MATLAB Plots

Week 7

Loosely follows Chapter 9

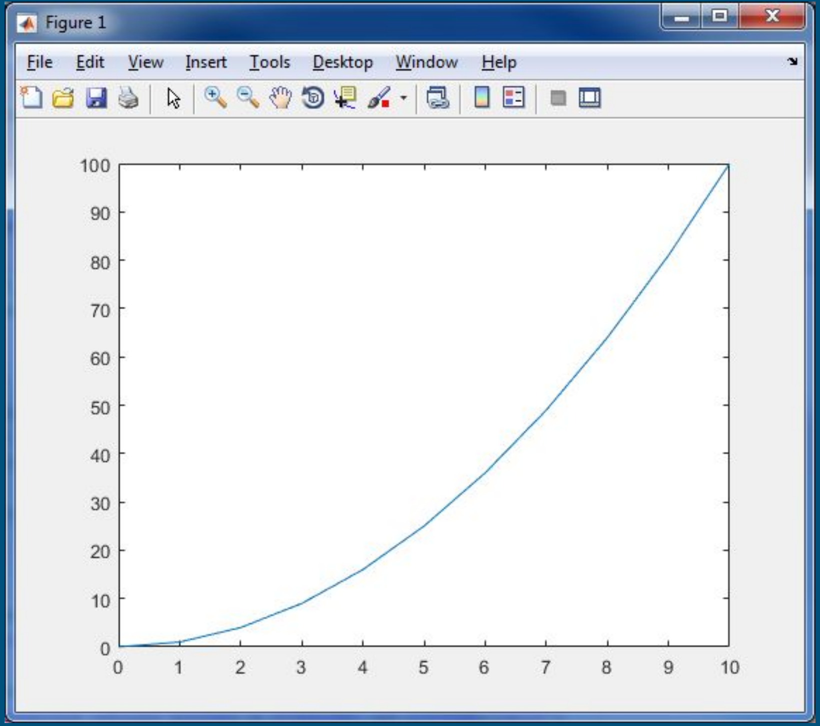
Plotting in 2D

Usefulness of Plotting Data

- Large sets of data can be difficult to view as tables or text
- Graphical techniques reduce large sets of data to help gain insight
- Graphical representations make trends and possible errors easily discoverable

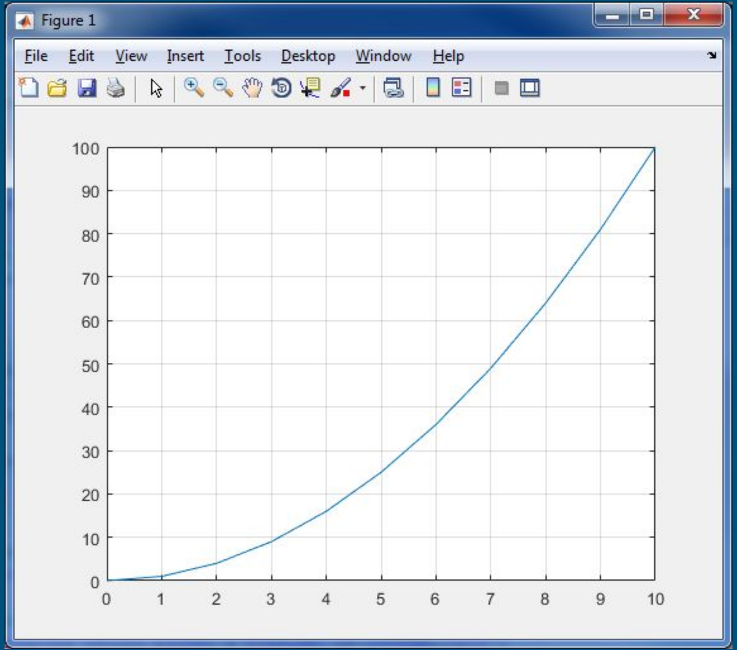
2D (x, y) plots

- Creating (x, y) plots is easy with plot
- Create two vectors
  - x = 0:10;
  - y = x.^2;
- Use the plot() command:
  - plot(x, y);



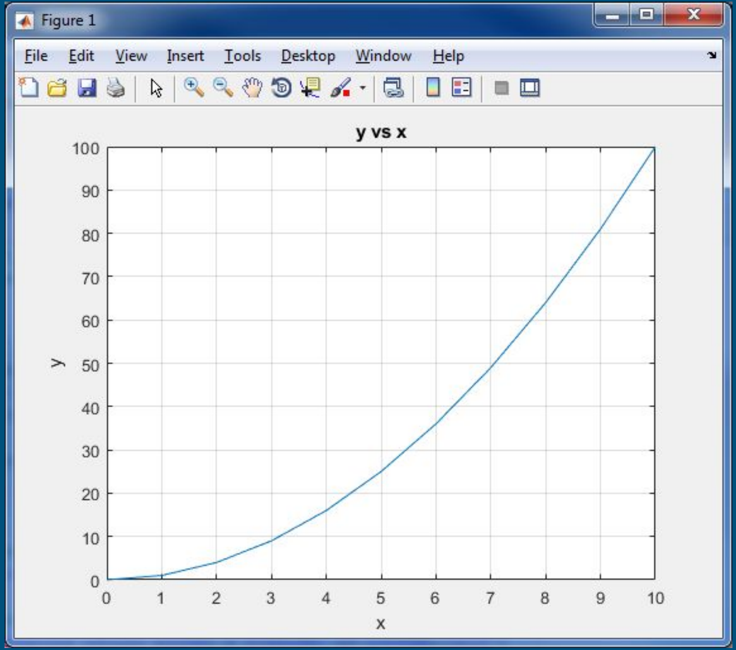
Add a grid

- Use the grid command to add a grid to the figure
    - grid on;



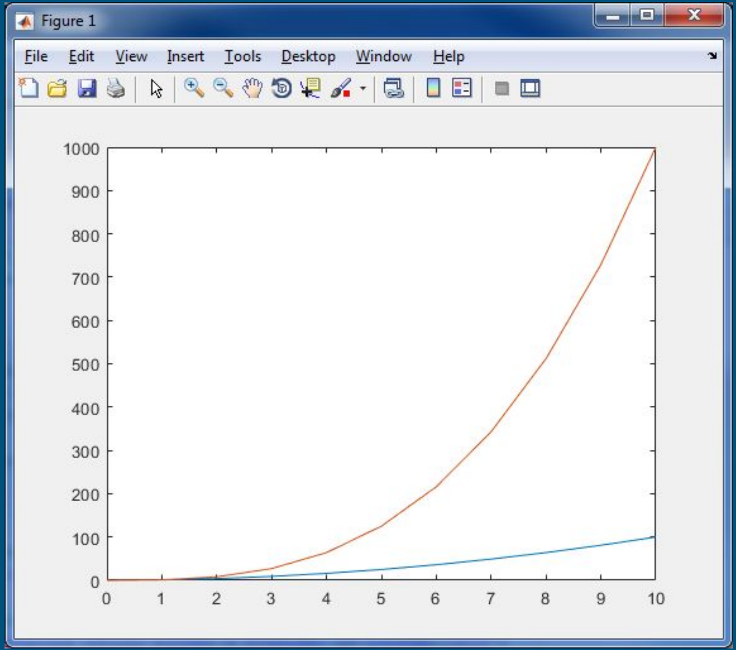
Add a Title and Labels

- Add a title and labels to a plot
  - title('y vs x');
  - xlabel('x');
  - ylabel('y');



