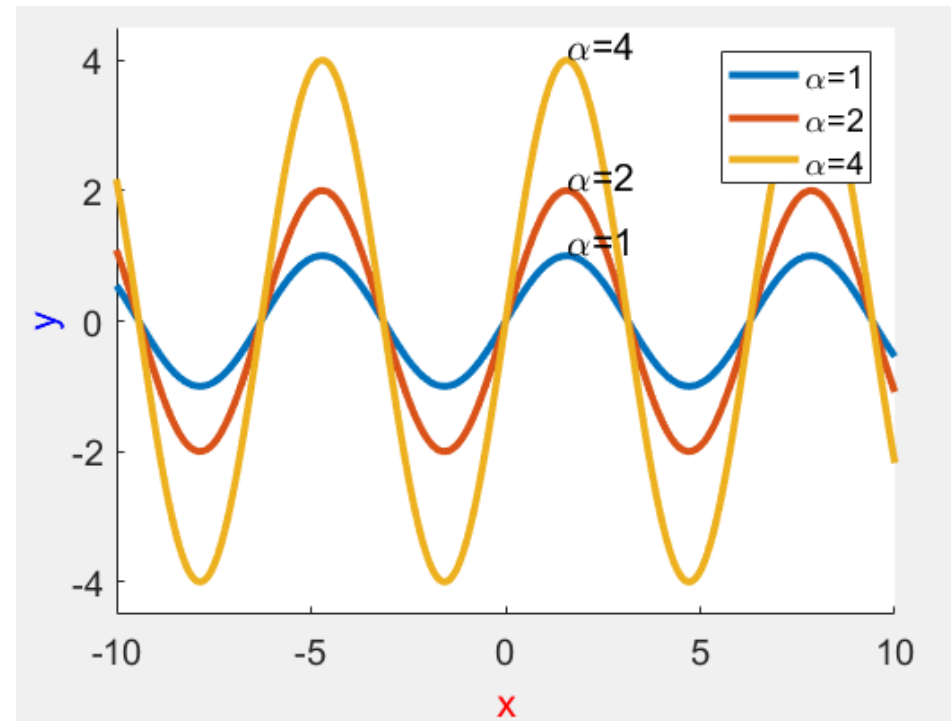
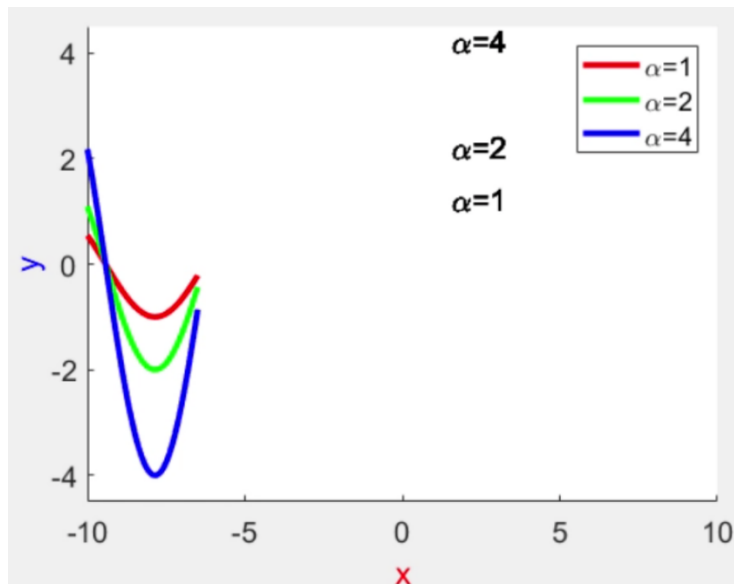
The structure plan:

1. Draw all the curves
2. Draw all the labels
3. Draw the legend
4. Set the font size of numbers to be large enough
5. Label the axes. Set the axis ranges properly.

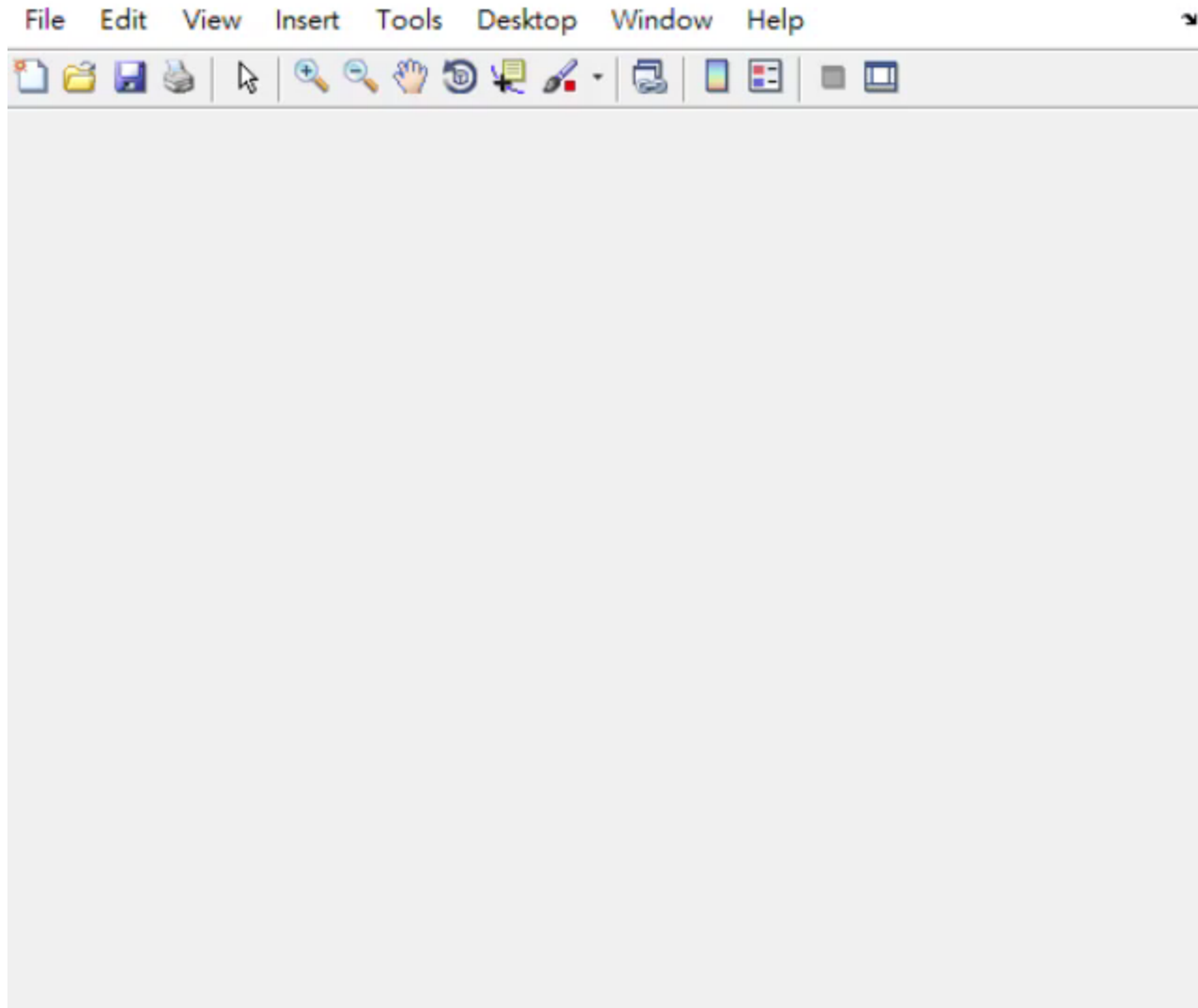


# Curve animation

Draw the three curves  
in an animation.



# Demo. Curve animation

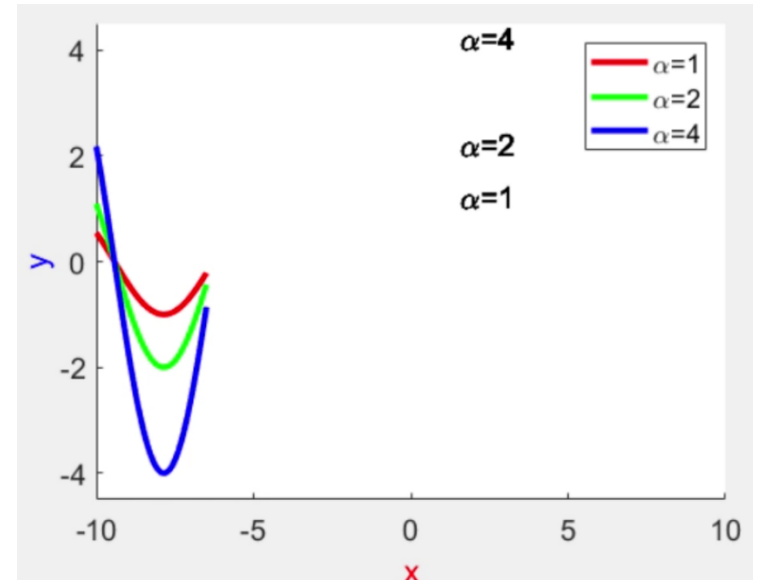


# Curve animation

Draw the three curves  
in an animation.

Plan:

1. Set the range of  $x$ .
2. Produce the sample points of the three curves.
3. Draw the three curves.
4. Update the range of  $x$ .
5. Repeat 2-4 until finish.



