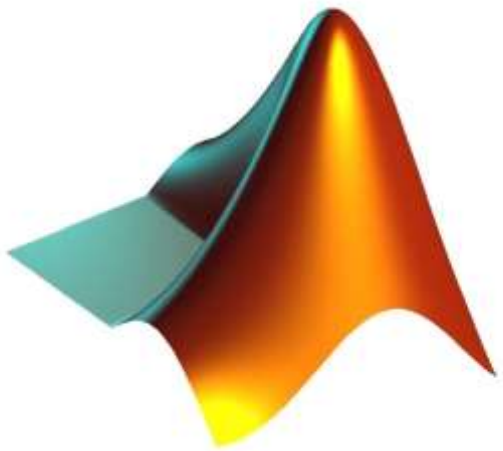
APPLICATIONS OF MATLAB IN ENGINEERING

Yan-Fu Kuo
Dept. of Bio-industrial Mechatronics Engineering
National Taiwan University

Fall 2015

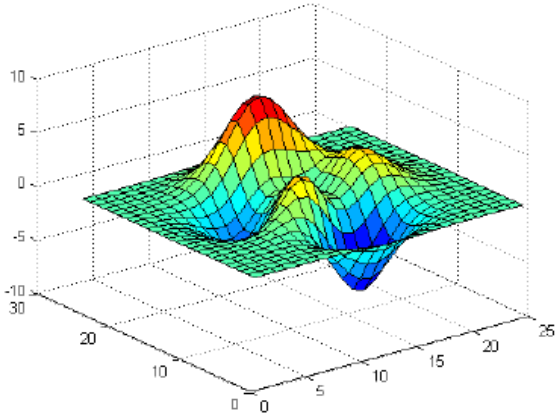
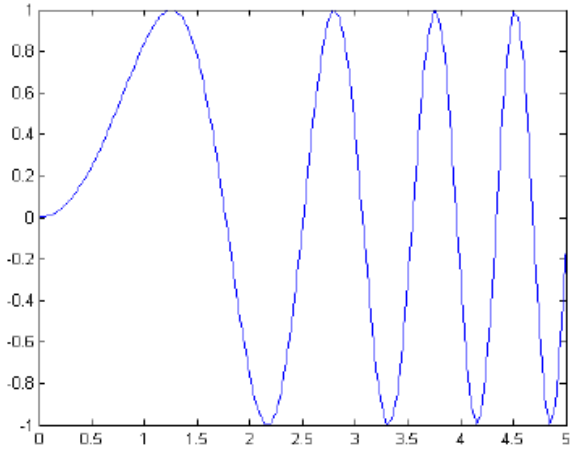
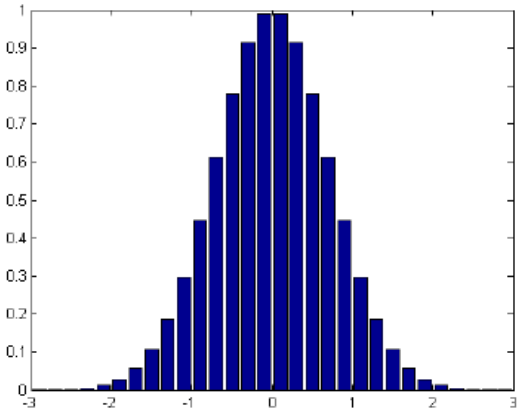
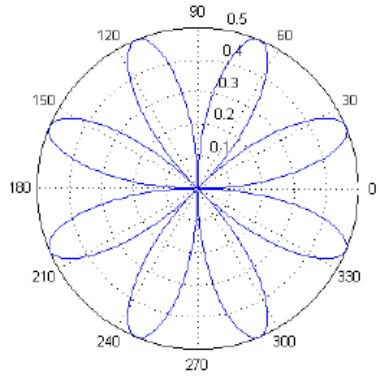
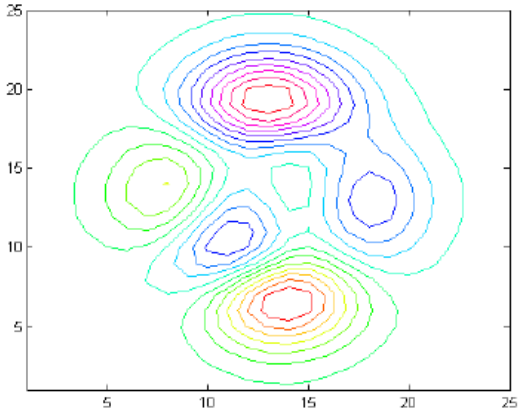
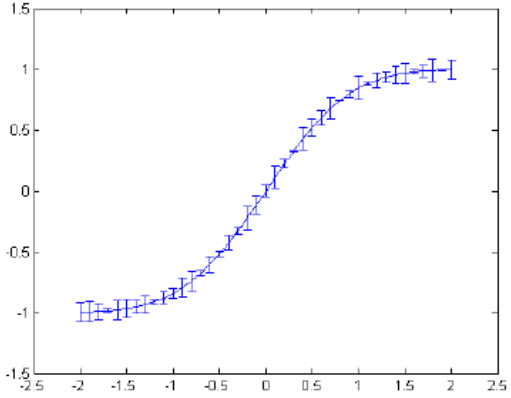
Today:

- Basic plotting
- Graphical object properties



Basics

- MATLAB has a powerful plotting engine that can generate a wide variety of plots

Plot from "Data"

- MATLAB does not understand functions

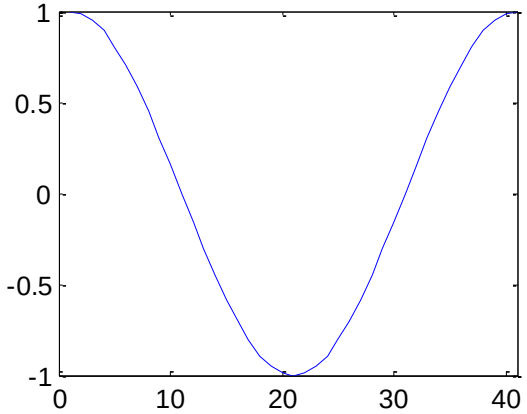
$$f(t) = \sin(2\pi t)$$

- Strategies:
  1. Generate the numeric values of a function over a specific range
  2. Display the data "points" in a graphical way

plot()

- `plot(x,y)` plots each vector pairs (`x,y`)
- `plot(y)` plots each vector pairs (`x,y`), where `x=[1…n], n=length(y)`
- Example:

```
plot(cos(0:pi/20:2*pi));
```