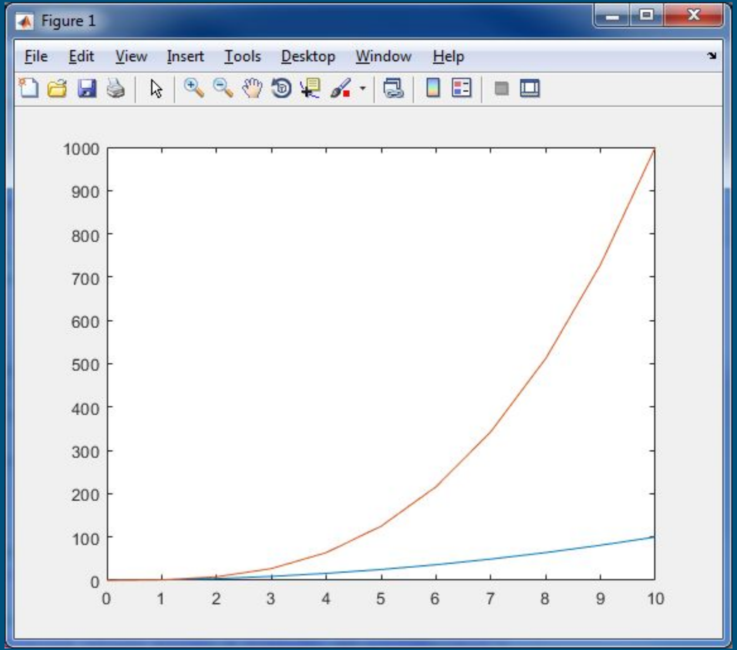
Plot Multiple Curves

- Plot multiple curves on a figure with the following
  - x = 1:10;
  - y1 = x.^2;
  - y2 = x.^3;
  - plot(x, y1, x, y2);



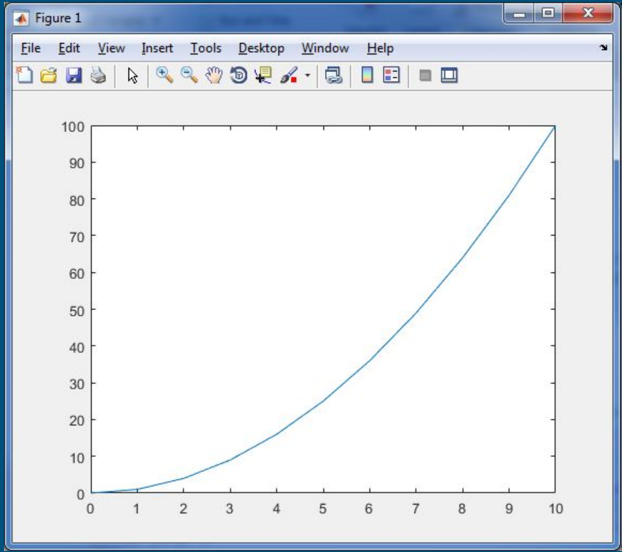
Plot Multiple Curves

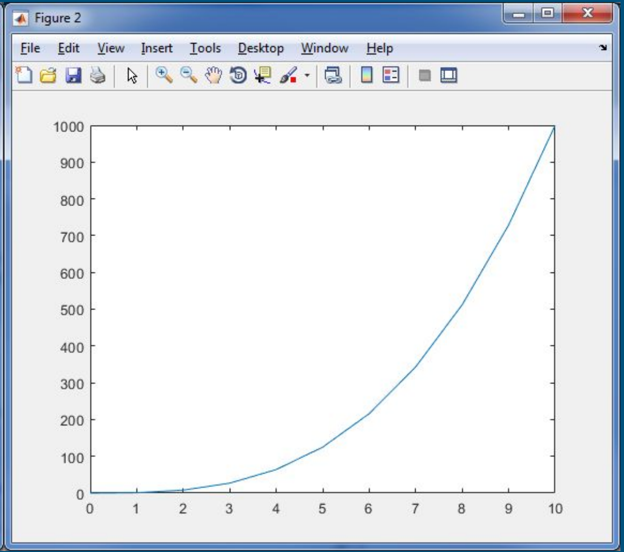
- Multiple curves can be plotted using the hold command
    - plot(x, y1);
    - hold on;
    - plot(x, y2);
    - …
    - hold off;



Multiple Figures

- Single (or multiple) plots can be created in multiple figures.
  - figure(1);
  - plot(x, y1);
  - figure(2);
  - plot(x, y2);





Subplots

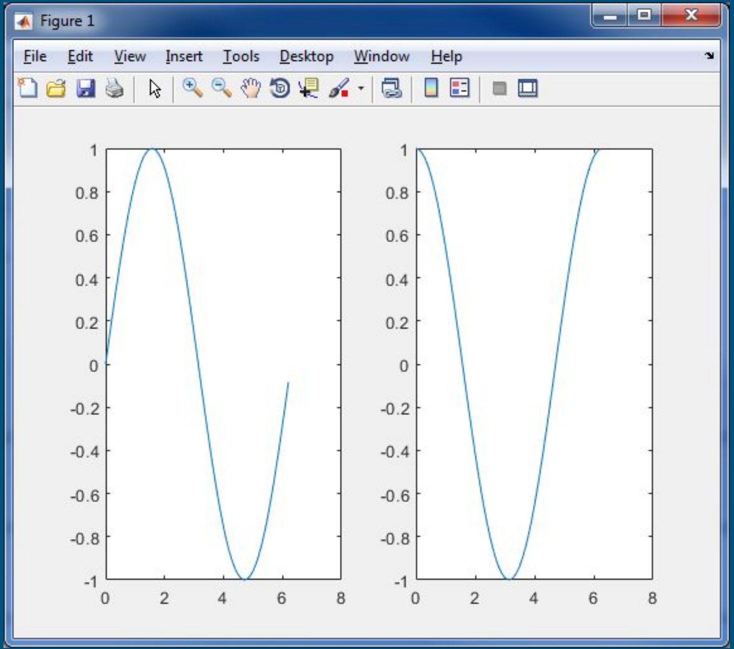
- The subplot command allows you to put multiple graphs in one figure
- The format is subplot(m, n, p) where a grid is created with
  - m rows
  - n columns
  - p representing the active ‘cell’ in which to plot or manipulate the plot.

p == 1	p == 2
p == 3	p == 4

Examples of Subplots

- To graph sin(x) and cos(x) on the same figure side-by-side perform the following set of commands.