# Curve Animation: First Attempt

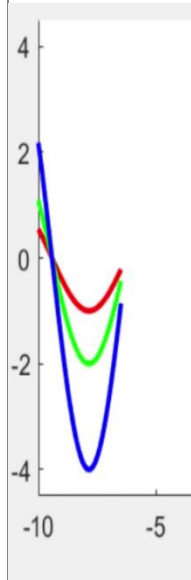
```
clf; clear;
input('Press Enter to start...');
for xUpper = [-10:0.25:10]
    hold on
    ymin = 0; ymax = 0;
    msg = []; i = 1; A = [ 1 2 4];
    color= ['r' 'g' 'b'];
    for a = A
        x = [-10:0.05:xUpper];
        y = a.*sin(x);
        ymin = min([y ymin]);
        ymax = max([y ymax]);
        plot(x,y, 'LineWidth', 3, ...
            'color', color(i));
```

```
        s = sprintf('\\alpha=%d',a);
        i = i + 1;
        msg = [msg;s];
    end
    legend(msg);
    for i = [1:3]
        text(pi/2, A(i)+0.25, ...
            msg(i,:), 'FontSize',16)
    end
    hold off
    set(gca,'FontSize',15);
    axis([ -10 10 -4.5 4.5]);
    xlabel('x', 'color', 'red');
    ylabel('y', 'color', 'blue');
    pause(0.033);
end
```



# Curve Animation: First Attempt

```
clf; clear;
input('Press Enter to start...');
for xUpper = [-10:0.25:10]
    hold on
    ymin = 0; ymax = 0;
    msg = []; i = 1; A = [ 1 2 4];
    color= ['r' 'g' 'b'];
    for a = A
        x = [-10:0.05:xUpper];
        y = a.*sin(x);
        ymin = min([y ymin]);
        ymax = max([y ymax]);
        plot(x,y, 'LineWidth', 3, ...
            'color', color(i));
```



```
        s = sprintf('\\alpha=%d',a);
        i = i + 1;
        msg = [msg;s];
    end
    legend(msg);
    for i = [1:3]
        text(pi/2, A(i)+0.25, ...
            msg(i,:), 'FontSize',16)
    end
    hold off
    set(gca,'FontSize',15);
    axis([ -10 10 -4.5 4.5]);
    xlabel('x', 'color', 'red');
    ylabel('y', 'color', 'blue');
    pause(0.033);
end
```

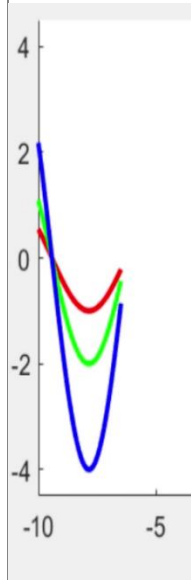
**What is the problem?**





# Curve Animation: First Attempt

```
clf; clear;
input('Press Enter to start...');
for xUpper = [-10:0.25:10]
    hold on
    ymin = 0; ymax = 0;
    msg = []; i = 1; A = [ 1 2 4];
    color= ['r' 'g' 'b'];
    for a = A
        x = [-10:0.05:xUpper];
        y = a.*sin(x);
        ymin = min([y ymin]);
        ymax = max([y ymax]);
        plot(x,y, 'LineWidth', 3, ...
            'color', color(i));
```



```
        s = sprintf('\\alpha=%d',a);
        i = i + 1;
        msg = [msg;s];
    end
    legend(msg);
    for i = [1:3]
        text(pi/2, A(i)+0.25, ...
            msg(i,:), 'FontSize',16)
    end
    hold off
    set(gca,'FontSize',15);
    axis([ -10 10 -4.5 4.5]);
    xlabel('x', 'color', 'red');
    ylabel('y', 'color', 'blue');
    pause(0.033);
end
```

**Many curves to be drawn.  
The figure keeps all the curves.**



# Curve Animation

```
clf; clear;
input('Press Enter to start...');
for xUpper = [-10:0.25:10]
    clf; hold on
    ymin = 0; ymax = 0;
    msg = []; i = 1; A = [ 1 2 4];
    color= ['r' 'g' 'b'];
    for a = A
        x = [-10:0.05:xUpper];
        y = a.*sin(x);
        ymin = min([y ymin]);
        ymax = max([y ymax]);
        plot(x,y, 'LineWidth', 3, ...
            'color', color(i));
```

```
        s = sprintf('\\alpha=%d',a);
        i = i + 1;
        msg = [msg;s];
    end
    legend(msg);
    for i = [1:3]
        text(pi/2, A(i)+0.25, ...
            msg(i,:), 'FontSize',16)
    end
    hold off
    set(gca,'FontSize',15);
    axis([ -10 10 -4.5 4.5]);
    xlabel('x', 'color', 'red');
    ylabel('y', 'color', 'blue');
    pause(0.033);
end
```

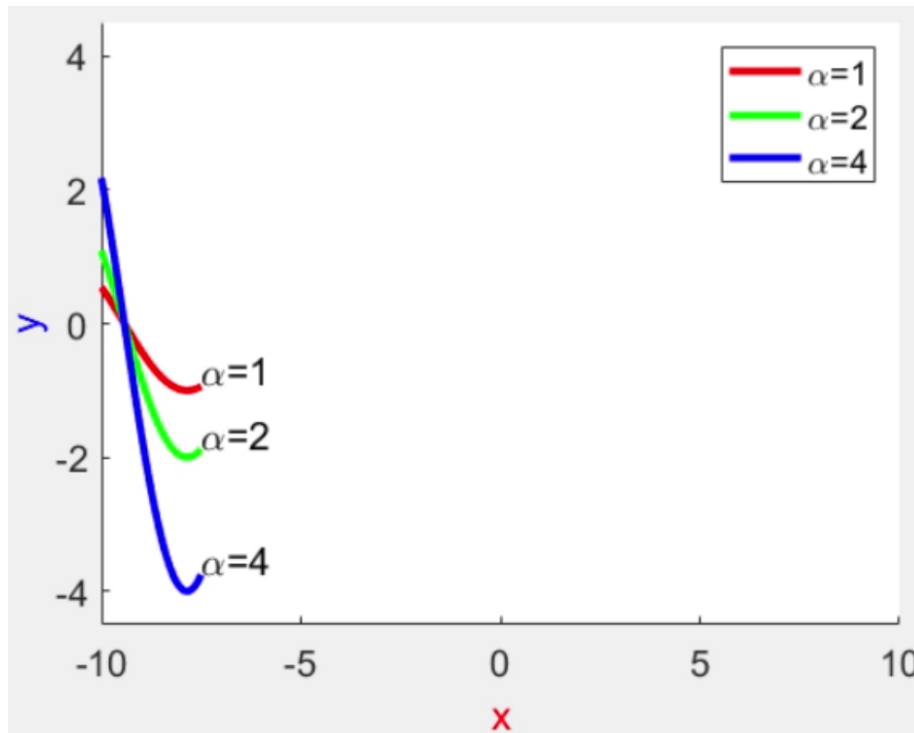
**There are too many curves  
drawn in the same figure.**



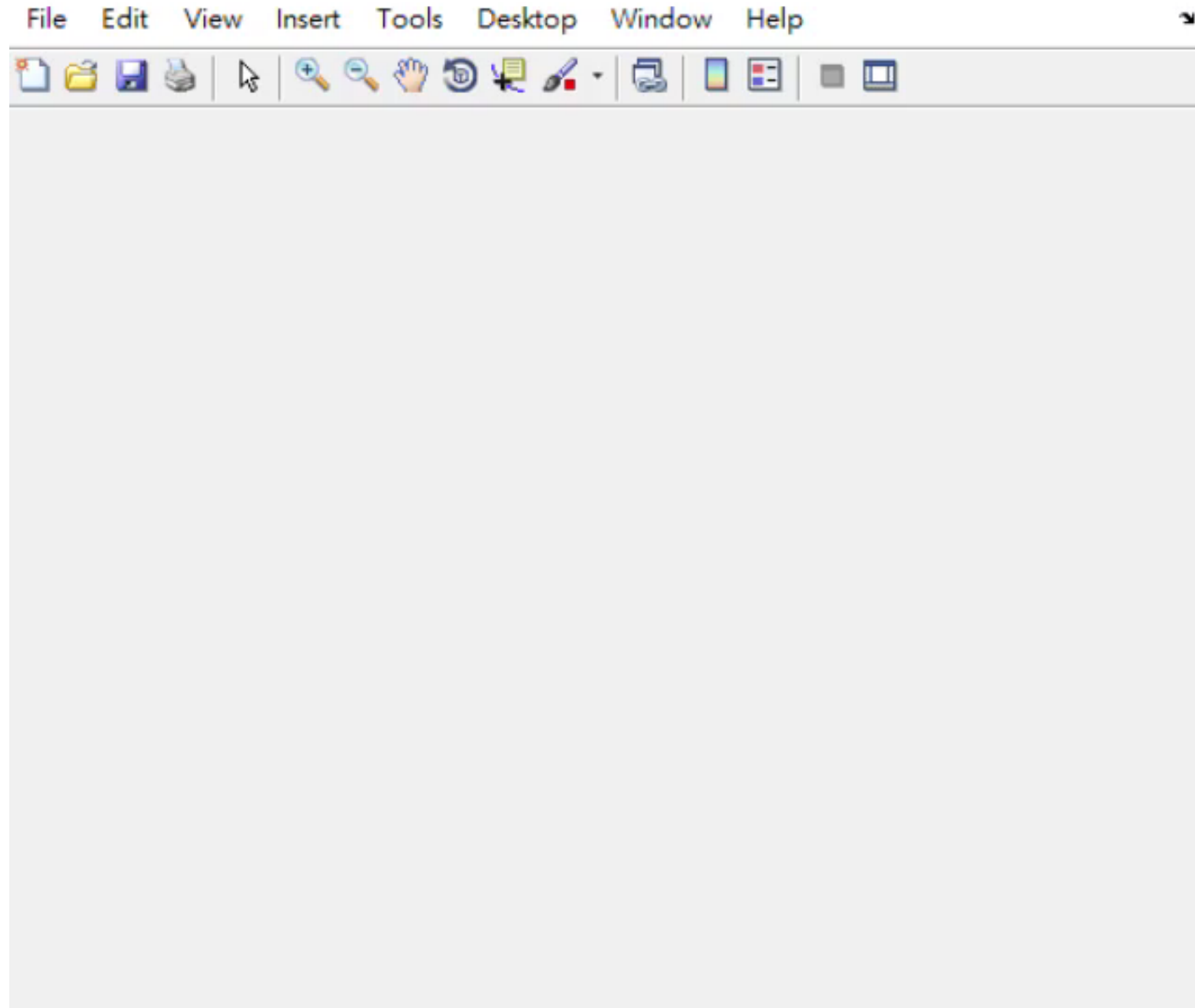
# Curve animation

Draw the three curves  
in an animation.

Mark the curves while they are being drawn.