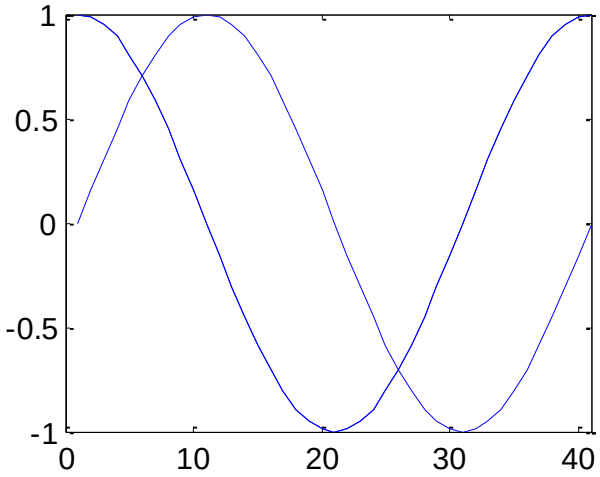
- What do you see after this script?

```
plot(cos(0:pi/20:2*pi));
plot(sin(0:pi/20:2*pi));
```

hold on/off

- Use `hold on` to have both plots in one figure

```
hold on
plot(cos(0:pi/20:2*pi));
plot(sin(0:pi/20:2*pi));
hold off
```



Plot Style

- `plot(x,y,'str')` plots each vector pairs (`x,y`) using the format defined in `str` (check linespec)

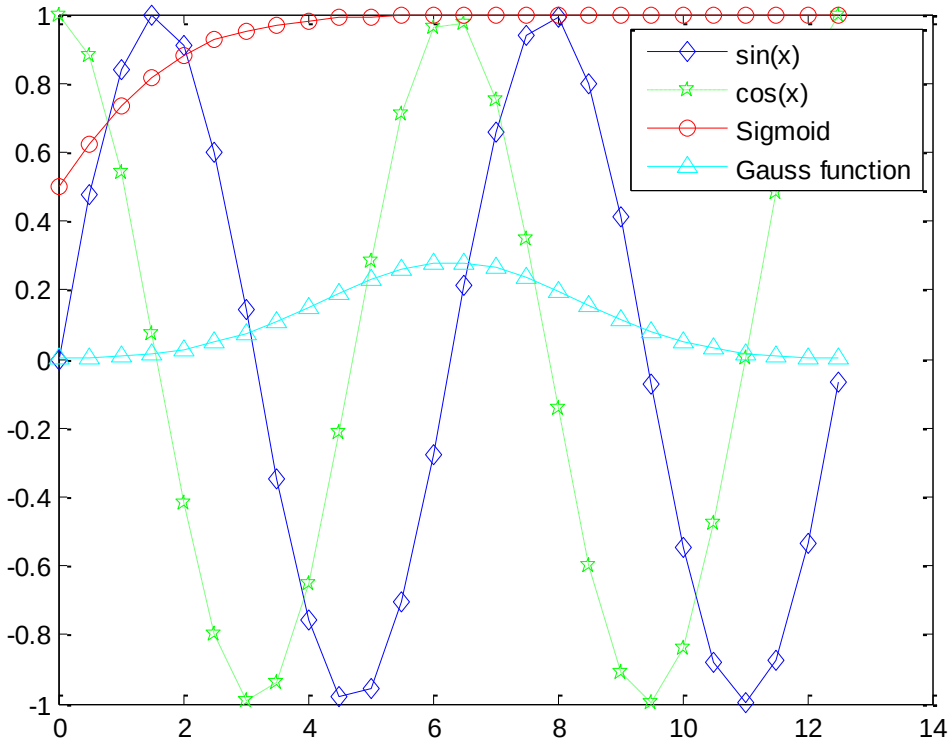
Data markers		Line types		Colors	
Dot (.)	`.`	Solid line	`-`	Black	`k`
Asterisk (*)	`*`	Dashed line	`--`	Blue	`b`
Cross (×)	`x`	Dash-dotted line	`-.`	Cyan	`c`
Circle (○)	`o`	Dotted line	`:`	Green	`g`
Plus sign (+)	`+`			Magenta	`m`
Square (□)	`s`			Red	`r`
Diamond (◇)	`d`			White	`w`
Five-pointed star (☆)	`p`			Yellow	`y`
Triangle (down ∇)	`v`				
Triangle (up Δ)	`^`				
Triangle (left ▷)	`<`				
Triangle (right ◁)	`>`				
hexagram	`H`				

legend()

- Add legend to graph

```
legend('L1',...)
```

- Position adjustment

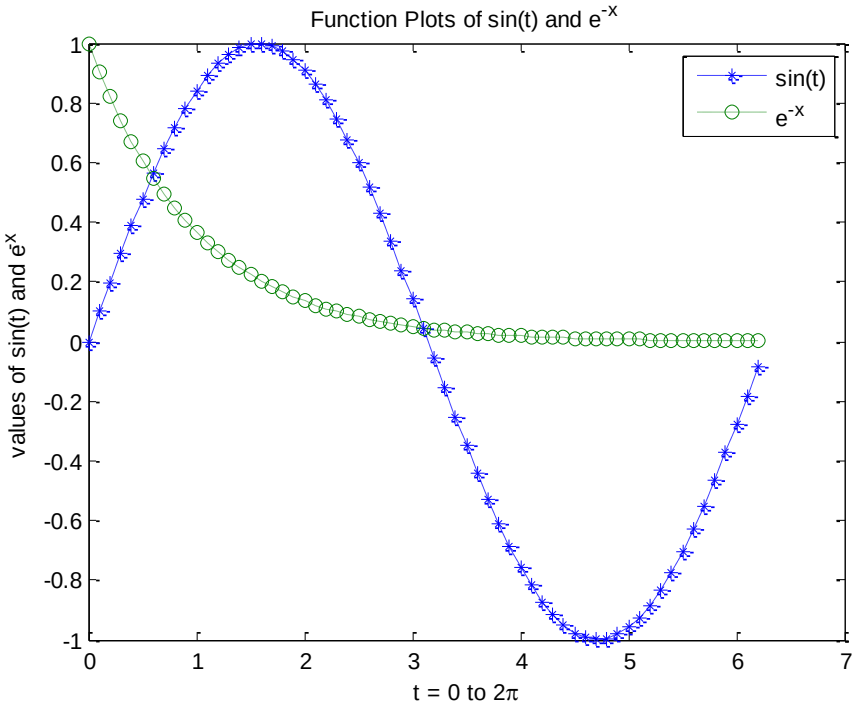



```
x=0:0.5:4*pi;
y=sin(x);  h=cos(x);  w=1./(1+exp(-x));
g=(1/(2*pi*2)^0.5).*exp((-1.*(x-2*pi).^2)./(2*2^2));
plot(x,y,'bd-',x,h,'gp:',x,w,'ro-',x,g,'c^-');
```

```
legend('sin(x)','cos(x)','Sigmoid','Gauss function');
```

title() and ?label()