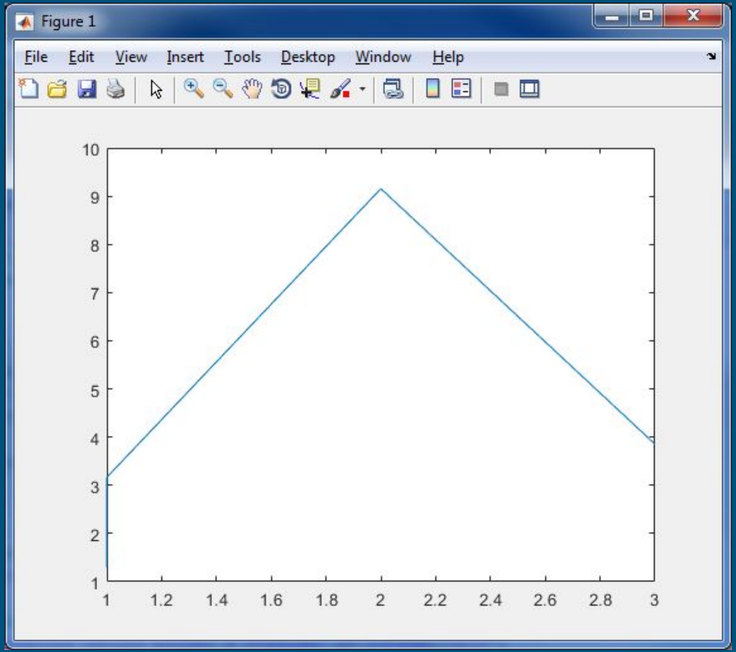
```
x = 0:0.1:2*pi;
subplot(1,2,1);
plot(x, sin(x));
subplot(1,2,2);
plot(x, cos(x));
```



Plots of Complex Numbers

- MATLAB plots the real component on the X axis and the imaginary on the Y axis. However, with a single axis of imaginary numbers…

```
A = [ 1+1i, 1+2i, 2+3i, 3+4i ];
B = sin(A);
plot(A, B);
```



Warning: Imaginary parts of complex X and/or Y arguments ignored.

Line Style, Point Style, Color

- If we want to plot a ‘dash-dot’ line with red stars for points

```
x = 1:10;
y = [4, 6, 8, 3, 11, 8, 5, 2, 9, 4];
plot(x, y, 'r-.*');
```

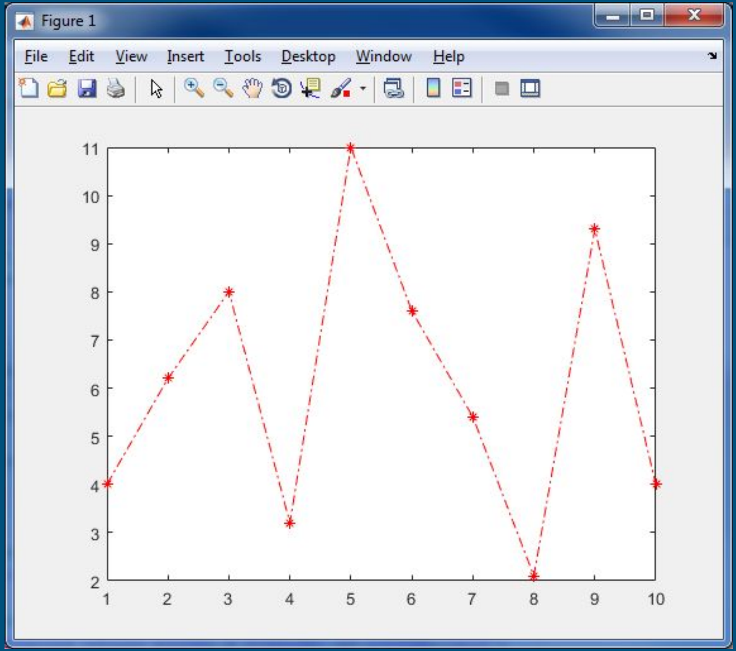


Table of Options

Line Type	Indicator
solid	-
dotted	:
dash-dot	-.
dashed	--

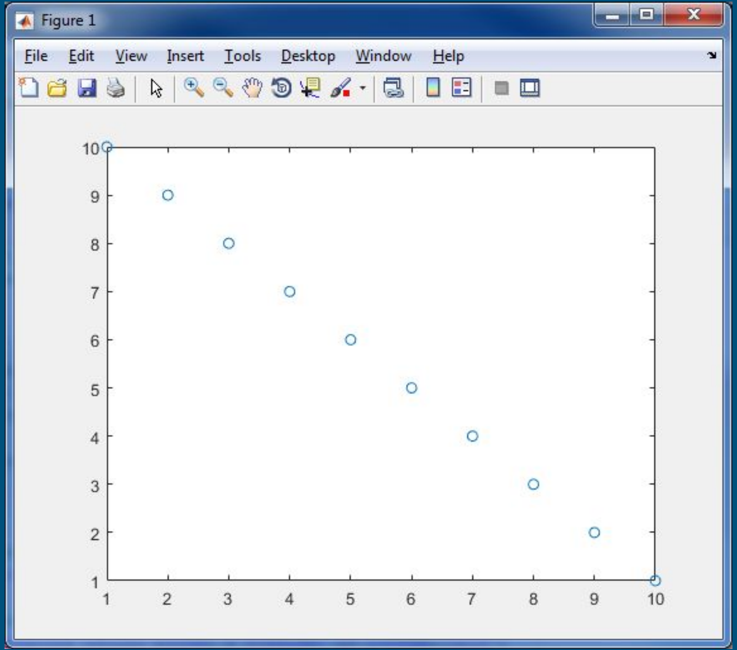
Point Type	Indicator
point	.
circle	o
x-mark	x
plus	+
star	*
square	s
diamond	d
triangle down	v
triangle up	^
triangle left	<
triangle right	>
pentagram	p
hexagram	h

Color	Indicator
blue	b
green	g
red	r
cyan	c
magenta	m
yellow	y
black	b

Plot Individual Data Points

- Setting a marker type without specifying a line type suppresses the default straight line normally drawn between points.

```
x = 1:10;
y = 10:-1:1;
plot(x, y);
plot(x, y, 'o');
```