# Demo. Curve animation



# Problem(s)? Window size

```
clf; clear
input('Press Enter to start...');
for xUpper = [-10:0.25:10]
    clf; hold on; ymin = 0; ymax = 0;
    msg = []; i = 1; A = [ 1 2 4];
    color= ['r' 'g' 'b'];
    x = [-10:0.05:xUpper];
    sx = size(x);
    y = zeros(3, sx(2));
    for a = A
        y(i,:) = a.*sin(x);
        plot(x,y(i,:), 'LineWidth', 3, ...
            'color', color(i));
```

```
        s = sprintf('\\alpha=%d',a);
        i = i + 1;
        msg = [msg;s];
    end
    legend(msg);

    for i = [1:3]
        text(x(end), y(i,end)+0.25, ...
            msg(i,:), 'FontSize',16)
    end

    hold off
    set(gca,'FontSize',15);
    axis([ -10 10 -4.5 4.5]);
    xlabel('x', 'color', 'red');
    ylabel('y', 'color', 'blue');
    pause(0.033);
end
```

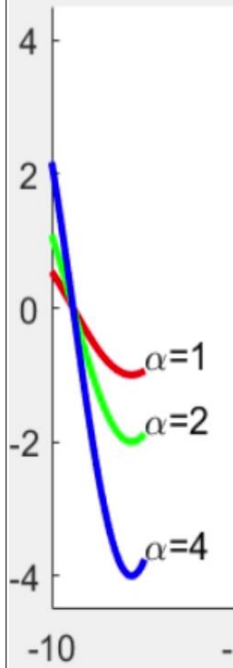


figure  
flickering



# Curve Animation. Marking

```
clf; clear
input('Press Enter to start...');
for xUpper = [-10:0.25:10]
    clf; hold on; ymin = 0; ymax = 0;
    msg = []; i = 1; A = [ 1 2 4];
    color= ['r' 'g' 'b'];
    x = [-10:0.05:xUpper];
    sx = size(x);
    y = zeros(3, sx(2));
    for a = A
        y(i,:) = a.*sin(x);
        plot(x,y(i,:), 'LineWidth', 3, ...
            'color', color(i));
```

```
        s = sprintf('\\alpha=%d',a);
        i = i + 1;
        msg = [msg;s];
    end
    legend(msg);

    for i = [1:3]
        text(x(end), y(i,end)+0.25, ...
            msg(i,:), 'FontSize',16)
    end

    hold off
    set(gca,'FontSize',15);
    axis([ -10 10 -4.5 4.5]);
    xlabel('x', 'color', 'red');
    ylabel('y', 'color', 'blue');
    pause(0.033);
end
```

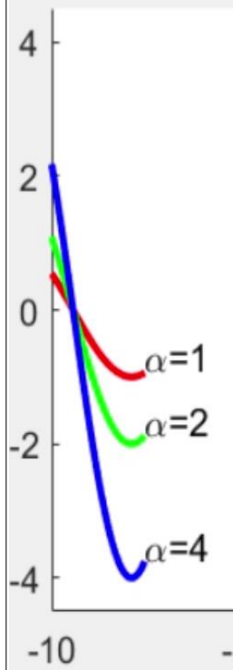


figure  
flickering



# Problem(s)? Window size

```
close all; clf; clear
figure, axis([-10 10 -4.5 4.5]);
input('Press Enter to start...');
for xUpper = [-10:0.25:10]
    clf; hold on; ymin = 0; ymax = 0;
    msg = []; i = 1; A = [ 1 2 4];
    color= ['r' 'g' 'b'];
    x = [-10:0.05:xUpper];
    sx = size(x);
    y = zeros(3, sx(2));
    for a = A
        y(i,:) = a.*sin(x);
        plot(x,y(i,:), 'LineWidth', 3, ...
            'color', color(i));
```

```
        s = sprintf('\\alpha=%d',a);
        i = i + 1;
        msg = [msg;s];
    end
    legend(msg);

    for i = [1:3]
        text(x(end), y(i,end)+0.25, ...
            msg(i,:), 'FontSize',16)
    end

    hold off
    set(gca,'FontSize',15);
    xlabel('x', 'color', 'red');
    ylabel('y', 'color', 'blue');
    pause(0.033);
end
```



# Problem(s)? Window size

```
close all; clf; clear
figure, axis([-10 10 -4.5 4.5]);
set(gca,'FontSize',15);
xlabel('x', 'color', 'red');
ylabel('y', 'color', 'blue');
input('Press Enter to start...');

for xUpper = [-10:0.25:10]
    clf; hold on; ymin = 0; ymax = 0;
    msg = []; i = 1; A = [ 1 2 4];
    color= ['r' 'g' 'b'];
    x = [-10:0.05:xUpper];
    sx = size(x);
    y = zeros(3, sx(2));
```

```
    for a = A
        y(i,:) = a.*sin(x);
        plot(x,y(i,:), 'LineWidth', 3, ...
            'color', color(i));
        s = sprintf('\alpha=%d',a);
        i = i + 1;
        msg = [msg;s];
    end

    legend(msg);

    for i = [1:3]
        text(x(end), y(i,end)+0.25, ...
            msg(i,:), 'FontSize',16)
    end

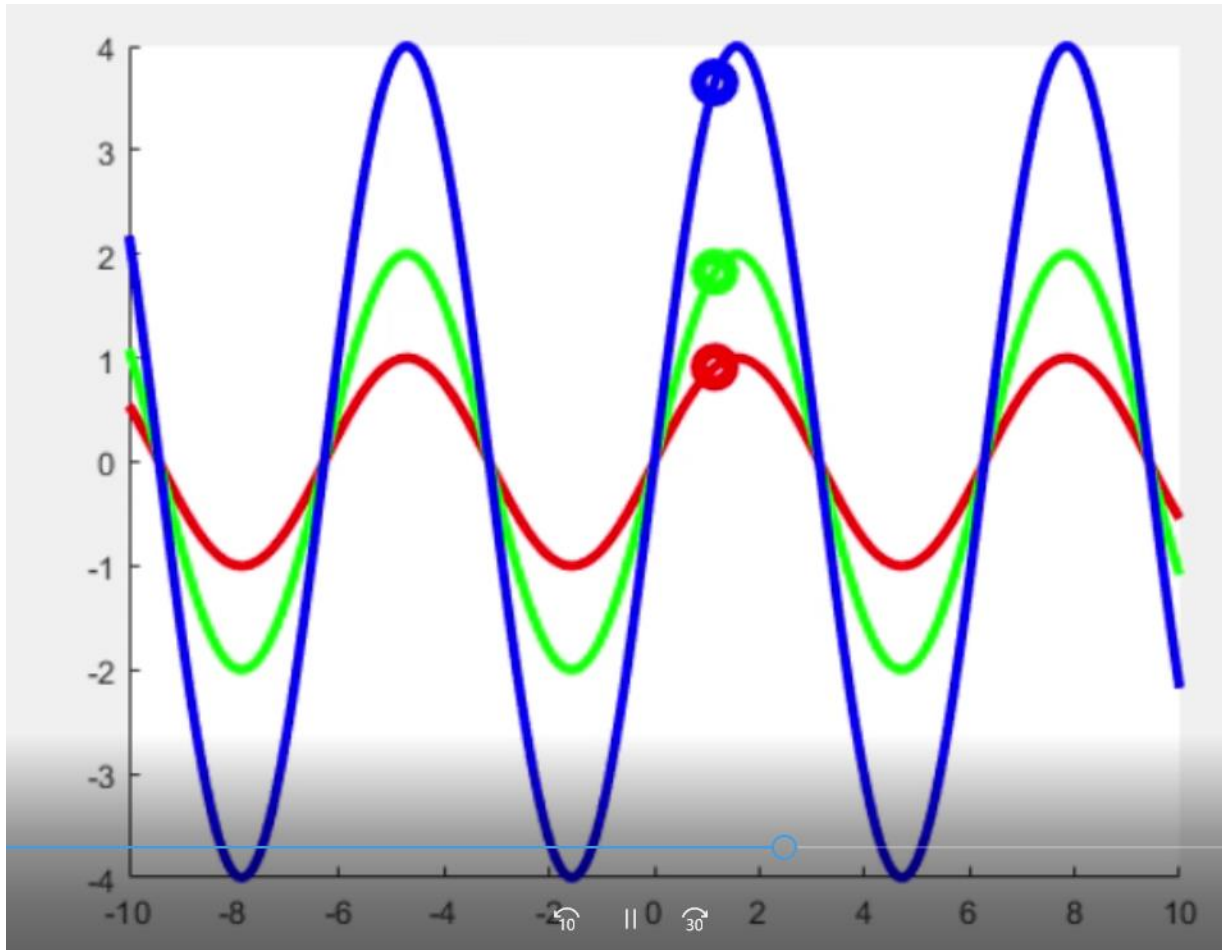
    hold off
    pause(0.033);
end
```



