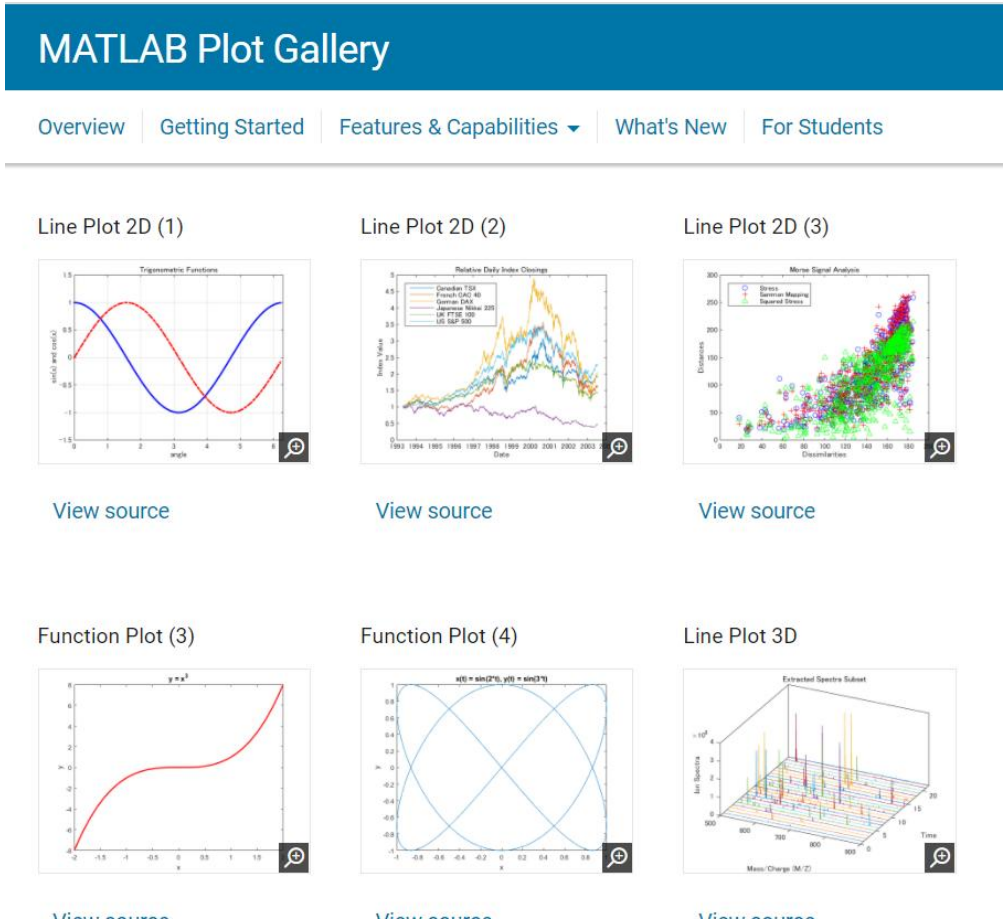
ENME 303 LAB

Week 10: 2D Plotting

Nameless Lab

This lecture covers the basics of 2D plotting. For more cool stuff on plotting, check out [Mathworks plotting gallery]

- Use '**close all**' to close all the figure and plots.



Plotting Overview

1. We will be covering:
    a. How to create plots from a set of data
    b. Set plot attributes to get it to look how you want
2. MATLAB sees no difference between a graph and a chart. **Everything is a plot**
    a. Graphs visually show relationship between 2 or more variables each associated with their own axis that vary continuous
        i. **Line, scatter, or histograms**
    b. Charts visually show data where the x-axis does not represent continuous variables
        i. **Pie or bar chart**

Matlab Built-in Functions for 2D Plots

Plot Type	**2D Plot**
Line graph	plot(), ezplot(), fplot(), semilogz(), semilogy(), loglog()
Scatter plot	scatter()
Histogram	hist(), histc()
Pie chart	pie()
Bar chart	bar(), barh()
Contour plot	contour()

Read the documentation for these functions. It contains a lot of useful information!