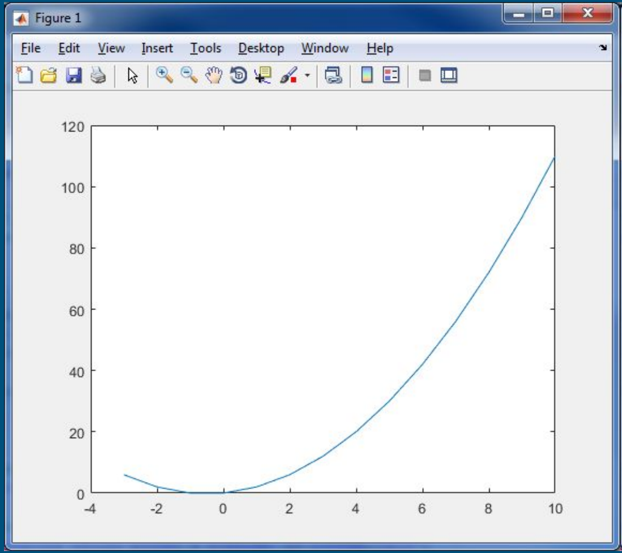


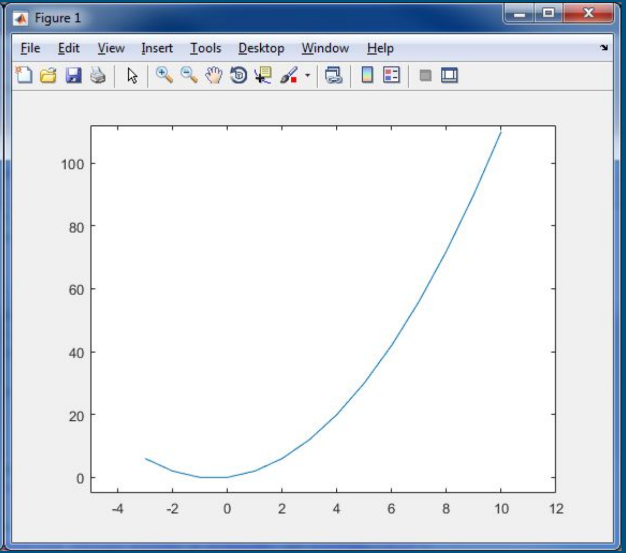
Adjust Axis Limits

- You can override axis limits, in this case let’s adjust so the line doesn’t touch the edges.

```
x = -3:10;
y = x.^2 + x;
plot(x, y);
axis([ -5, 12, -5, 112 ]);
```

axis([xmin, xmax, ymin, ymax])





Annotations

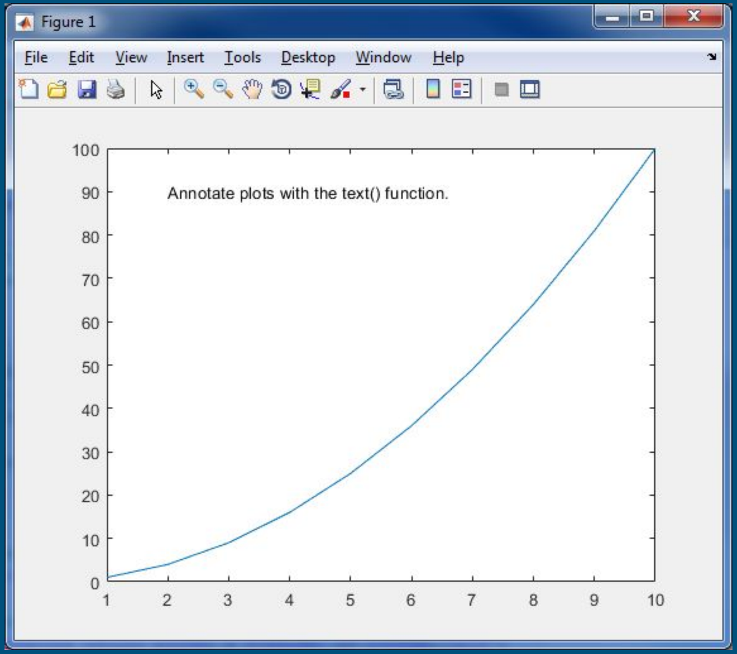
- Annotations can be used to add information

```
x = 1:10;
y = x.^2;
plot(x, y);
text(2, 90, “Annotate plots with the text() function.”);
```

- For interactive label placement

```
gtext(“place me anywhere”);
```

- Note, the position is with respect to the x,y coordinates

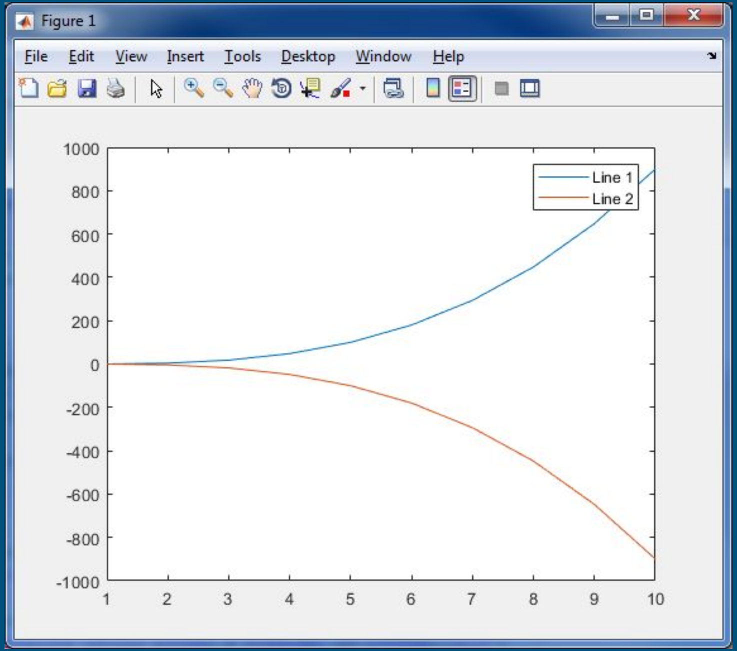


Legends

- Add a legend using the legend function.

```
x = 1:10;
y1 = x.^3 = x.^2;
y2 = (-1*x).^3 + x.^2;
plot(x, y1, x, y2);
legend('Line 1', 'Line 2');
```

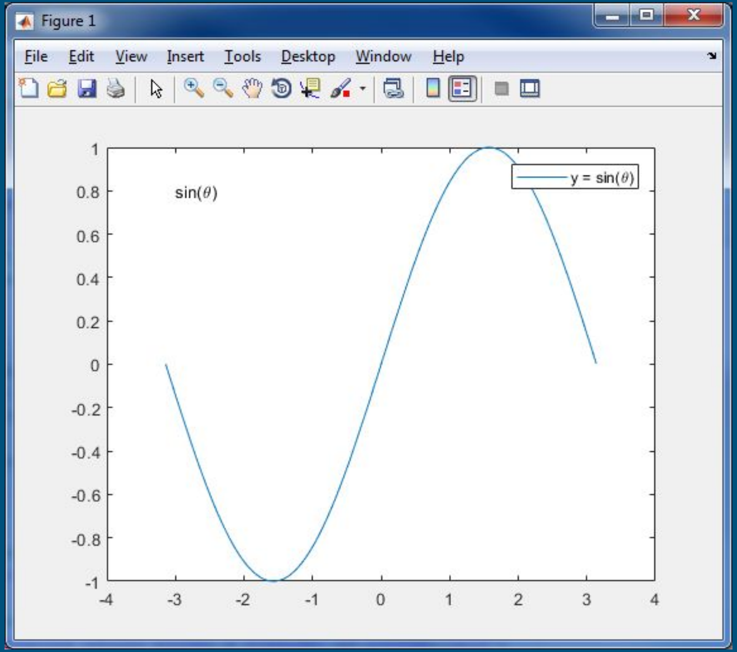
- Legends can be moved with the mouse



Special Characters in Text

- Special characters can be used in legends and text

```
theta = -pi:0.01:pi;
y = sin(theta);
plot(theta, y);
legend('y = sin(\theta)');
text(-3, 0.8, 'sin(\theta)');
```