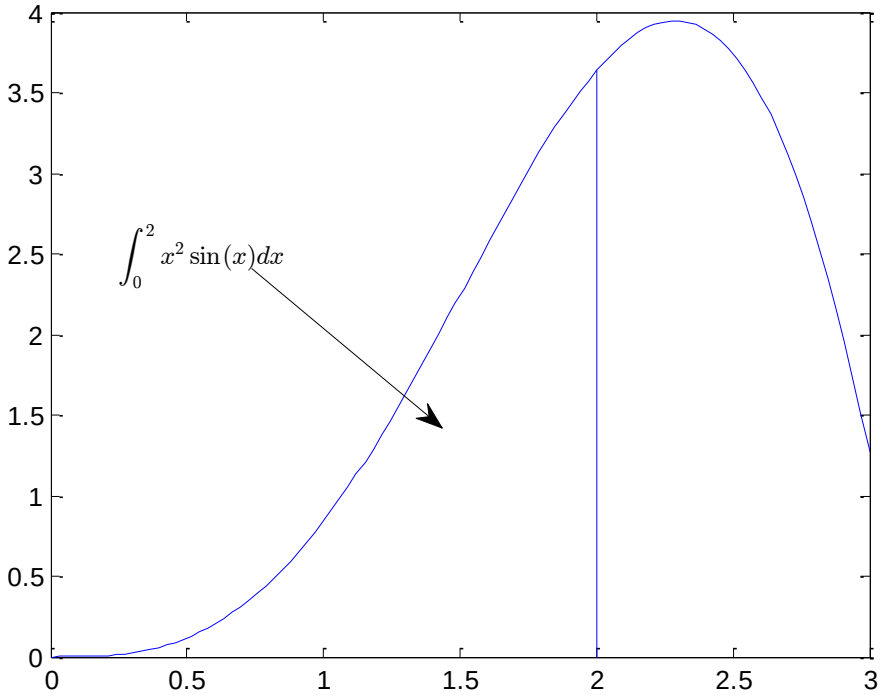
- title()
- xlabel()
- ylabel()
- zlabel()




```
x = 0:0.1:2*pi; y1 = sin(x); y2 = exp(-x);
plot(x, y1, '--*', x, y2, ':o');
xlabel('t = 0 to 2\pi');
ylabel('values of sin(t) and e^{-x}')
title('Function Plots of sin(t) and e^{-x}');
legend('sin(t)','e^{-x}');
```

text() and annotation()

- Text with mathematical expression using LaTex




```
x = linspace(0,3); y = x.^2.*sin(x); plot(x,y);
line([2,2],[0,2^2*sin(2)]);
str = '$$ \int_{0}^{2} x^2\sin(x) dx $$';
text(0.25,2.5,str,'Interpreter','latex');
annotation('arrow','X',[0.32,0.5],'Y',[0.6,0.4]);
```

Exercise

- Plot f as a black line and g as a series of red circles for the range t = 1 to 2 in one figure

$$f = t^2 \quad \text{and} \quad g = \sin(2\pi t)$$

- Label each axis, and add title and legend

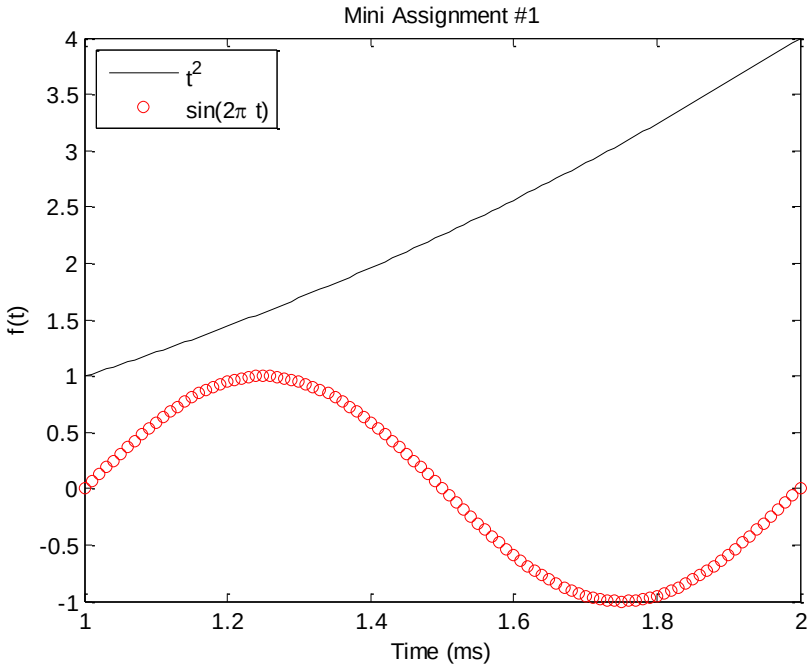
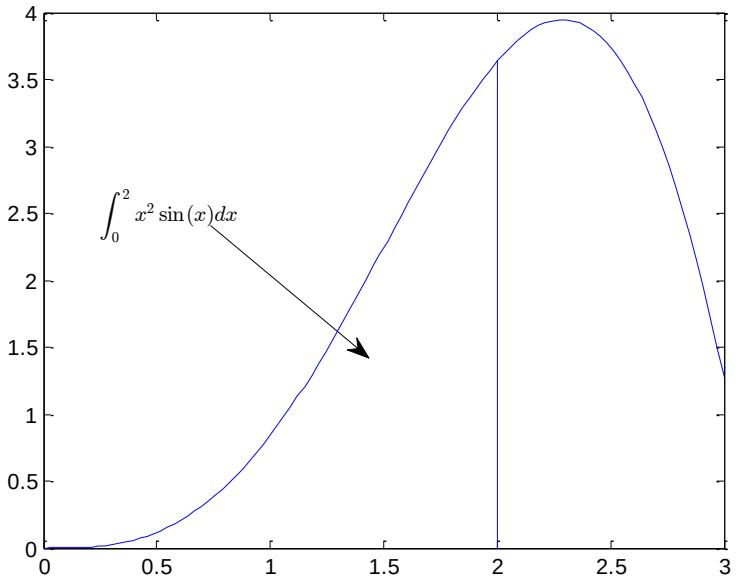
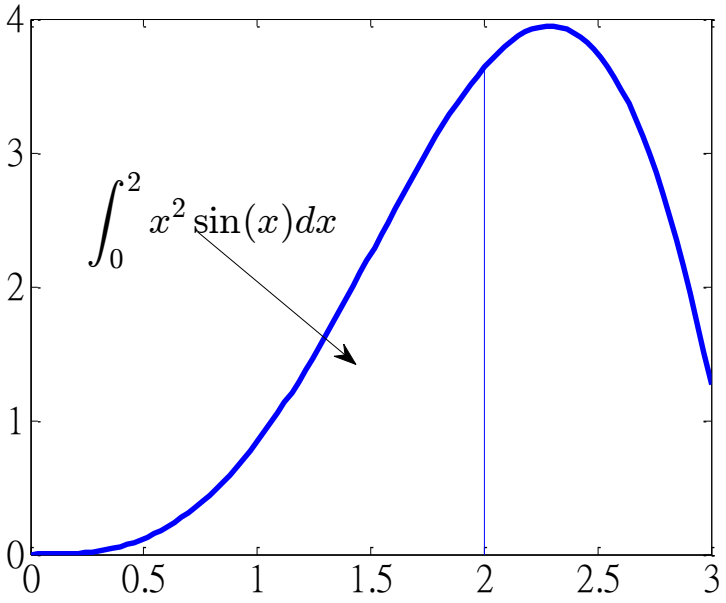


Figure Adjustment

- Several properties:
  - Font
  - Font size
  - Line width
  - Axis limit
  - Tick position
  - Tick label
- But how?