Plotting in MATLAB with plot()

Common way to plot line graphs is with **plot(X,Y)** function, where X and Y are row vectors of the same size that represent the set of values on the x and y axes.

General process for plotting any function y = f(x):

• Define your interval of interest ○ The domain and size of steps	% From -2*pi to 2*pi in increments of 0.1 **X = -2*pi:0.1:2*pi**
• Define your function	**Y = sin(X)**
• Use appropriate 2D function	**plot(X,Y)**

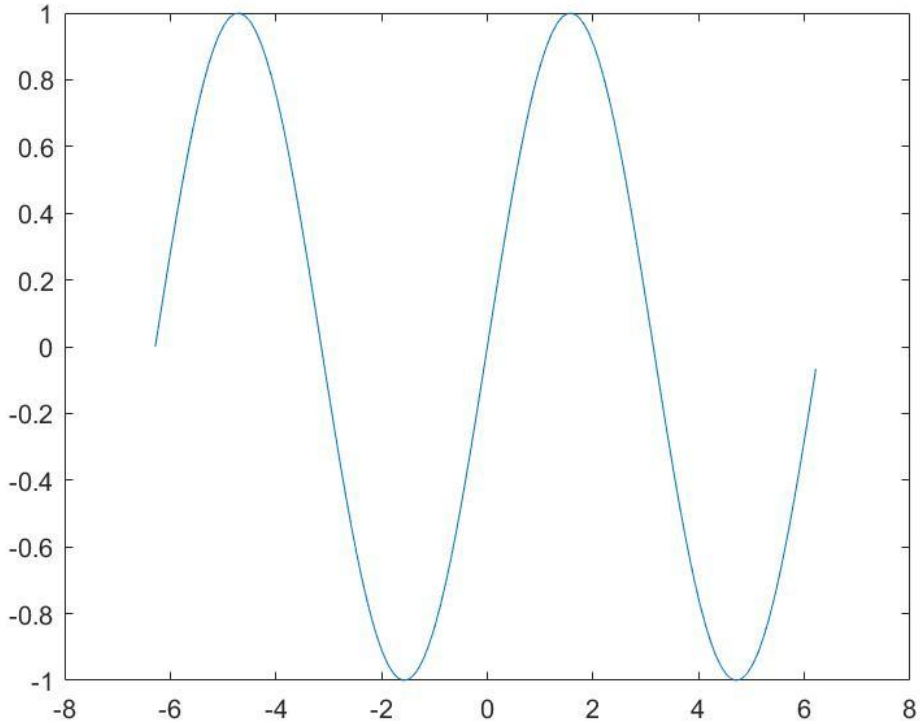
%% Plotting Intro Example

% Domain says from -2*pi to 2*pi in increments of 0.1

X = -2*pi:**0.1**:2*pi;

Y = sin(X);

plot(X,Y)