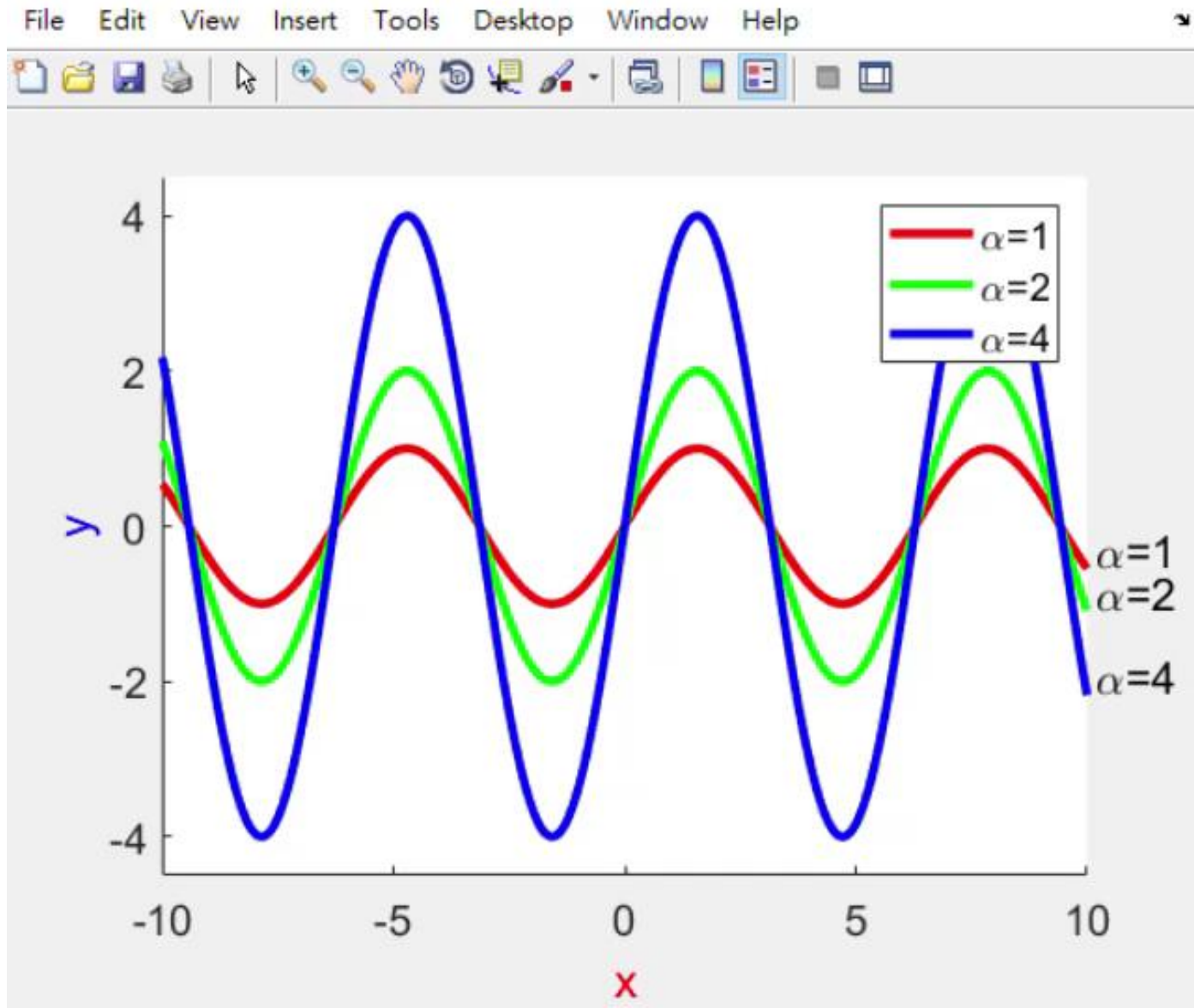


# Point animation

Draw the three curves. Use a point to trace each curve.



# Demo. Point animation



# Point Animation

```
close all; clf; clear; hold on;
ymin = 0; ymax = 0;
msg = []; i = 1; A = [ 1 2 4];
color= ['r' 'g' 'b'];
set(gca,'FontSize',15);
axis([ -10 10 -4.5 4.5]);
xlabel('x', 'color', 'red');
ylabel('y', 'color', 'blue');
x = [-10:0.05:10]; sx = size(x);
y = zeros(3, sx(2));
for a = A
    y(i,:) = a.*sin(x);
    i = i + 1;
end
input('Press Enter to start...');
```

Precompute the sample points of the curves and then run over the sample points one by one to display the tracing points.

```
for xpoint = [-10:0.05:10]
    clf; hold on
    i = 1;
    for a = A

        y0 = a.*sin(xpoint);

        plot(x,y(i,:), 'LineWidth', 3,...
            'color', color(i));

        plot(xpoint, y0, 'o', ...
            'MarkerSize', 10, ...
            'LineWidth', 5,...
            'color', color(i));
        i = i + 1;
    end
    pause(0.033);
end
hold off
```



# Point Animation

```
close all; clf; clear; hold on;
ymin = 0; ymax = 0;
msg = []; i = 1; A = [ 1 2 4];
color= ['r' 'g' 'b'];
set(gca,'FontSize',15);
axis([ -10 10 -4.5 4.5]);
xlabel('x', 'color', 'red');
ylabel('y', 'color', 'blue');
x = [-10:0.05:10]; sx = size(x);
y = zeros(3, sx(2));
for a = A
    y(i,:) = a.*sin(x);
    i = i + 1;
end
input('Press Enter to start...');
```

Precompute the sample points of the curves and then run over the sample points one by one to display the tracing points.

```
for xpoint = [-10:0.05:10]
    clf; hold on
    i = 1;
    for a = A

        y0 = a.*sin(xpoint);

        plot(x,y(i,:), 'LineWidth', 3,...
            'color', color(i));

        plot(xpoint, y0, 'o', ...
            'MarkerSize', 10, ...
            'LineWidth', 5,...
            'color', color(i));
        i = i + 1;
    end
    pause(0.033);
end
hold off
```



# Point Animation

```
close all; clf; clear; hold on;
ymin = 0; ymax = 0;
msg = []; i = 1; A = [ 1 2 4];
color= ['r' 'g' 'b'];
set(gca,'FontSize',15);
axis([ -10 10 -4.5 4.5]);
xlabel('x', 'color', 'red');
ylabel('y', 'color', 'blue');
x = [-10:0.05:10]; sx = size(x);
y = zeros(3, sx(2));
for a = A
    y(i,:) = a.*sin(x);
    i = i + 1;
end
input('Press Enter to start...');
```

Precompute the sample points of the curves and then run over the sample points one by one to display the tracing points.

```
for xpoint = [-10:0.05:10]
    clf; hold on
    i = 1;
    for a = A

        y0 = a.*sin(xpoint);

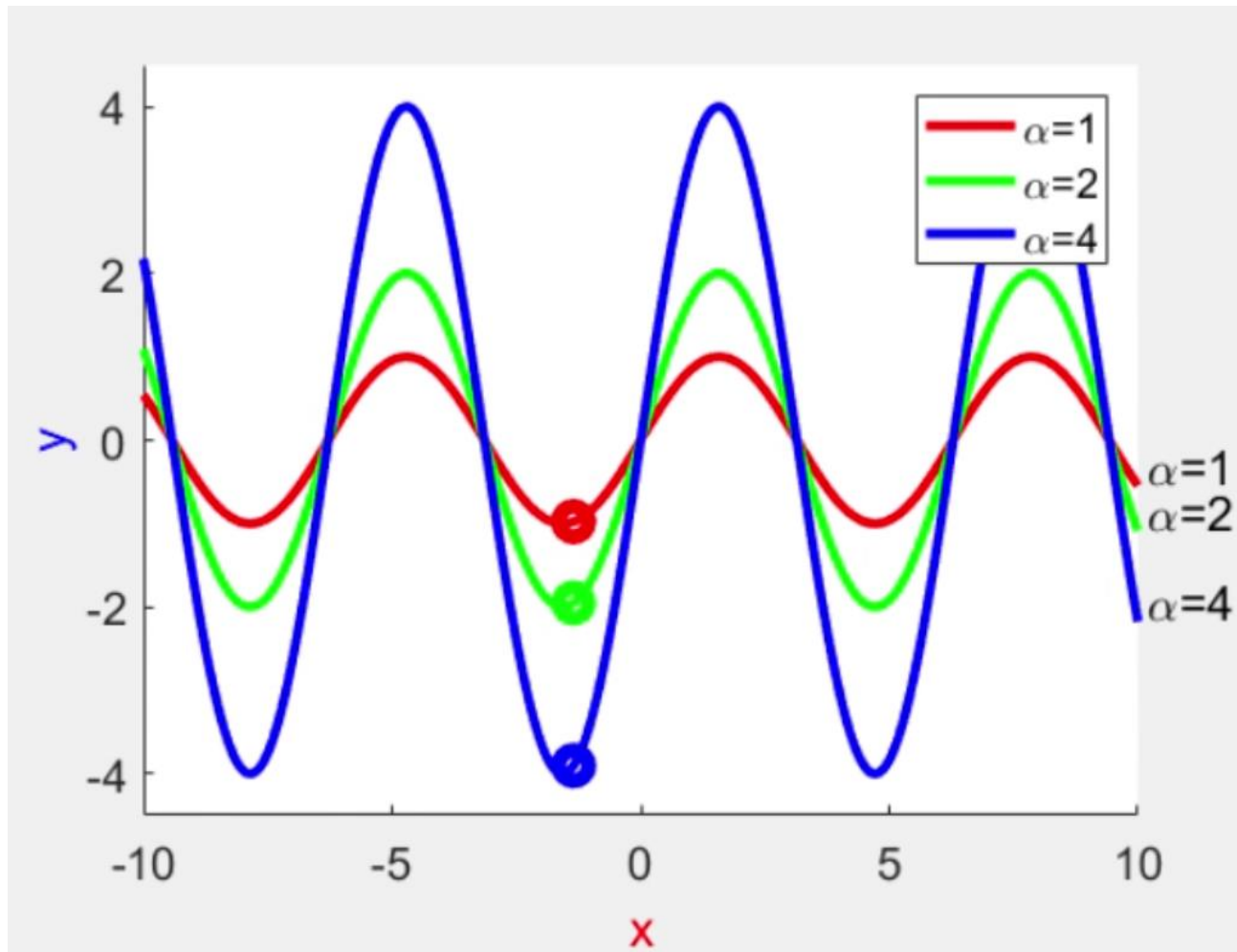
        plot(x,y(i,:), 'LineWidth', 3,...
            'color', color(i));

        plot(xpoint, y0, 'o', ...
            'MarkerSize', 10, ...
            'LineWidth', 5,...
            'color', color(i));
        i = i + 1;
    end
    pause(0.033);
end
hold off
```