Key Takeaways

- Adding a grid, labels, text and a legend
- Scaling the axis
- Plotting multiple curves on a figure
- Plotting with multiple figures
- Plotting complex arrays
- Changing line styles, color and point styles

Types of Plots

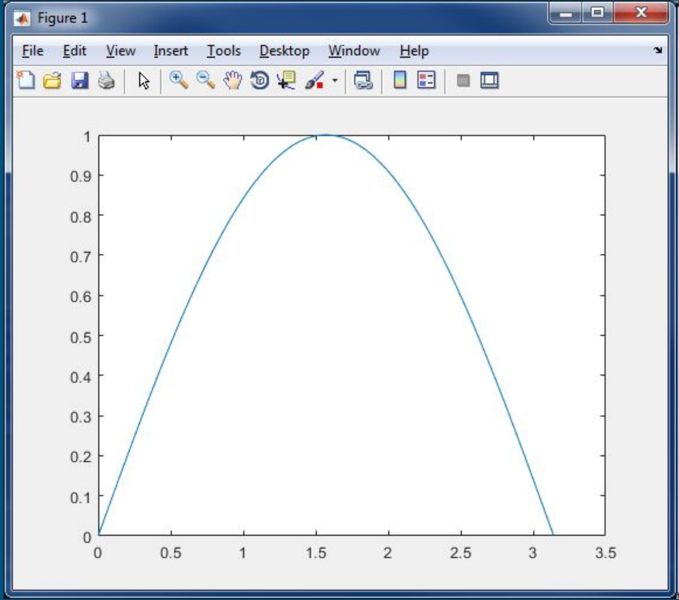
Week 7

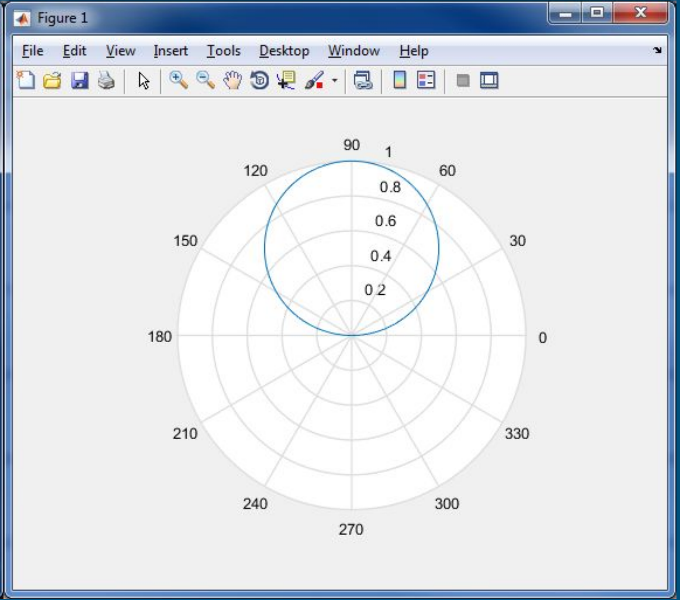
Loosely follows Chapter 9

Polar Plots

- MATLAB supports tools for plotting in polar coordinates.

```
theta = 0:0.01:pi;
r = sin(theta);
plot(theta, r);
polar(theta, r);
```



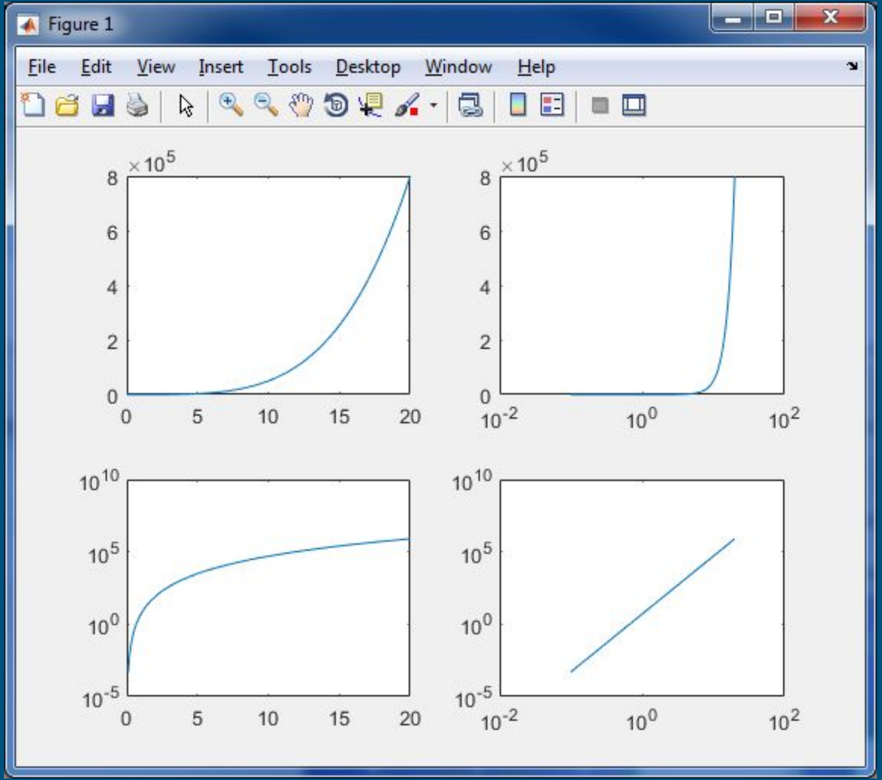


Logarithmic Plots

- MATLAB has tools for three kinds of logarithmic plots
    - semilogx
    - semilogy
    - loglog
- These utilities automatically replace linear scales with logarithmic scales
- Logarithmic scales are useful when a variable ranges over many orders of magnitude.