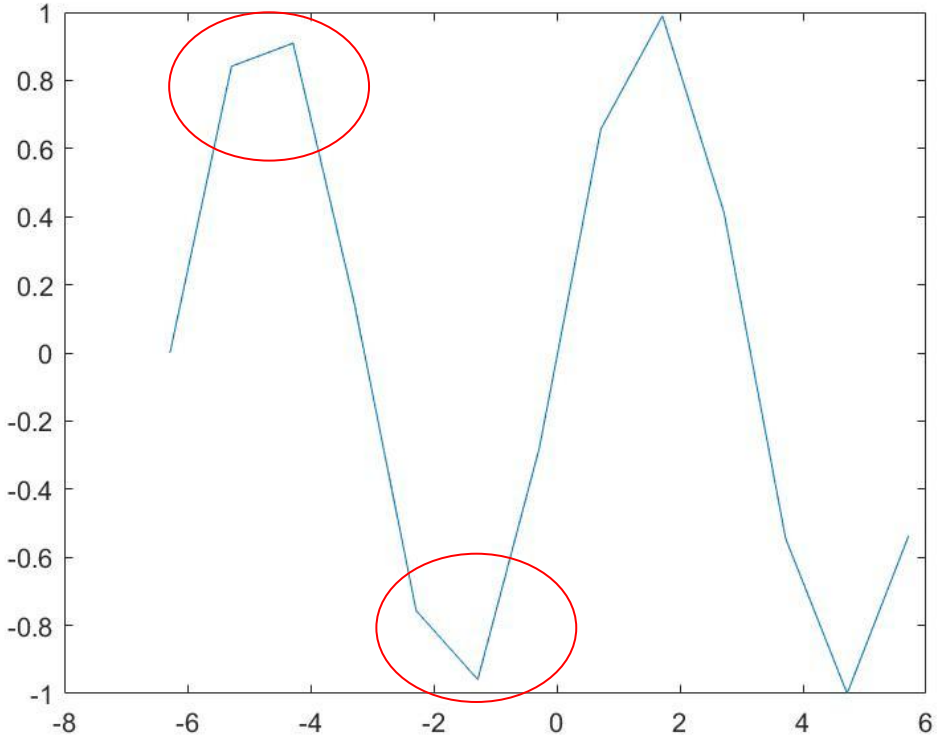
%% Plotting Intro Example- less # of points

% Domain says from -2*pi to 2*pi in increments of 1

X = -2*pi:**1**:2*pi;

Y = sin(X);

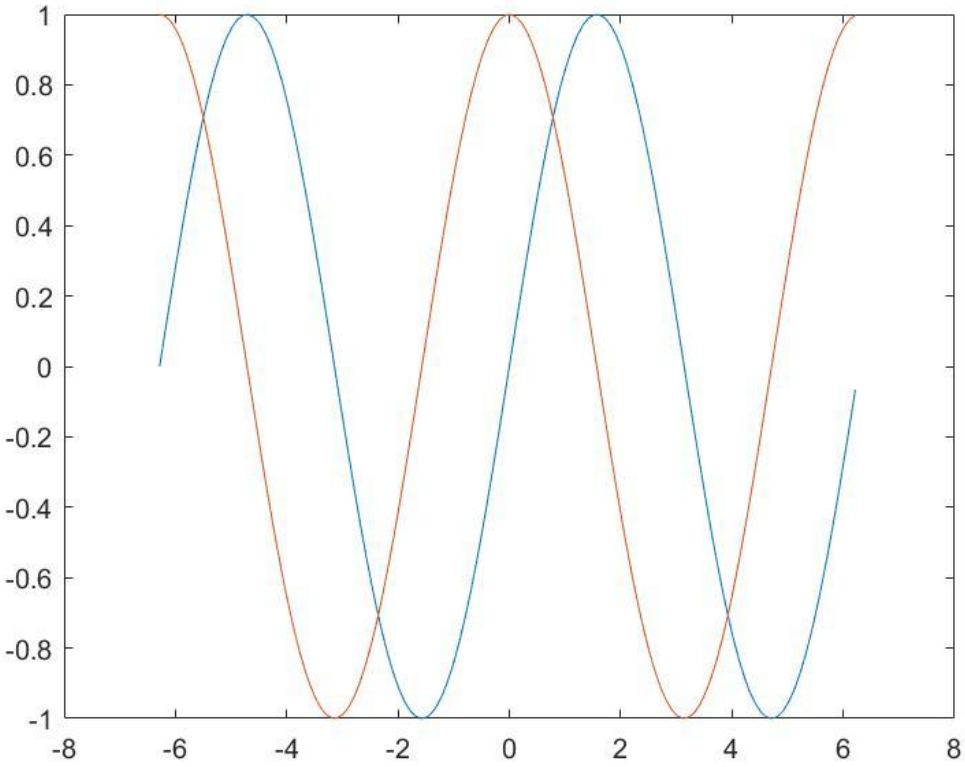
plot(X,Y)