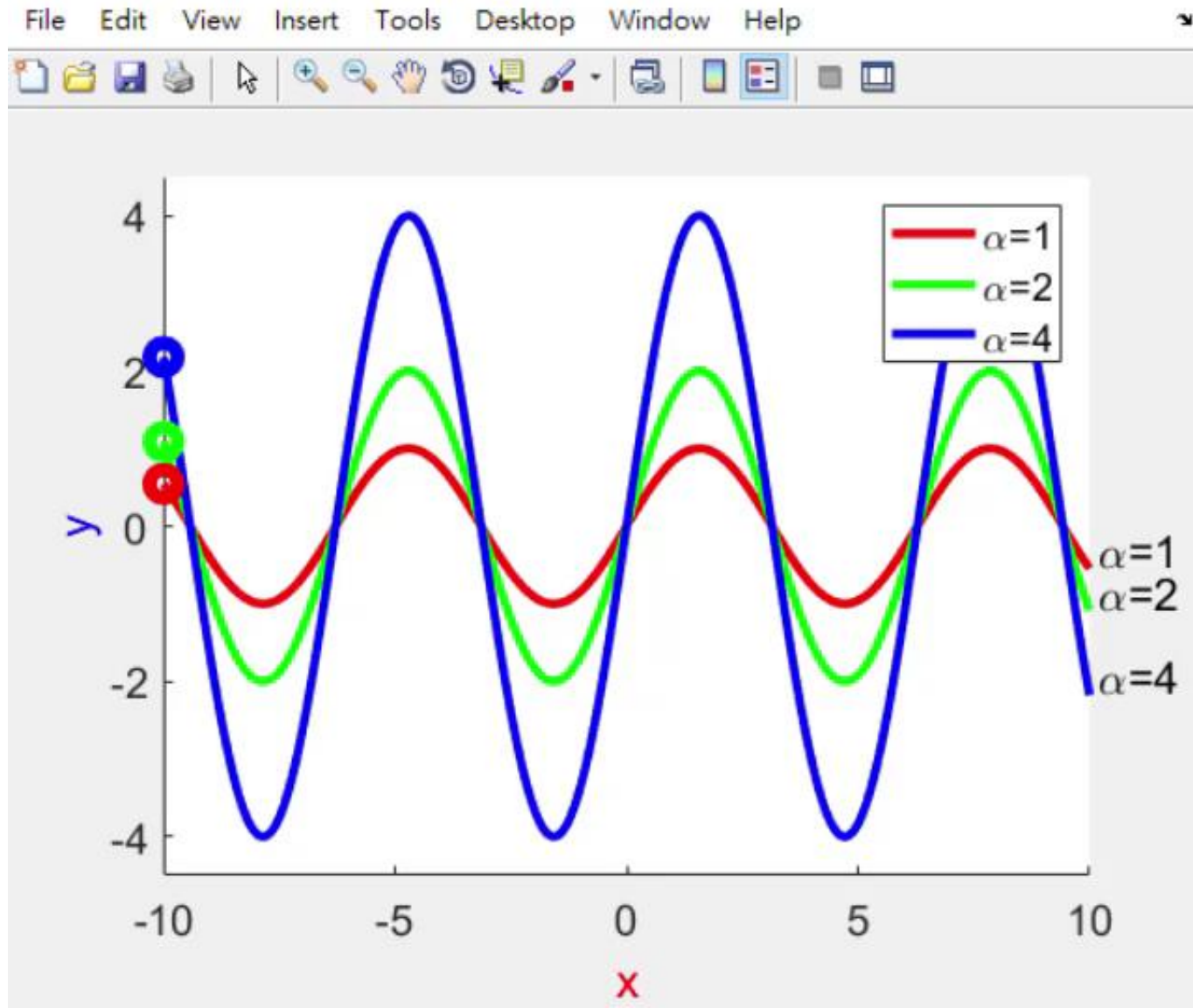


# Point animation. With legends...

Draw the three curves. Trace a point of each curve.



# Demo. Point animation. With legends, axis labels,...



# Point Animation. With legends, labels...

```
close all; clf; clear; hold on;
msg = []; i = 1; A = [ 1 2 4];
color= ['r' 'g' 'b'];
set(gca,'FontSize',15);
axis([ -10 10 -4.5 4.5]);
x = [-10:0.05:10]; sx = size(x);
y = zeros(3, sx(2));
for a = A
    y(i,:) = a.*sin(x);
    s = sprintf('\\alpha=%d',a);
    i = i + 1;
    msg = [msg;s];
end
hold on
```

```
for xpoint = [-10:0.05:10]
    i = 1;
    for a = A        % First, draw all curves
        plot(x,y(i,:), 'LineWidth', 3, 'color', ...,
            color(i));
        i = i + 1;
    end
    i = 1;
    for a = A        % Second, draw all markers
        y0 = a.*sin(xpoint);
        plot(xpoint, y0, 'o', ...
            'MarkerSize', 10, ...
            'LineWidth', 5,...
            'color', color(i));
        i = i + 1;
    end
    legend(msg); % Third, draw the legend
    for i = [1:3] % Forth, display labels
        %text(pi/2, A(i)+0.25, msg(i,:),
        'FontSize',16)
        text(x(end)+0.2, y(i,end)+0.25,
```





# Point Animation. With legends, labels...

```
close all; clf; clear; hold on;
msg = []; i = 1; A = [ 1 2 4];
color= ['r' 'g' 'b'];
set(gca,'FontSize',15);
axis([ -10 10 -4.5 4.5]);
x = [-10:0.05:10]; sx = size(x);
y = zeros(3, sx(2));
for a = A
    y(i,:) = a.*sin(x);
    s = sprintf('\\alpha=%d',a);
    i = i + 1;
    msg = [msg;s];
end
hold on
```

```
for xpoint = [-10:0.05:10]
    i = 1;
    for a = A        % First, draw all curves
        plot(x,y(i,:), 'LineWidth', 3, 'color', ...,
            color(i));
        i = i + 1;
    end
    i = 1;
    for a = A        % Second, draw all markers
        y0 = a.*sin(xpoint);
        plot(xpoint, y0, 'o', ...
            'MarkerSize', 10, ...
            'LineWidth', 5,...
            'color', color(i));
        i = i + 1;
    end
    legend(msg); % Third, draw the legend
    for i = [1:3] % Forth, display labels
        %text(pi/2, A(i)+0.25, msg(i,:),
        'FontSize',16)
        text(x(end)+0.2, y(i,end)+0.25,
```



# Point Animation. With legends, labels...

```
for xpoint = [-10:0.05:10]
    i = 1;
    for a = A
        plot(x,y(i,:), 'LineWidth', 3, ...
            'color', color(i));
        i = i + 1;
    end
    i = 1;
    for a = A
        y0 = a.*sin(xpoint);
        plot(xpoint, y0, 'o', ...
            'MarkerSize', 10, ...
            'LineWidth', 5,...
            'color', color(i));
        i = i + 1;
    end

    legend(msg);
```

```
for i = [1:3]
    text(x(end)+0.2, y(i,end)+0.25, ...
        msg(i,:), 'FontSize',16)
end

xlabel('x', 'color', 'red');
ylabel('y', 'color', 'blue');

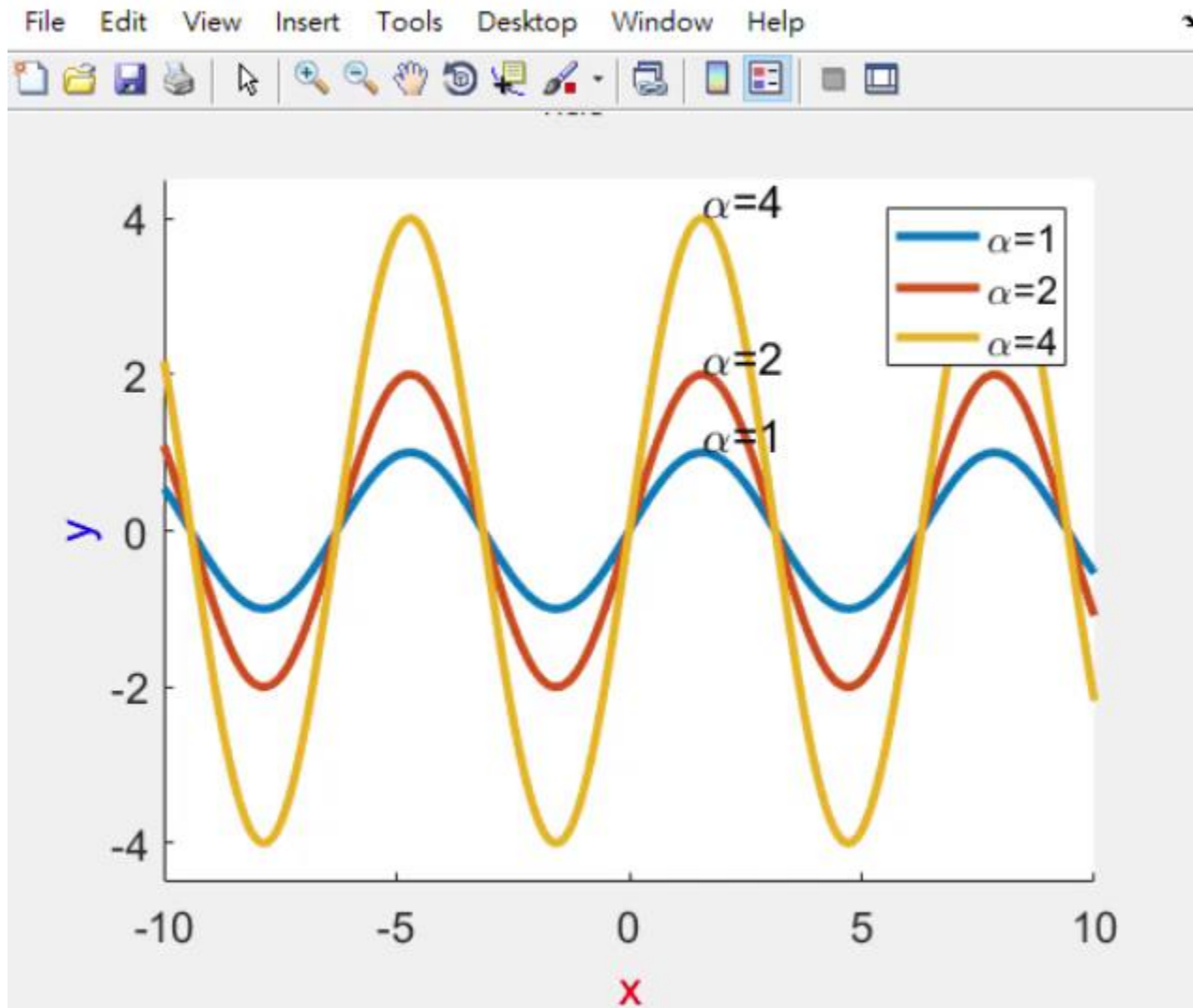
if (xpoint ~= -10)
    pause(0.033); clf; hold on;
    set(gca,'FontSize',15);
    axis([ -10 10 -4.5 4.5]);
end
if xpoint == -10 ...
    input('Press Enter to start...'); end

end
hold off
```