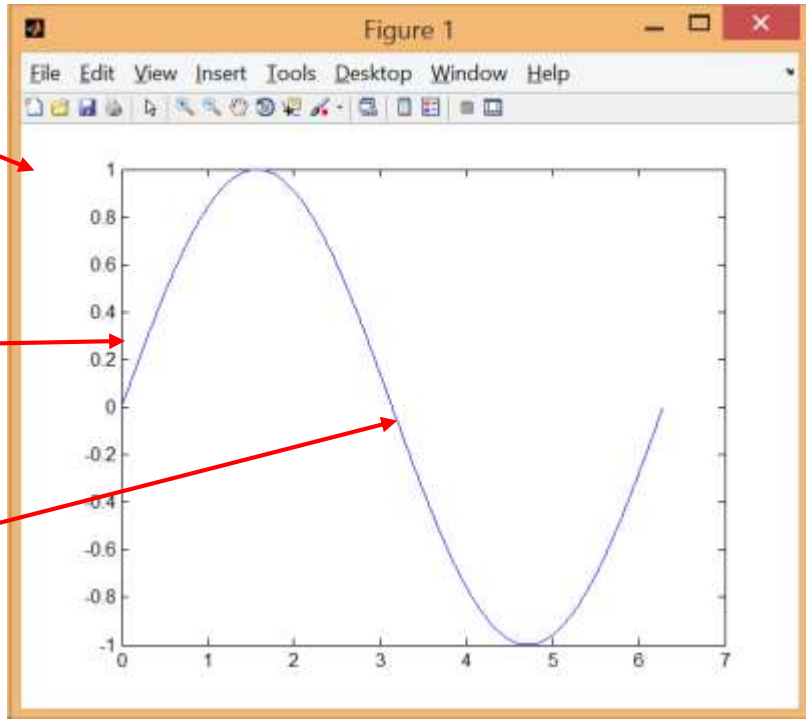
Graphical Objects

- A figure is composed of many objects

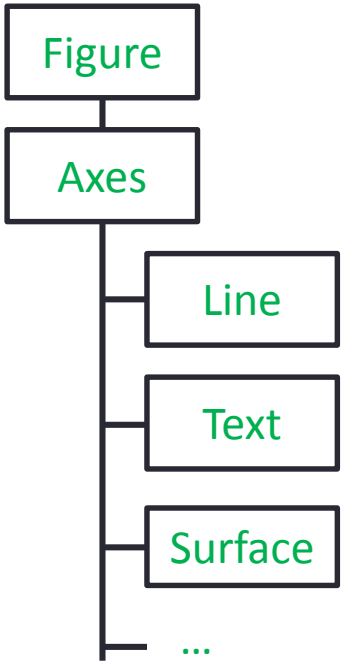
Figure object

Axes object

Line object

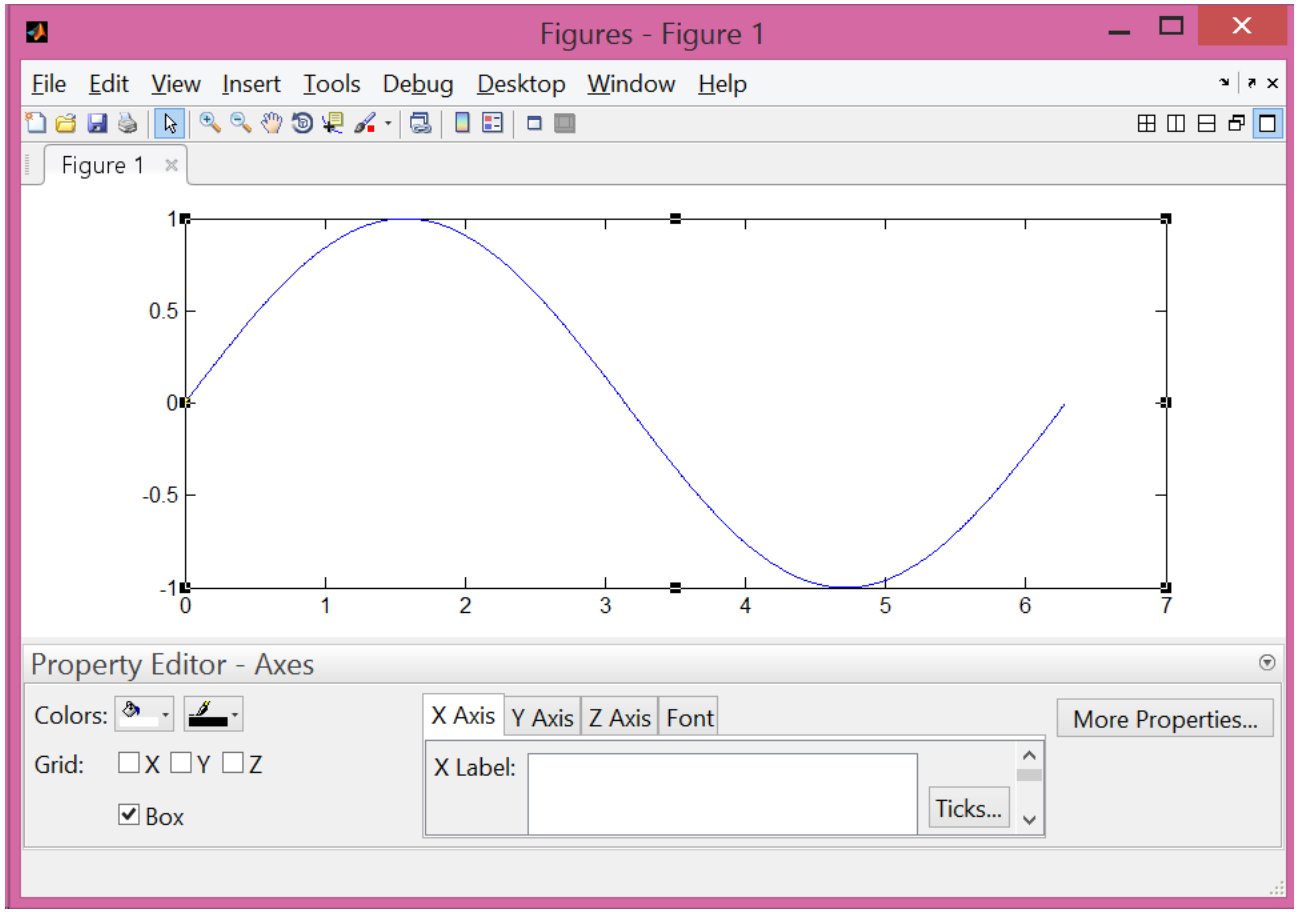


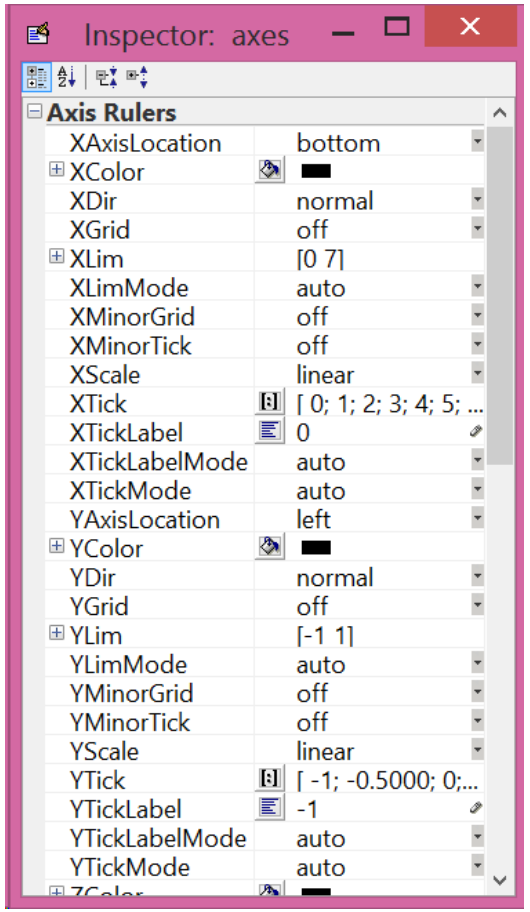
Hierarchy




```
x = linspace(0, 2*pi, 1000);  y = sin(x);
plot(x,y);  set(gcf, 'Color', [1 1 1]);
```