%% Plotting Intro Example: Multiple lines in same graph

%2)Use the 'hold-on' function call to keep
% subsequent plot commands drawing into the same axis (without replacing the existing graph)

```
X = -2*pi:0.1:2*pi;
plot(X, sin(X))
hold on
plot(X, cos(X));
hold off
```



Annotating your Plot

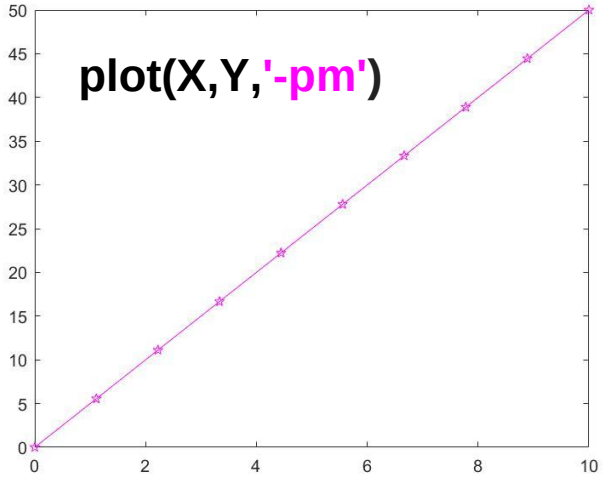
Annotation	Built in Function	Example
Add a title	title()	title('My Plot’)
Label the x-axis	xlabel()	xlabel(‘x’);
Label the y-axis	ylabel()	ylabel(‘y’);
Write text on plot	text()	text(x, y, 'text');
Add a background grid	grid()	grid('on');
Legend	legend()	legend('sin(x)', 'cos(x)');

Line Specs

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

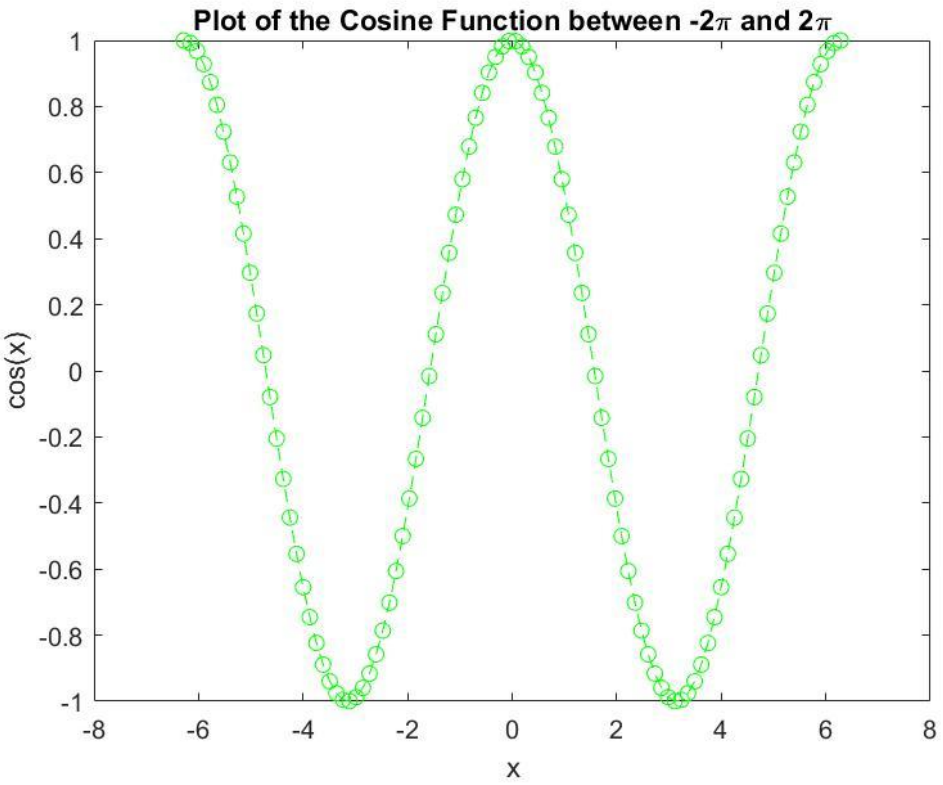
Marker Symbol	
+	Plus sign
o	Circle
*	Asterisk
.	Point
x	Cross
s	Square
d	Diamond
^	Upward triangle
v	Downward triangle
>	Right triangle
<	Left triangle
p	pentagram
h	hexagram (star)