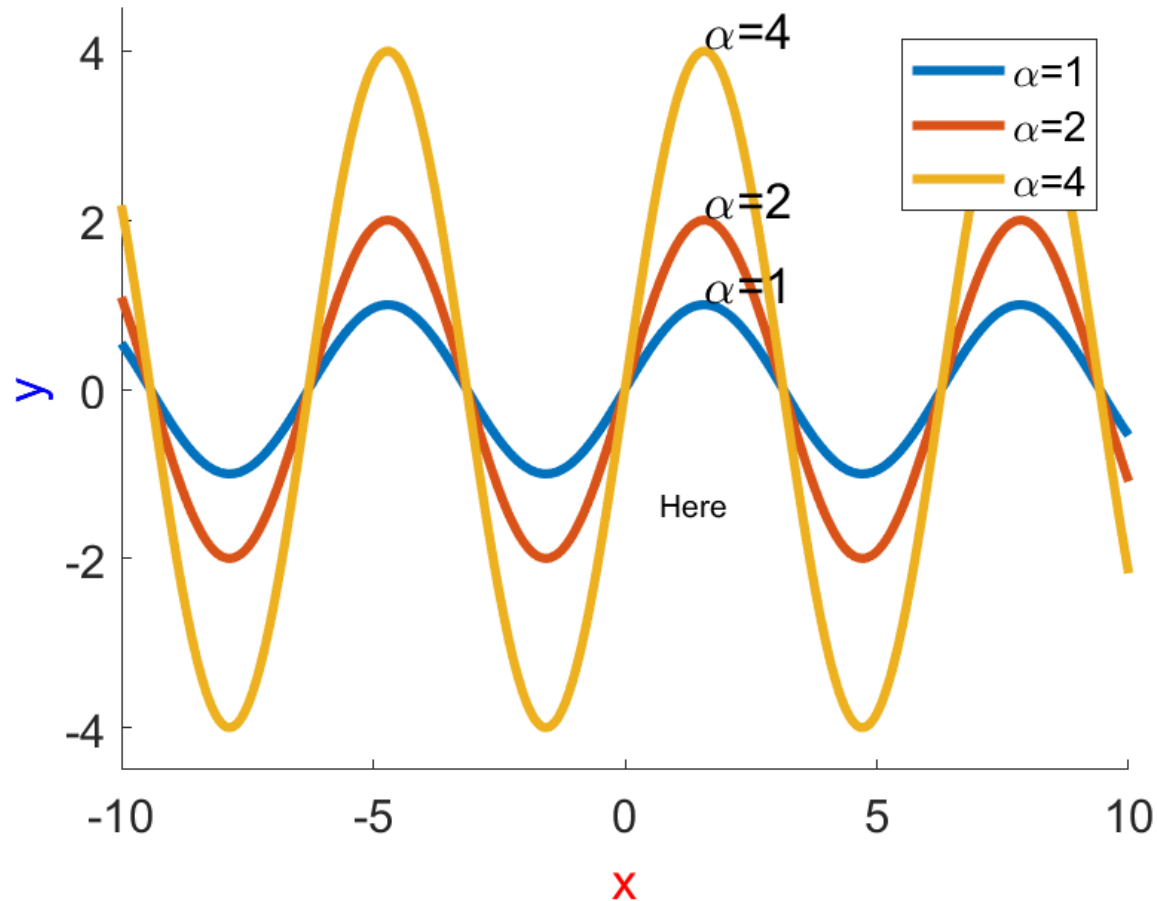
# Demo

gtext: Place text with mouse



# saveas

`saveas(gcf,'sin.png')`



# saveas : file format

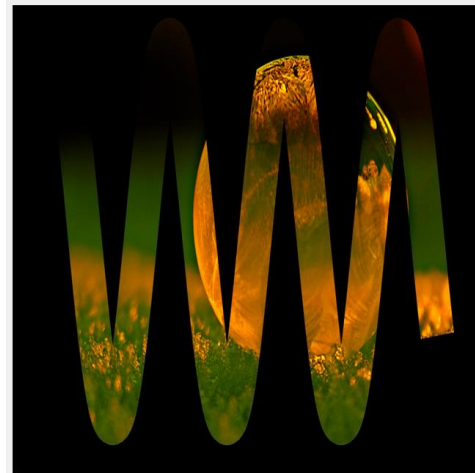
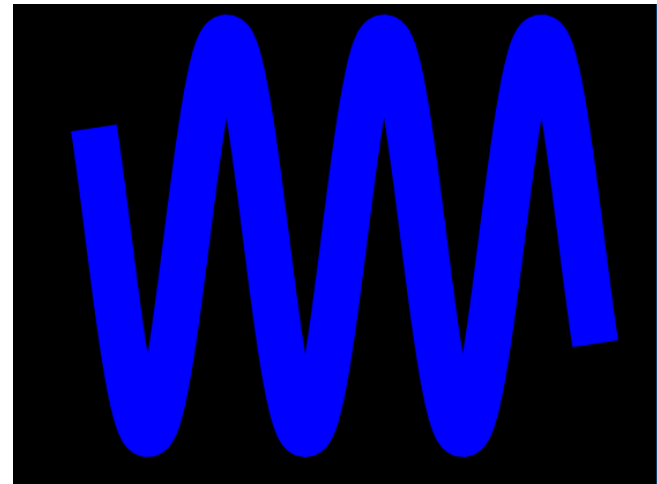
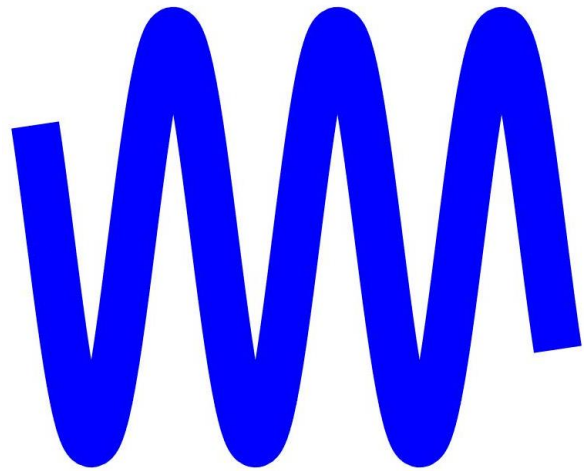
Extension	Resulting Format
.fig	MATLAB <sup>®</sup> FIG-file (invalid for Simulink block diagrams)
.m	MATLAB FIG-file and MATLAB code that opens figure (invalid for Simulink block diagrams)
.jpg	JPEG image
.png	Portable Network Graphics
.eps	EPS Level 3 Black and White
.pdf	Portable Document Format
.bmp	Windows <sup>®</sup> bitmap
.emf	Enhanced metafile
.pbm	Portable bitmap
.pcx	Paintbrush 24-bit
.pgm	Portable Graymap
.ppm	Portable Pixmap
.tif	TIFF image, compressed

<https://nl.mathworks.com/help/matlab/ref/saveas.html>



# Exercise

## curves + image processing

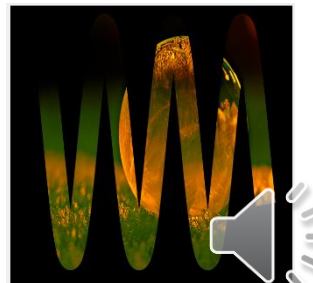
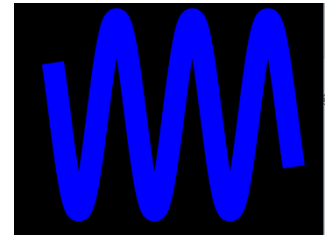
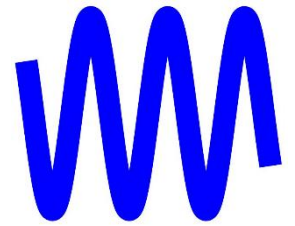


```
close all; clear; clf
x= -10:0.05:10; y = sin(x);
figure, plot(x, y,...
    'linewidth', 30, 'color', 'b');
set(gcf, 'color', 'k');
set(gca, 'visible', 'off'); %set axes off
saveas(gcf, 'sin_01.jpg');

I = imread('sin_01.jpg');
figure, imshow(I);

I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;

L = imread('tmp.png');
figure, imshow(L)
L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```

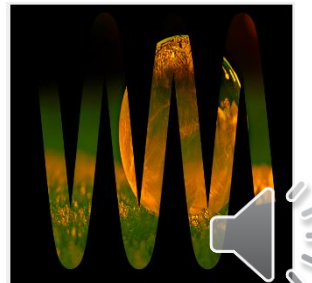
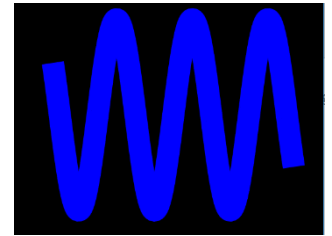
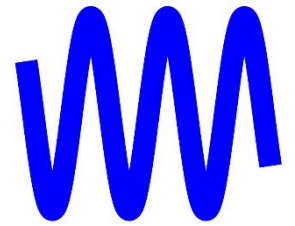


```
close all; clear; clf
x= -10:0.05:10; y = sin(x);
figure, plot(x, y,...
    'linewidth', 30, 'color', 'b');
set(gcf, 'color', 'k');
set(gca, 'visible', 'off'); %set axes off
saveas(gcf, 'sin_01.jpg');

I = imread('sin_01.jpg');
figure, imshow(I);

I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;

L = imread('tmp.png');
figure, imshow(L)
L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```



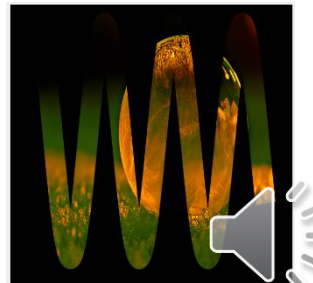
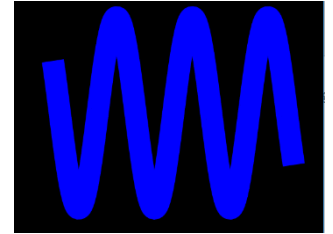
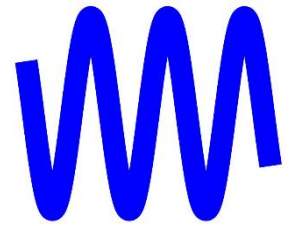


```
close all; clear; clf
x= -10:0.05:10; y = sin(x);
figure, plot(x, y,...
    'linewidth', 30, 'color', 'b');
set(gcf, 'color', 'k');           %black
set(gca, 'visible', 'off');      %set axes off
saveas(gcf, 'sin_01.jpg');
```

```
I = imread('sin_01.jpg');
figure, imshow(I);
```

```
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;
```

```
L = imread('tmp.png');
figure, imshow(L)
L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```

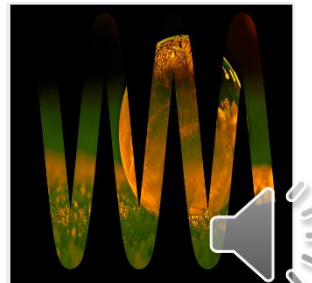
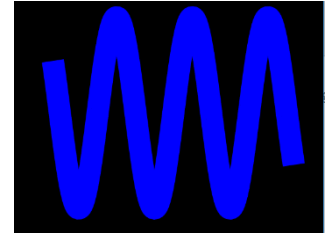
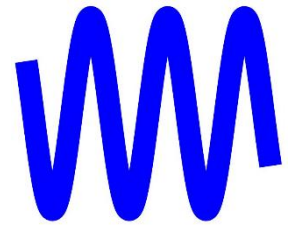
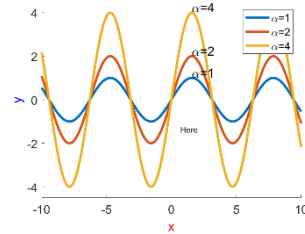