Figure Properties



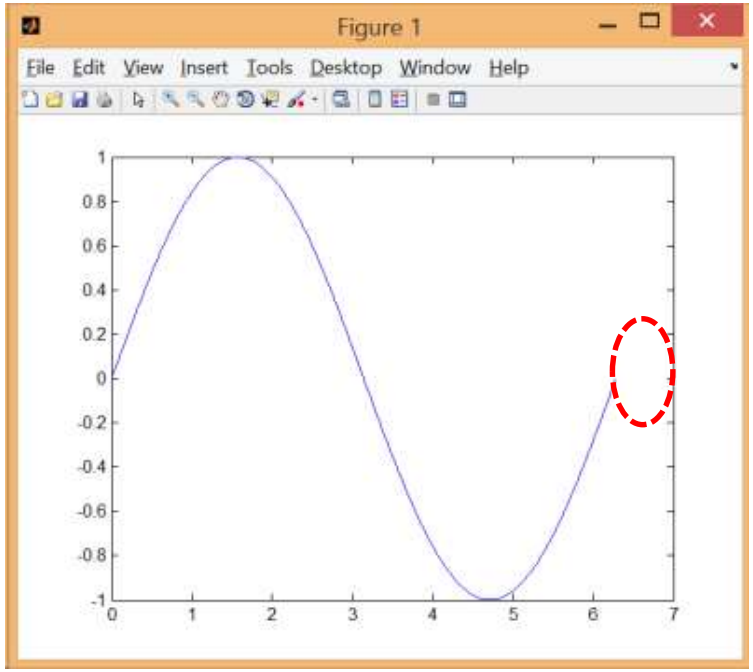


- http://www.mathworks.com/help/matlab/ref/figure-properties.html

Modifying Properties of An Object

- Strategy:
  1. Identify the “handle” of an object
  2. Fetch or modify the object’s properties

- For example, to change the limits of the x-axis:
  1. Find the handle of the x-axis
  2. Modify the limits



1. Identifying the Handle of An Object

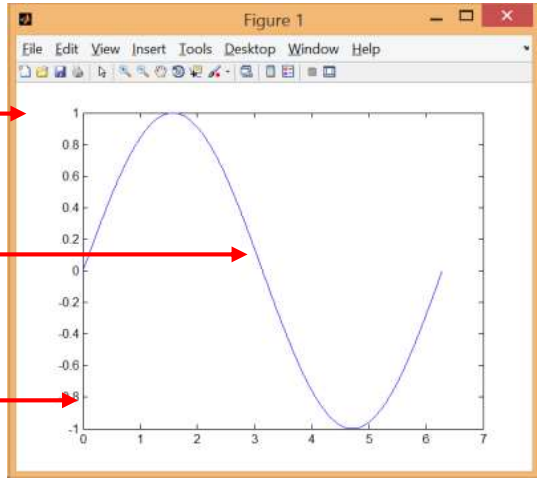
- Upon creation:

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

`gcf` Figure

Line

`gca` Axes



- Utility functions:

Function	Purpose
gca	Return the handle of the “current” axes
gcf	Return the handle of the “current” figure
allchild	Find all children of specified objects
ancestor	Find ancestor of graphics object
delete	Delete an object
findall	Find all graphics objects

Fetching or Modifying Properties

- To fetch properties, use

```
get()
```

- To modify properties, use

```
set()
```

Getting Object Properties

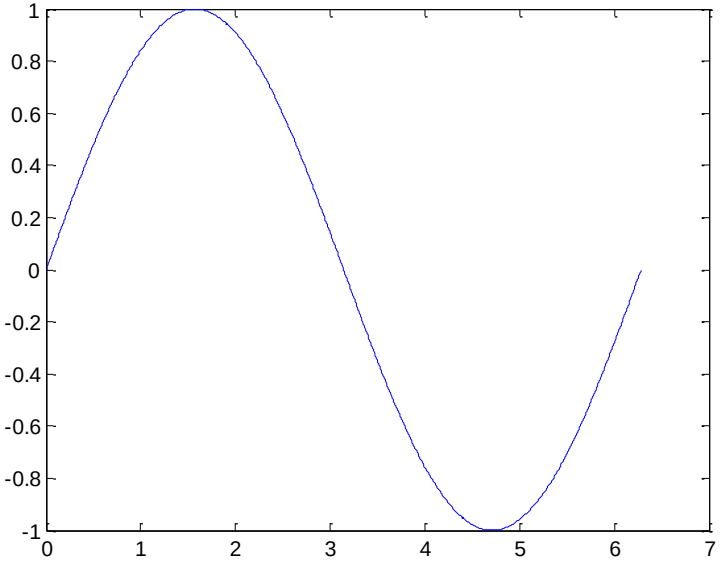
- Getting properties of a graphical object: `get()`
- What do you see after running?

```
x = linspace(0, 2*pi, 1000);
y = sin(x); h = plot(x,y);
get(h)
```

- What do you see?

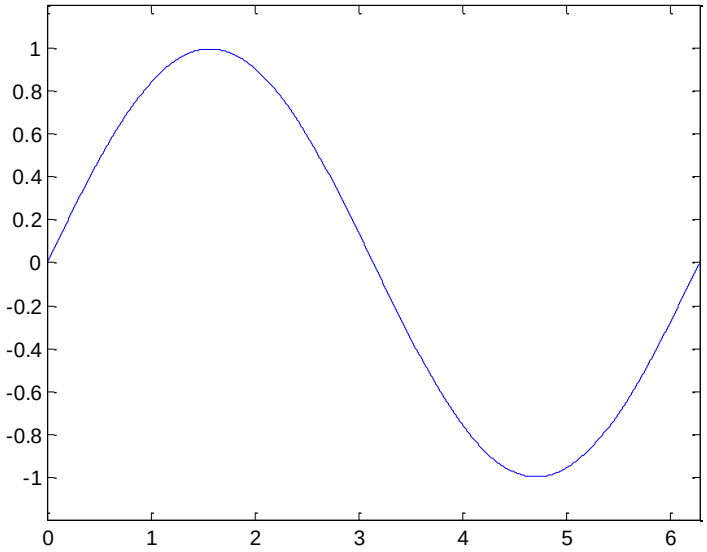
```
get(gca)
```

- Where do we modify the limits of the x-axis?