Color	
r	Red
g	Green
b	Blue
c	Cyan
m	Magenta
y	Yellow
k	Black
w	White



Plotting Cosine Function

```
X = [-2*pi:0.1:2*pi];

Y = cos(X);

plot(X,Y,'--og')

xlabel('x')

ylabel('cos(x)')

title('Plot of the Cosine Function between -2\pi and 2\pi')
```

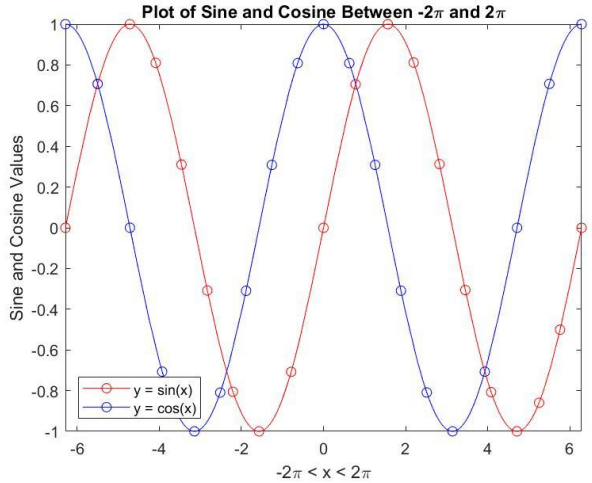


Plotting in MATLAB with fplot()

The **fplot(f, xinterval)** plots over the specified interval. Specify the interval as a two-element vector of the form [xmin xmax]. If not specified, the default interval is [-5, 5].
Anonymous function can be used to define equations.

- Benefits: eliminates the need to make a vector of domain (X) values and automatically adjusts the number of points graphed

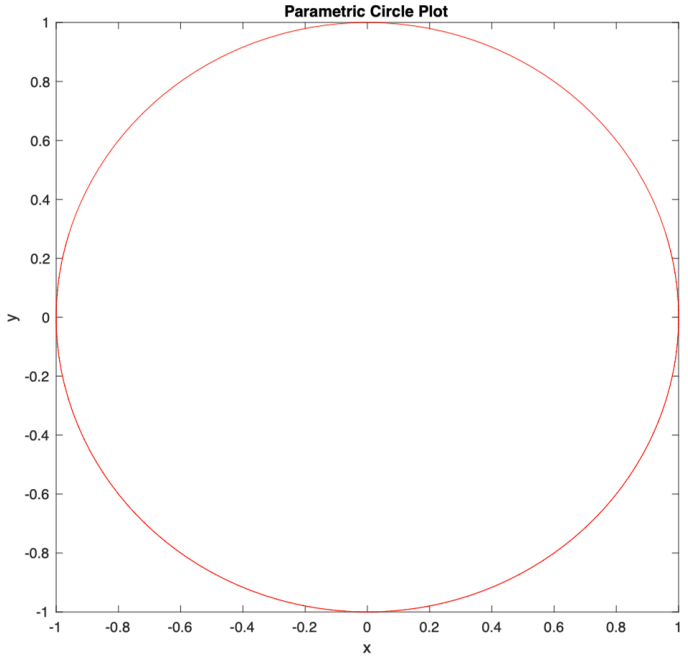
```
%% Fplot example

fplot(@(x) sin(x),[-2*pi,2*pi],'-or')
hold on
fplot(@(x) cos(x),[-2*pi,2*pi],'-ob')
hold off

title('Plot of Sine and Cosine Between -2\pi and 2\pi')
xlabel('-2\pi < x < 2\pi')
ylabel('Sine and Cosine Values')
legend({'y = sin(x)','y = cos(x)'},'Location','southwest')
```



Plotting in MATLAB with fplot()

The command **fplot(xfun, yfun)** plots a parametric function, which is composed of two different functions of a single variable.