Logarithmic Example

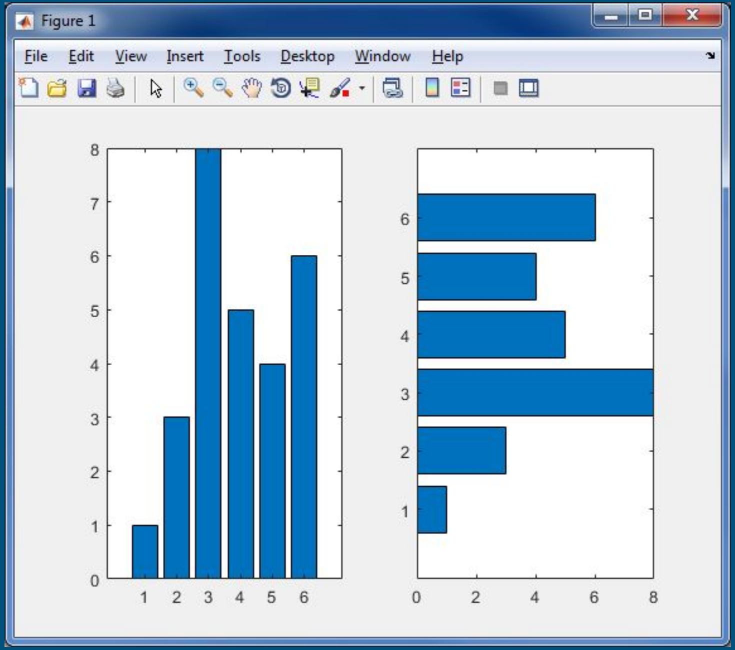
```
x = 0:0.1:20;
y = 5*x.^4
subplot(2,2,1);
plot(x, y);
subplot(2,2,2);
semilogx(x, y);
subplot(2,2,3);
semilogy(x, y);
subplot(2,2,4);
loglog(x, y);
```



Bar Charts

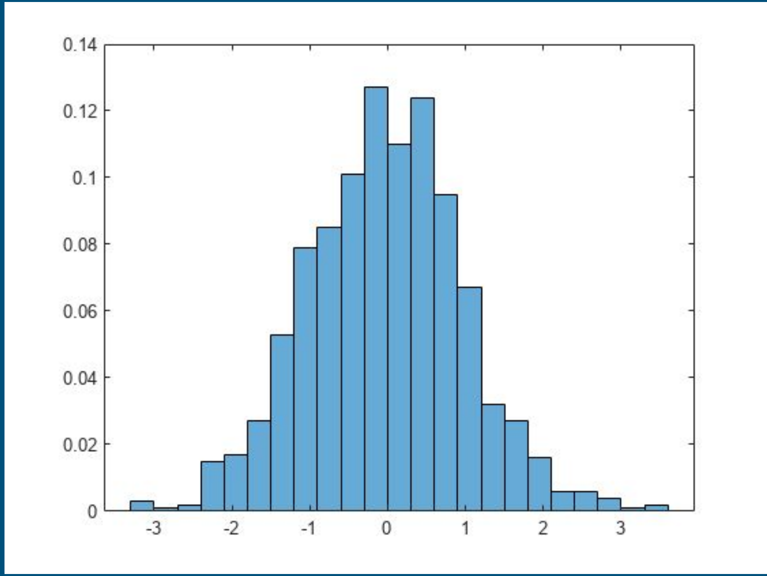
- Bar Charts are useful for comparing categorical data.
- For a vertical bar chart, use bar(x)
- For a horizontal bar chart, use barh(x)

```
x = [1,3,8,5,4,6];
subplot(1,2,1);
bar(x);
subplot(1,2,2);
barh(x);
```



Histograms

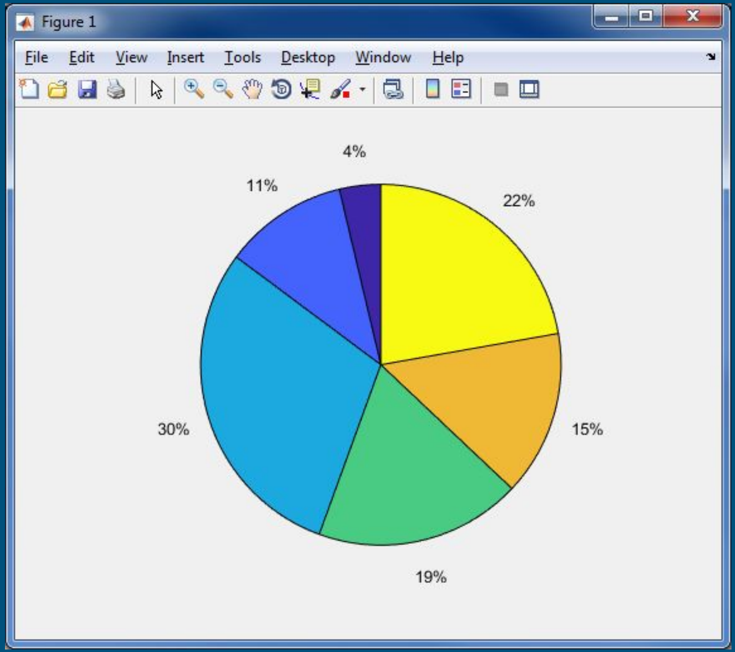
- Bar Charts are useful for viewing quantitative/continuous data.
- For a histogram, use histogram(x)



Pie Charts

- Pie charts offer another useful means of comparing categorical data.

```
x = [1,3,8,5,4,6];
pie(x);
```



8/(1+3+8+5+4+6) ≈ 30%
6/(1+3+8+5+4+6) ≈ 19%
etc.