Setting Axes Limits

```
set(gca, 'XLim', [0, 2*pi]);
set(gca, 'YLim', [-1.2, 1.2]);
```

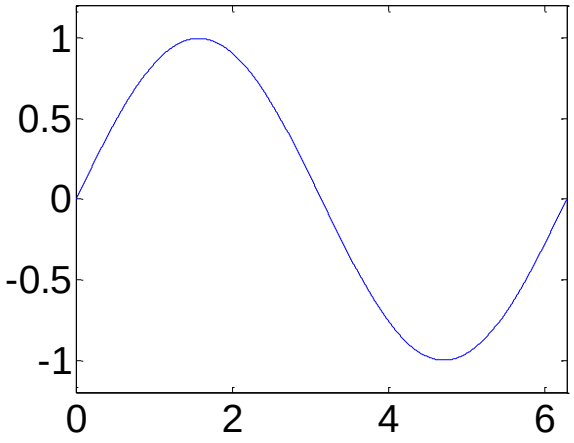


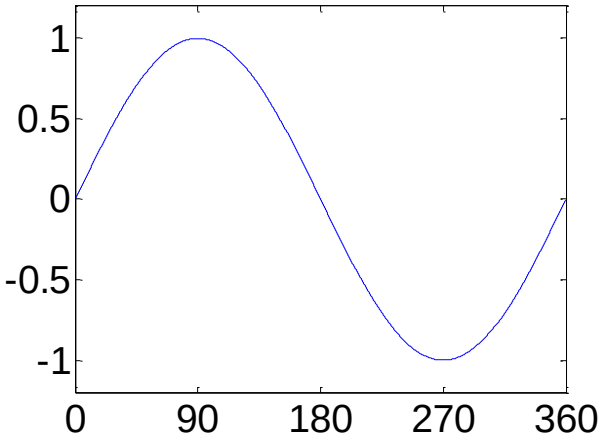
- Alternative:

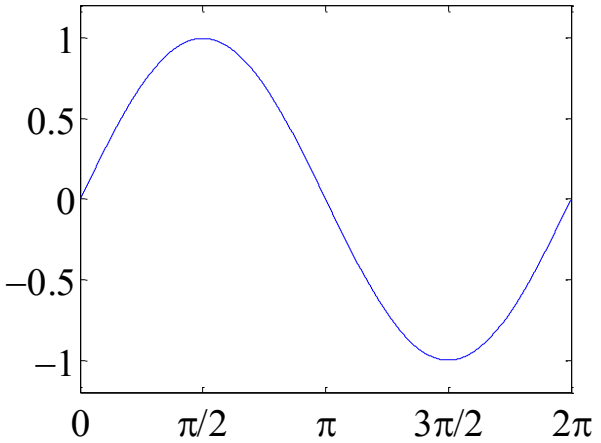
```
xlim([0, 2*pi]);
ylim([-1.2, 1.2]);
```

Setting Font and Tick of Axes

```
set(gca, 'FontSize', 25);
```








```
set(gca, 'XTick', 0:pi/2:2*pi);
set(gca, 'XTickLabel', 0:90:360);
```

```
set(gca, 'FontName', 'symbol');
set(gca, 'XTickLabel', {'0', 'p/2', 'p', '3p/2', '2p'});
```

Line Specification

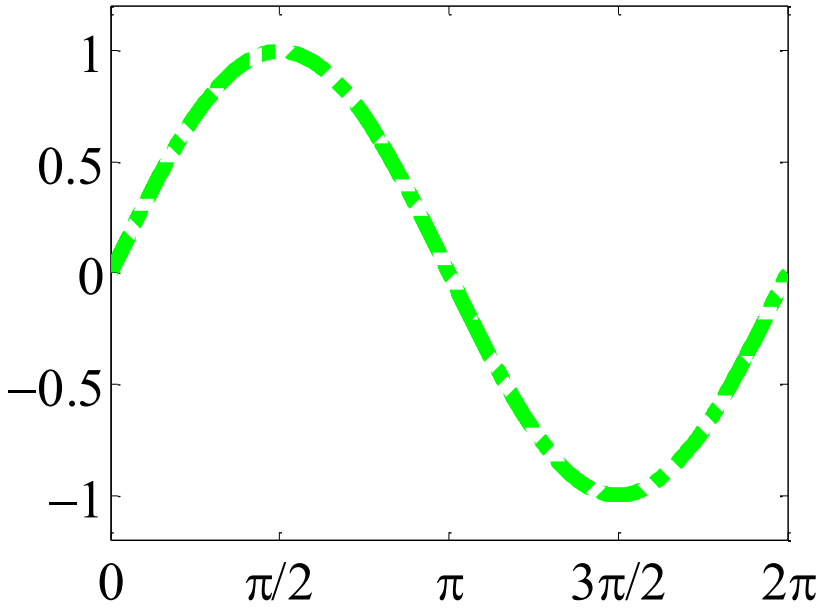
- Line style and width:

```
set(h, 'LineStyle', '-.',...
    'LineWidth', 7.0, 'Color', 'g');
```