```
close all; clear; clf
x= -10:0.05:10; y = sin(x);
figure, plot(x, y,...
    'linewidth', 30, 'color', 'b');
set(gcf, 'color', 'k'); %black
set(gca, 'visible', 'off'); %set axes off
saveas(gcf, 'sin_01.jpg');
```

```
I = imread('sin_01.jpg');
figure, imshow(I);
```

```
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;
```

```
L = imread('tmp.png');
figure, imshow(L)
L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```

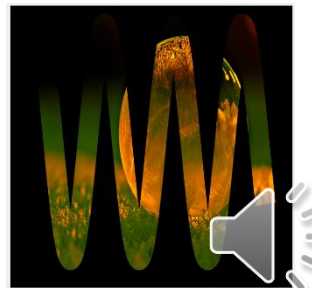
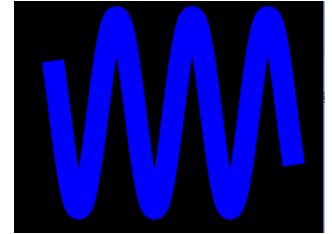
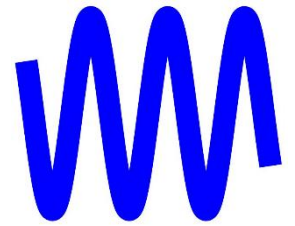


```
close all; clear; clf
x= -10:0.05:10; y = sin(x);
figure, plot(x, y,...
    'linewidth', 30, 'color', 'b');
set(gcf, 'color', 'k');
set(gca, 'visible', 'off'); %set axes off
saveas(gcf, 'sin_01.jpg');
```

```
I = imread('sin_01.jpg');
figure, imshow(I);
```

```
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;
```

```
L = imread('tmp.png');
figure, imshow(L)
L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```



```

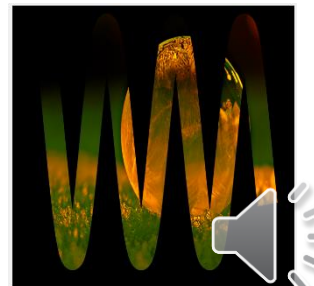
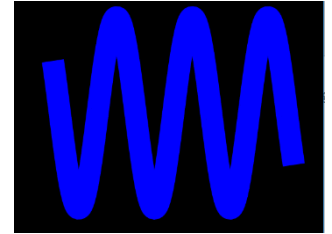
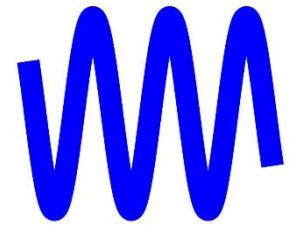
close all; clear; clf
x= -10:0.05:10; y = sin(x);
figure, plot(x, y,...
    'linewidth', 30, 'color', 'b');
set(gcf, 'color', 'k');
set(gca, 'visible', 'off'); %set axes off
saveas(gcf, 'sin_01.jpg');

I = imread('sin_01.jpg');
figure, imshow(I);

I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;           % 1-(0,0,1) = (1,1,0)

L = imread('tmp.png');
figure, imshow(L)
L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);

```

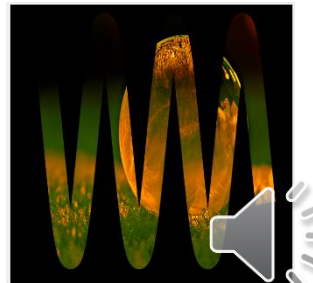
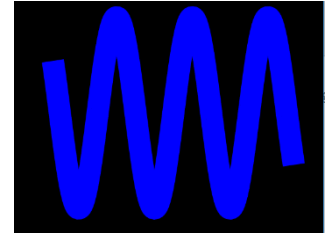
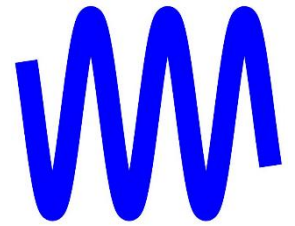


```
close all; clear; clf
x= -10:0.05:10; y = sin(x);
figure, plot(x, y,...
    'linewidth', 30, 'color', 'b');
set(gcf, 'color', 'k');
set(gca, 'visible', 'off'); %set axes off
saveas(gcf, 'sin_01.jpg');
```

```
I = imread('sin_01.jpg');
figure, imshow(I);
```

```
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;
```

```
L = imread('tmp.png');
figure, imshow(L)
L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```

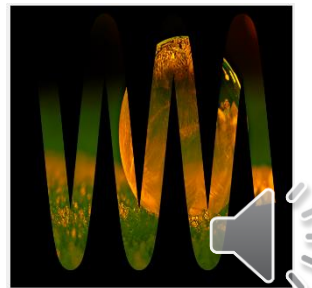
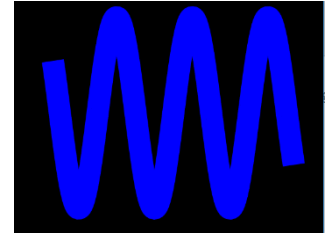
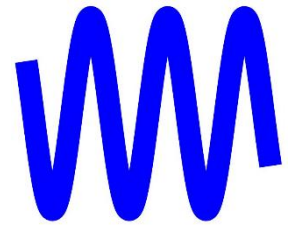


```
close all; clear; clf
x= -10:0.05:10; y = sin(x);
figure, plot(x, y,...
    'linewidth', 30, 'color', 'b');
set(gcf, 'color', 'k');
set(gca, 'visible', 'off'); %set axes off
saveas(gcf, 'sin_01.jpg');

I = imread('sin_01.jpg');
figure, imshow(I);

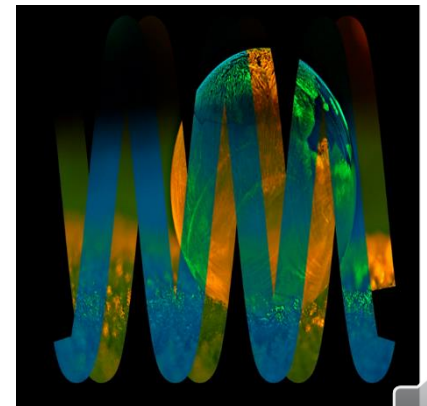
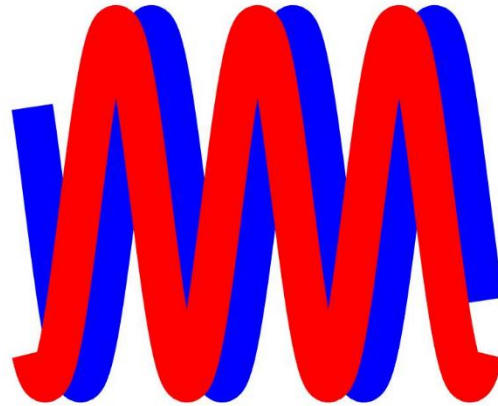
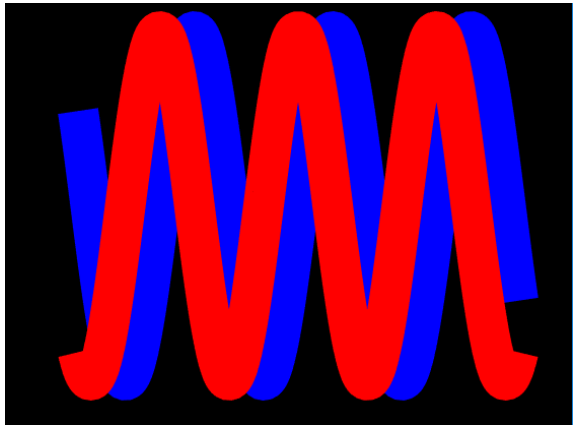
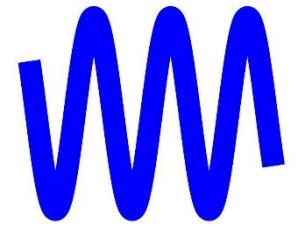
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;

L = imread('tmp.png');
figure, imshow(L)
L1 = im2double(L); % intensity between 0 and 1
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```



# Exercise

## curves + image processing



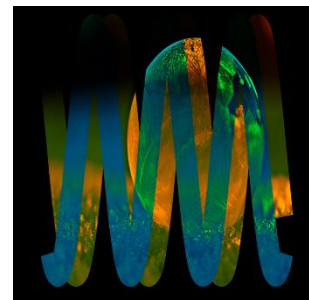
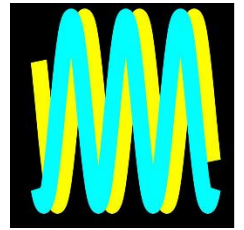
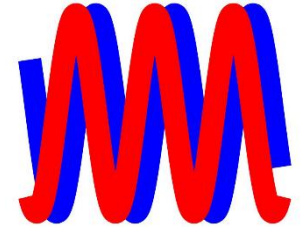
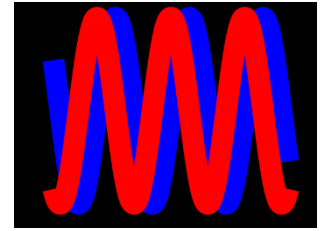