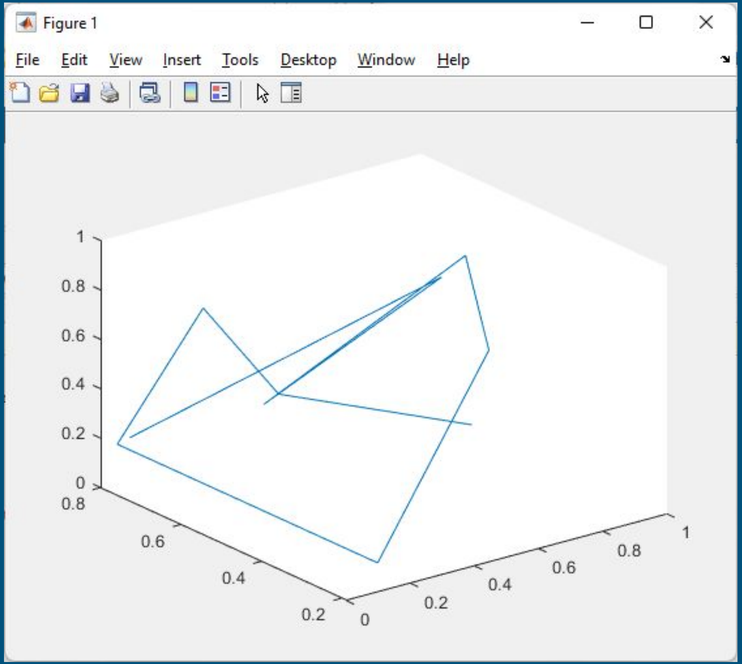
3D Plots

Week 7

Loosely follows Chapter 9

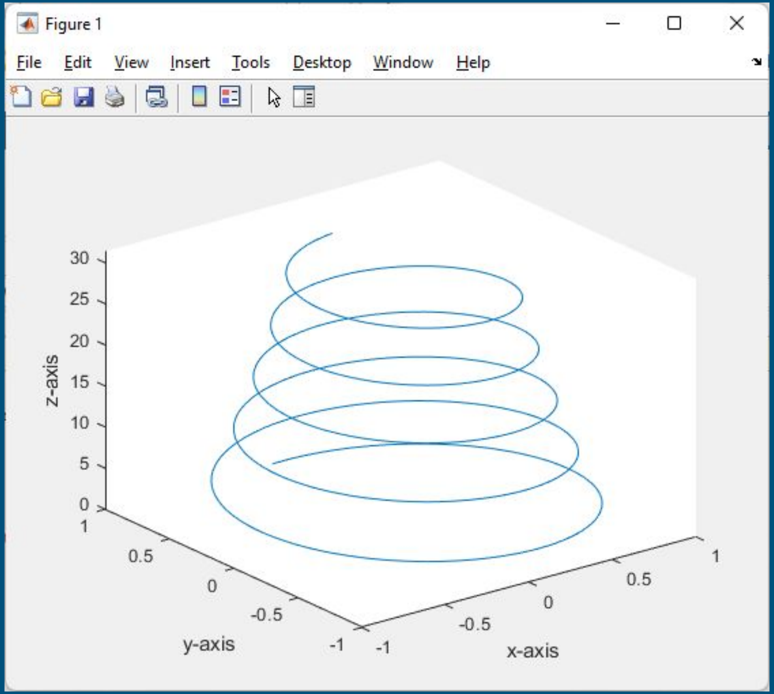
Plot Random Values

```
plot3(rand(1,10), rand(1,10), rand(1,10))
```



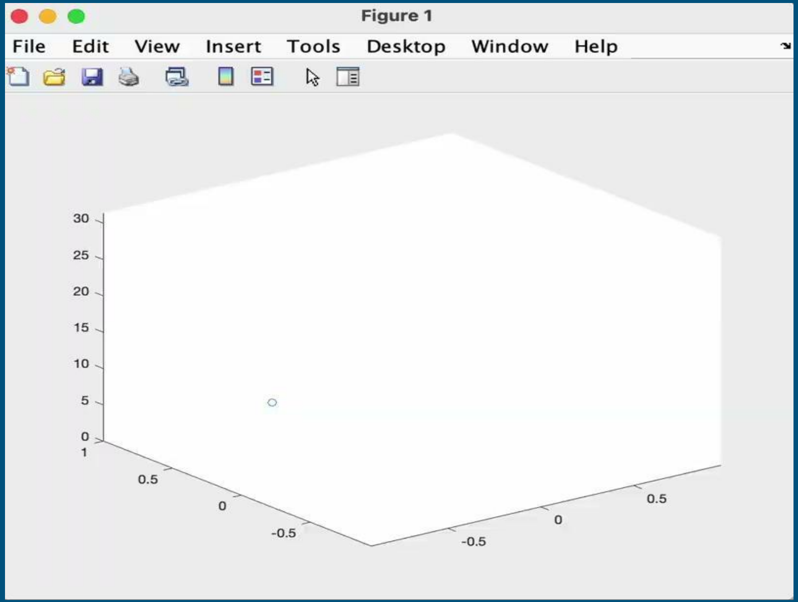
Cone Shaped Spring

```
t = 0:pi/50:10*pi;
plot3(exp(-0.02*t).*sin(t), exp(-0.02*t).*cos(t),t), ...
        xlabel('x-axis'), ylabel('y-axis'), zlabel('z-axis')
```



This is fun

```
t = 0:pi/50:10*pi;
comet3(exp(-0.02*t).*sin(t), exp(-0.02*t).*cos(t),t), ...
        xlabel('x-axis'), ylabel('y-axis'), zlabel('z-axis')
```



Mesh Surfaces

Say we want to see $z = x^2 - y^2$

We'll set up a meshgrid for values 0 to 5

Curve Fitting

Week 7

Loosely follows Chapter 9

Plotting and Curve Fitting

- Curve fitting is a powerful way to use a set of data to find a mathematical model that approximates that set of data.

Curve Fitting

- The simplest way to fit a set of 2D data is a straight line
- Linear Regression is one method of fitting data with a straight line

- Linear regression minimizes the squared distance between data points and the equation modeling those points. This prevents positive and negative "errors" from cancelling.

Linear Approximation by Hand

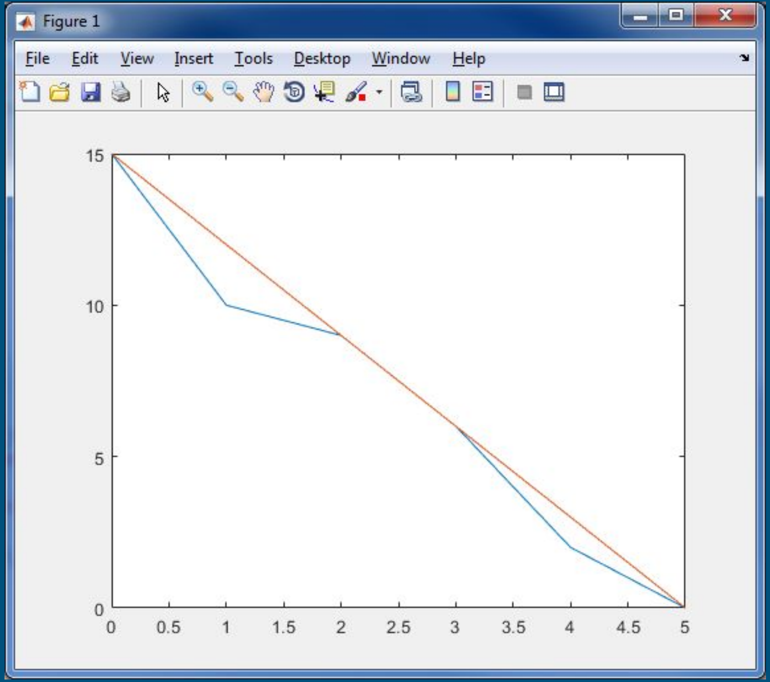
- Given

```
x = [ 0, 1, 2, 3, 4, 5 ];
y = [ 15, 10, 9, 6, 2, 0 ];
```

- slope ≈ $(y_2 - y_1)/(x_2 - x_1) = (0 - 15)/(5 - 0) = -3$
- This crosses the y axis at 15 therefore

  $y_{hand} = -3x + 15$

```
y_hand = -3.*x+15;
sum_of_squares = sum((y - y_hand).^2) = 5;
plot(x, y, x, y_hand);
```