- Alternative:

```
plot(x,y, '-.g',...
    'LineWidth', 7.0);
```

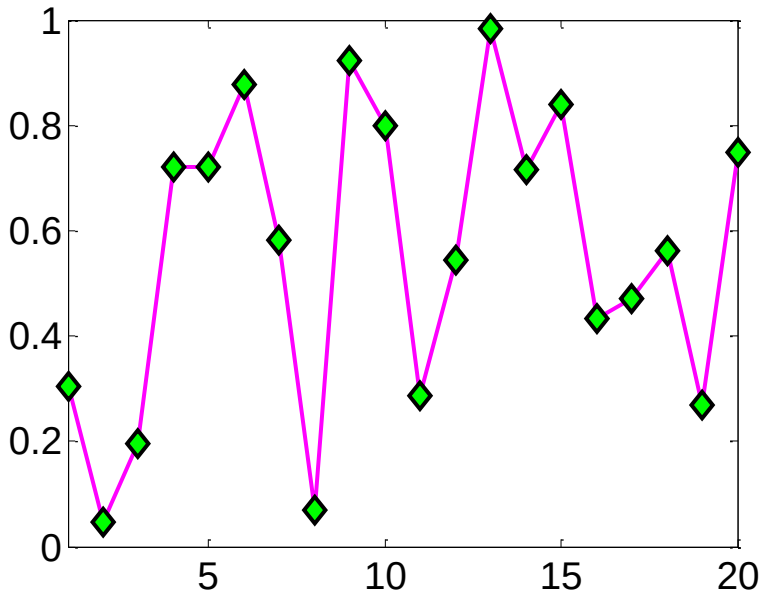
- Try: `delete(h);`



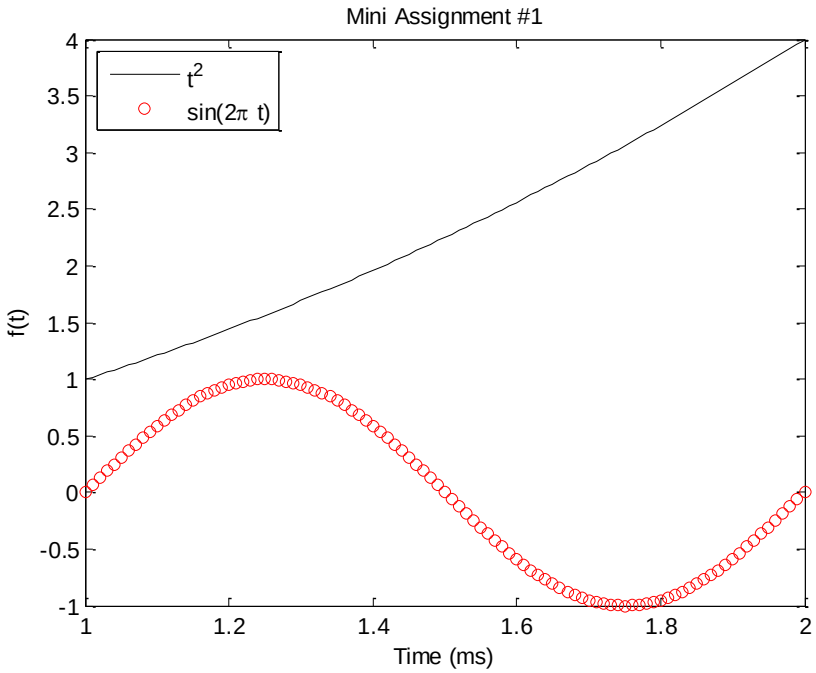
Marker Specification

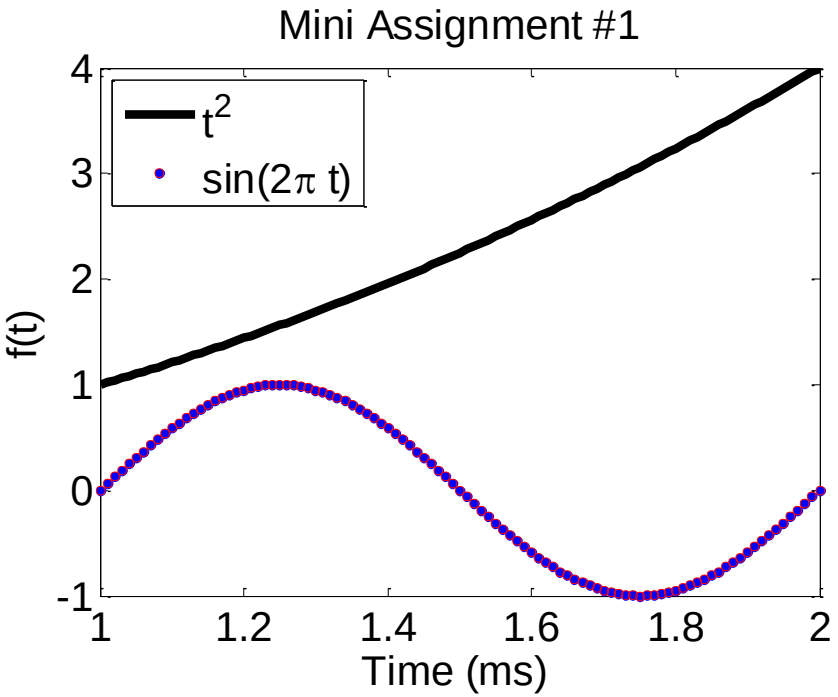
- Face and edge colors of the markder

```
x=rand(20,1);  set(gca, 'FontSize', 18);
plot(x,'-md','LineWidth', 2, 'MarkerEdgeColor', 'k',...
    'MarkerFaceColor', 'g', 'MarkerSize', 10);
xlim([1, 20]);
```



Exercise

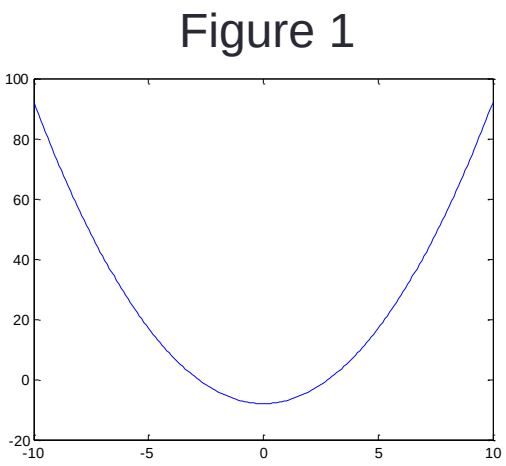

Mini Assignment #1
t2
sin(2π t)
f(t)
Time (ms)
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
1
1.2
1.4
1.6
1.8
2

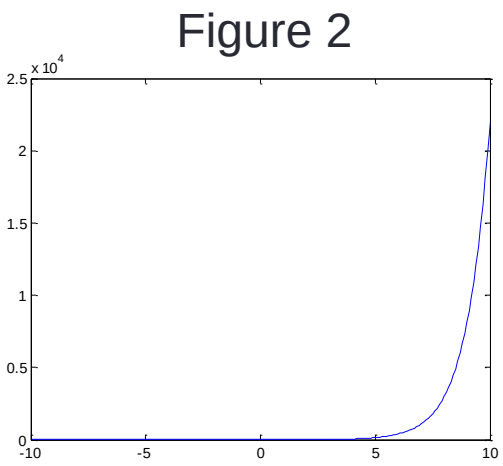

Mini Assignment #1
t2
sin(2π t)
f(t)
Time (ms)
4
3
2
1
0
-1
1
1.2
1.4
1.6
1.8
2

Multiple Figures

- Create a figure window by calling `figure`

```
x = -10:0.1:10;
y1 = x.^2 - 8;
y2 = exp(x);
figure, plot(x,y1);
figure, plot(x,y2);
```