```
close all; clear; clf
x= -10:0.05:10; y = sin(x); y1 = cos(x);
figure
plot(x, y,'linewidth', 30, 'color', 'b');
hold on
plot(x, y1,'linewidth', 30, 'color', 'r');
set(gcf,'color','k'); set(gca,'visible','off');
saveas(gcf,'sin_01.jpg');

I = imread('sin_01.jpg');
figure, imshow(I);

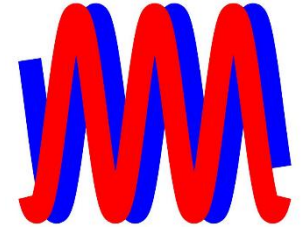
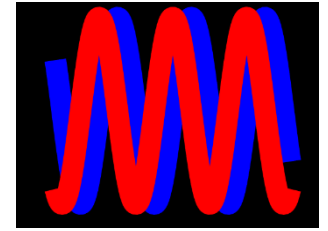
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;

L = imread('tmp.png');
figure, imshow(L); L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```

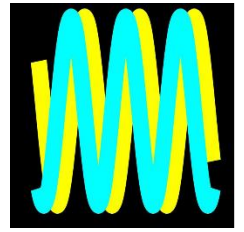




```
close all; clear; clf
x= -10:0.05:10; y = sin(x); y1 = cos(x);
figure
plot(x, y,'linewidth', 30, 'color', 'b');
hold on
plot(x, y1,'linewidth', 30, 'color', 'r');
set(gcf,'color','k'); set(gca,'visible','off');
saveas(gcf,'sin_01.jpg');
```



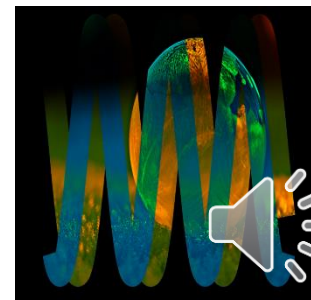
```
I = imread('sin_01.jpg');
figure, imshow(I);
```



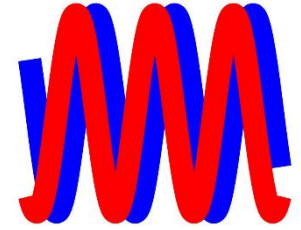
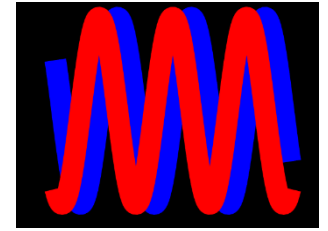
```
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;
```



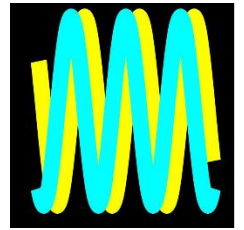
```
L = imread('tmp.png');
figure, imshow(L); L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```



```
close all; clear; clf
x= -10:0.05:10; y = sin(x); y1 = cos(x);
figure
plot(x, y,'linewidth', 30, 'color', 'b');
hold on
plot(x, y1,'linewidth', 30, 'color', 'r');
set(gcf,'color','k'); set(gca,'visible','off');
saveas(gcf,'sin_01.jpg');
```



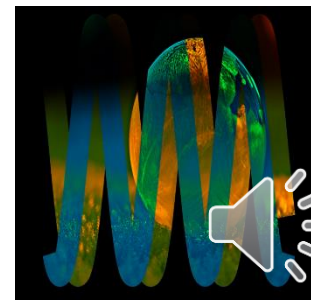
```
I = imread('sin_01.jpg');
figure, imshow(I);
```



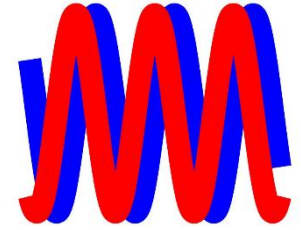
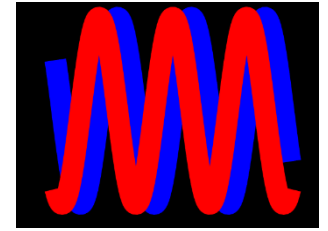
```
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;
```



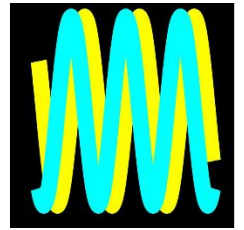
```
L = imread('tmp.png');
figure, imshow(L); L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```



```
close all; clear; clf
x= -10:0.05:10; y = sin(x); y1 = cos(x);
figure
plot(x, y,'linewidth', 30, 'color', 'b');
hold on
plot(x, y1,'linewidth', 30, 'color', 'r');
set(gcf,'color','k'); set(gca,'visible','off');
saveas(gcf,'sin_01.jpg');
```



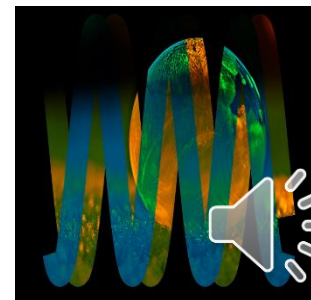
```
I = imread('sin_01.jpg');
figure, imshow(I);
```



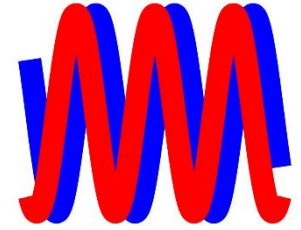
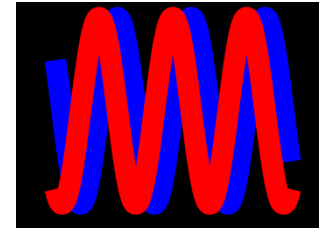
```
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;
```



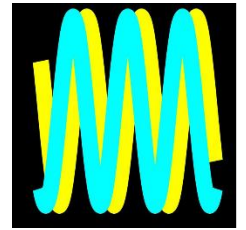
```
L = imread('tmp.png');
figure, imshow(L); L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```



```
close all; clear; clf
x= -10:0.05:10; y = sin(x); y1 = cos(x);
figure
plot(x, y,'linewidth', 30, 'color', 'b');
hold on
plot(x, y1,'linewidth', 30, 'color', 'r');
set(gcf,'color','k'); set(gca,'visible','off');
saveas(gcf,'sin_01.jpg');
```



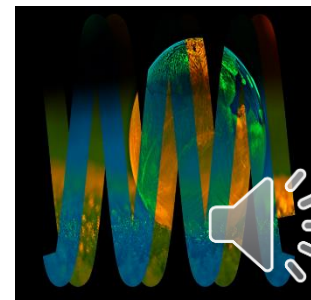
```
I = imread('sin_01.jpg');
figure, imshow(I);
```



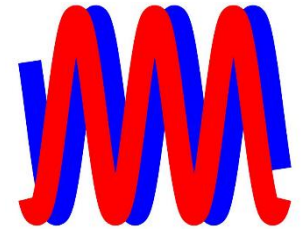
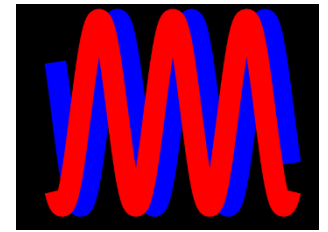
```
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;    % 1 - (1,0,0) = (0,1,1) = cyan
              % 1 - (0,0,1) = (1,1,0) = yellow
```



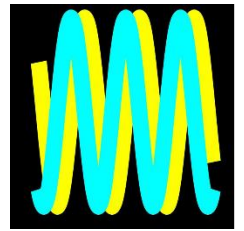
```
L = imread('tmp.png');
figure, imshow(L); L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```



```
close all; clear; clf
x= -10:0.05:10; y = sin(x); y1 = cos(x);
figure
plot(x, y,'linewidth', 30, 'color', 'b');
hold on
plot(x, y1,'linewidth', 30, 'color', 'r');
set(gcf,'color','k'); set(gca,'visible','off');
saveas(gcf,'sin_01.jpg');
```



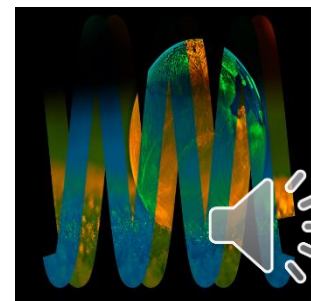
```
I = imread('sin_01.jpg');
figure, imshow(I);
```



```
I2 = I; K = I2; K = im2double(K);
K = imresize(K, [640 640]);
K1 = 1-K;
```



```
L = imread('tmp.png');
figure, imshow(L); L1 = im2double(L);
L1 = imresize(L1, [640 640]);
figure
imshow(L1.*K1);
```



# Exercises

help sprintf

help text

help saveas

See also xlabel, ylabel, zlabel, title, gtext, line, patch.



# Exercise

## legend

Plot a set of curves:  $y_i$

Mark each curve with the parameter values.  
show the legend of each curve. Set the linewidth to 3. Set the title.

Beautify the figure.

**How do we know that the best position to mark each curve?**



# Exercise

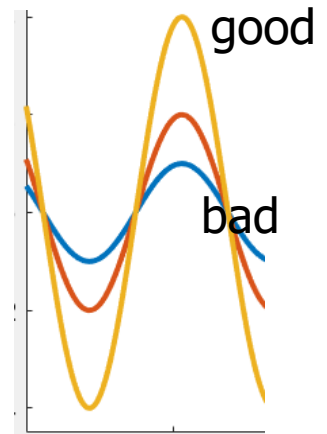
Plot a set of curves:  $y_i(x)$ ,  $i = 1, 2, \dots, n$

Mark each curve with the parameter values. show the legend of each curve. Set the linewidth to 3. Set the title.

Beautify the figure.

Major steps:

1. Draw all the figures
2. Construct legends. Set the legends.
3. Construct title. Set the title.
4. Find the best position to mark each curve. Avoid occlusion. Mark the curve at empty space.
5. Beautify the figure.





# Exercises

- In one figure, plot

$$y1 = x^2 - 2x + 3\sin \mu, \text{ and}$$

$$y2 = -x^3 + x^2 + 1/(x-1)$$

for  $x = -10$  to  $x = 10$ .  $\mu = (1-x^2)/x$ .

1. Make  $y1$  a solid line with blue circle points
2. Make  $y2$  a dashed line with black x-mark points.
3. Add a legend with both curves listed in the figure.
4. Add a grid.
5. Beautify the figure.