Polynomial Regression

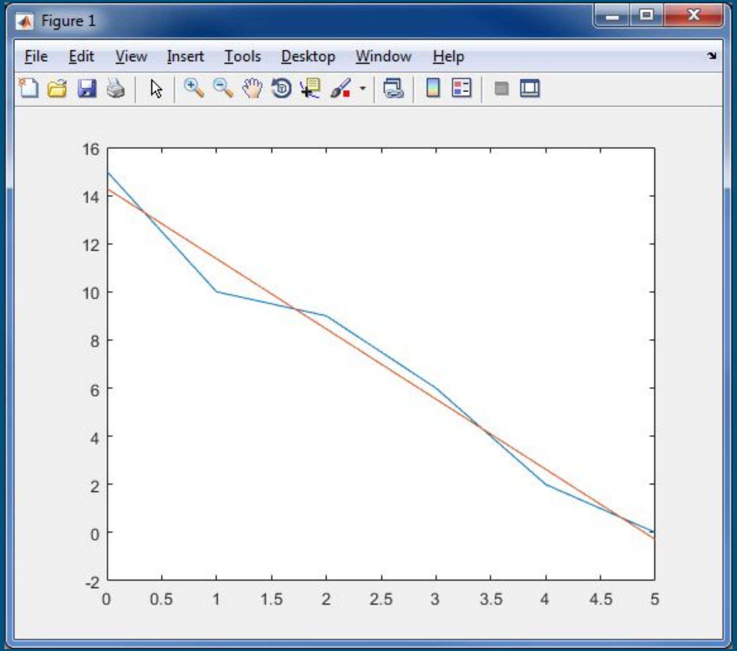
- Polynomial Regression is used to fit a set of data with a polynomial
- The polyfit function can be used to find the best fit polynomial of a specified degree; the result is the coefficients.

NOTE: Increasing the degree of the best fit polynomial can create mathematical models that may fit the data better. This can lead to overfitting so care must be taken in your interpretation of the result.

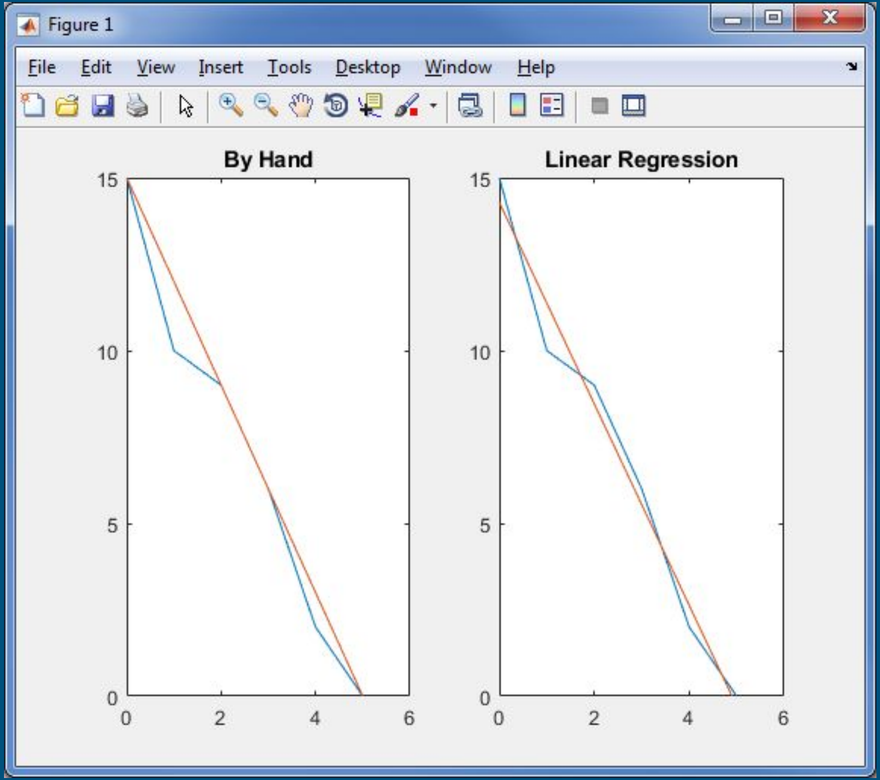
Polyfit Function

- The polyfit(x, y, n) function takes x and y as parameters and the degree of a polynomial n as input.
- Linear regression with polyfit

```
x = [ 0, 1, 2, 3, 4, 5 ];

y = [ 15, 10, 9, 6, 2, 0 ];

p = polyfit(x, y, 1);

y_calc = p(1)*x + p(2)

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



Best Fit Comparison



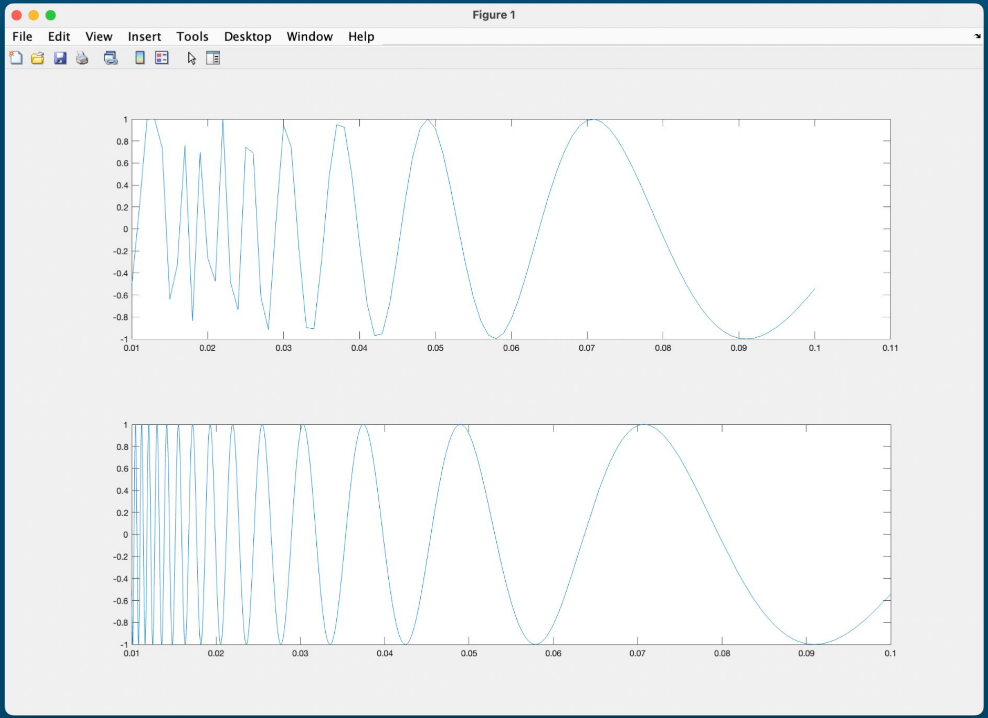
The polyval Function

- Remember that polyfit returns the coefficients of a polynomial that best fits the data.
- To evaluate the polynomial at any value of x, use the polyval function
- polyval(c, x) accepts the array of coefficients (c) and the array of x values (x) at which the polynomial is to be evaluated.

The fplot Function

- Good for rapidly changing data
    - ex. sin, cos, tan
- Improves line smoothness

```
figure(1)
subplot(2,1,1)
x = 0.01:0.001:0.1;
plot(x, sin(x./x)
subplot(2,1,2)
fplot('sin(1/x)', [0.01 0.1])
```



Key Takeaways

- Subplots
- Polar and Logarithmic plots
- Bar charts and pie charts
- Curve fitting using
    - Linear regression
    - polyfit
    - polyval