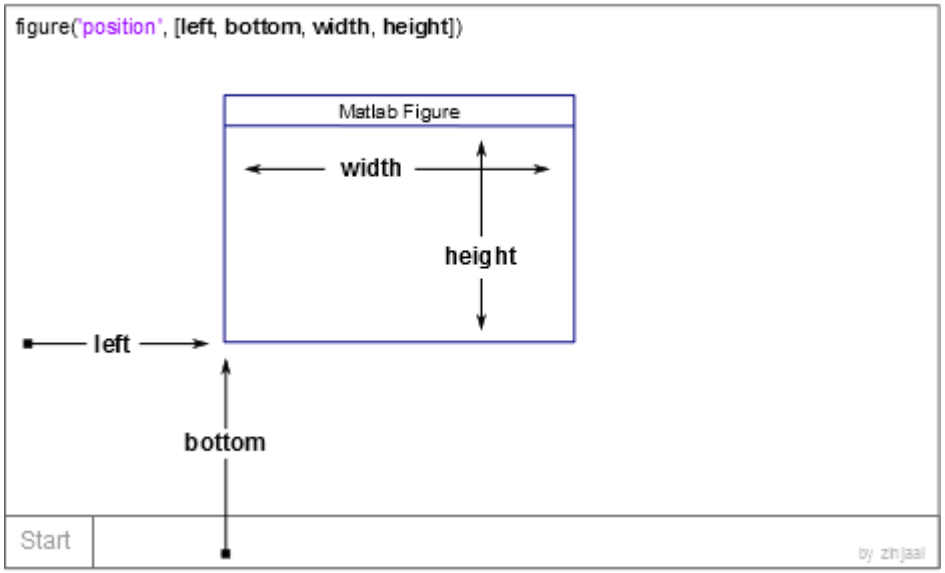




- Be careful when using the `gcf` handle where there exists multiple figures

Figure Position and Size

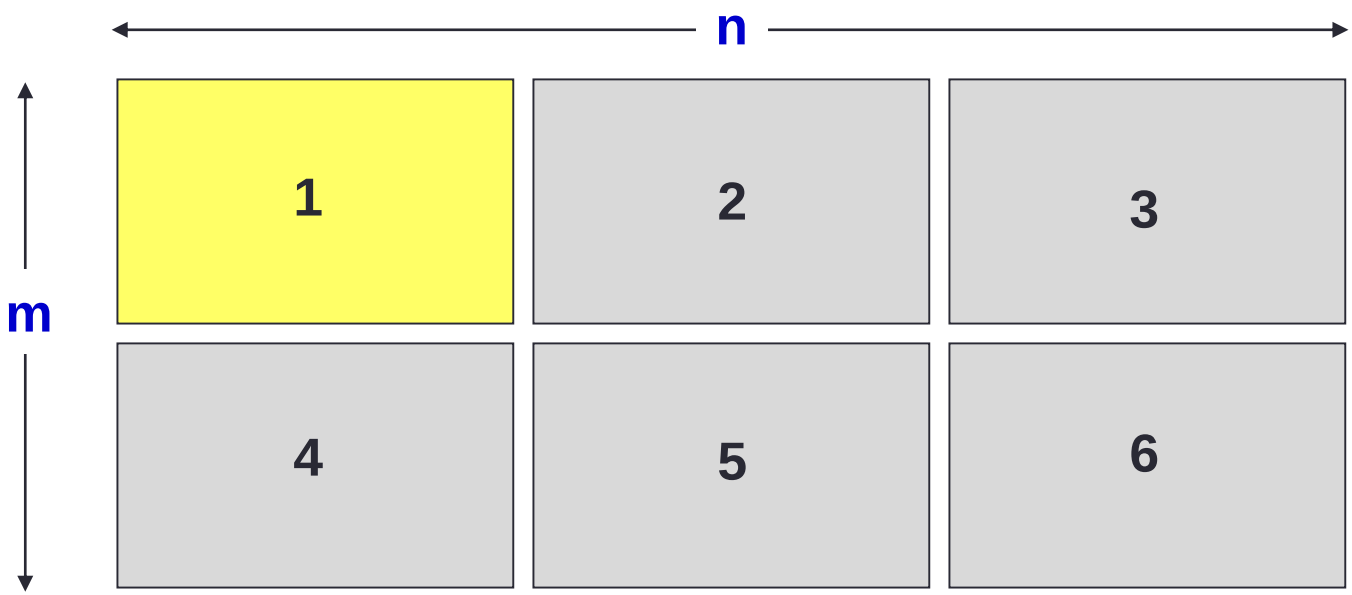
```
figure('Position', [left, bottom, width, height]);
```



Several Plots in One Figure

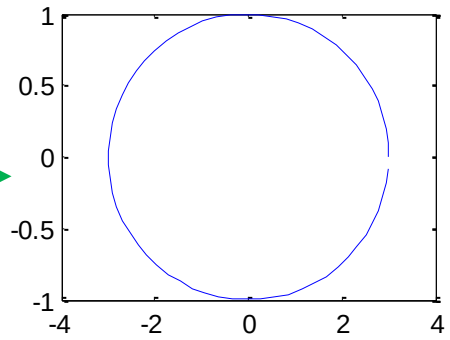
- Several small plots “in a figure”

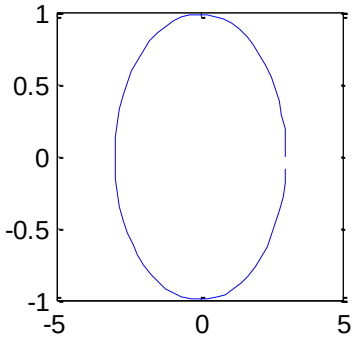
```
subplot(m, n,1);
```



subplot()

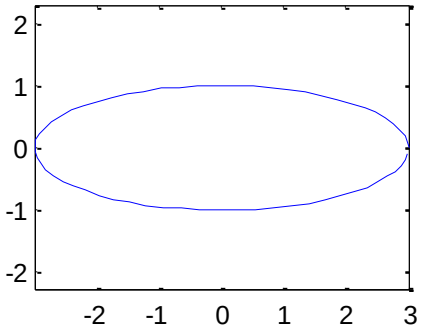
axis normal

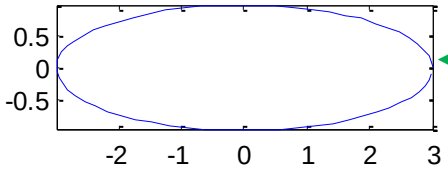




axis square

axis equal





axis square tight

```
t = 0:0.1:2*pi; x = 3*cos(t); y = sin(t);
subplot(2, 2, 1); plot(x, y); axis normal
subplot(2, 2, 2); plot(x, y); axis square
subplot(2, 2, 3); plot(x, y); axis equal
subplot(2, 2, 4); plot(x, y); axis equal tight
```

Control of Grid, Box, and Axis

`grid on/off`	Make the grid visible or invisible
`box on/off`	Make the box visible or invisible
`axis on/off`	Make the axes visible or invisible
`axis normal`	Automatically adjust the aspect ratio of the axes and the relative scaling of the data units
`axis square`	Make the current axes region square
`axis equal`	Set the aspect ratio so that the data units are the same in every direction
`axis equal tight`	Set the axis limits to the range of the data
`axis image`	Let the plot box fits tightly around the data
`axis ij`	Place the origin of the coordinate system in the upper left corner
`axis xy`	Place the origin in the lower left corner

Saving Figures into Files

```
saveas(gcf,'<filename>','<formattype>');
```

Option	Bitmap Image Format
'jpeg'	JPEG 24-bit
'png'	PNG 24-bit
'tiff'	TIFF 24-bit (compressed)
'bmpmono'	BMP Monochrome
'bmp'	BMP 24-bit
'bmp256'	BMP 8-bit (256 color, uses a fixed colormap)

Option	Vector Graphics Format
'pdf'	Full page Portable Document Format (PDF) color
'eps'	Encapsulated PostScript (EPS) Level 3 black and white
'epsc'	Encapsulated PostScript (EPS) Level 3 color
'meta'	Enhanced Metafile (Windows only)
'svg'	SVG (scalable vector graphics)
'ps'	Full-page PostScript (PS) Level 3 black and white
'psc'	Full-page PostScript (PS) Level 3 color

To control size and resolution, use `print` instead:
http://www.mathworks.com/help/matlab/ref/print.html#input_argument_formattype

End of Class