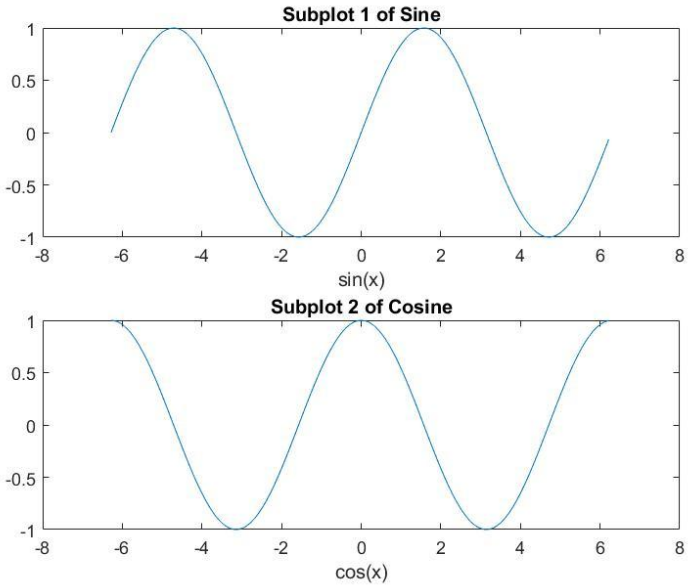
```
%% Parametric fplot
x = @(t) sin(t);
y = @(t) cos(t);
fplot(x, y, '-r')

title('Parametric Circle Plot')
xlabel('x')
ylabel('y')
```



Plotting in MATLAB with subplot()

The **subplot(r,c,p)** function creates a figure window with **r** rows and **c** columns of axes and creates an axes at position **p**, where **p** is the row-major index for the **r x c** positions in the figure.

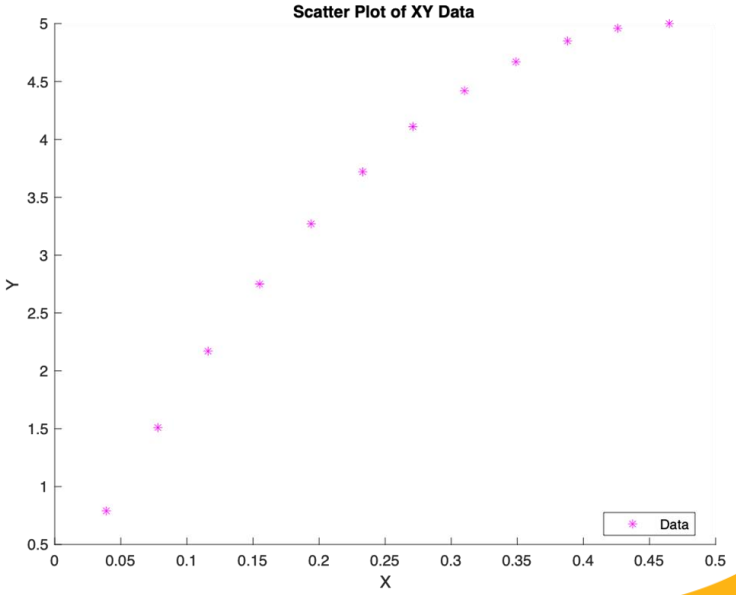
```
%% Subplot Example

%Create a figure with 2 axes (r), one (c)above the other
x = -2*pi:0.1:2*pi;

subplot(2,1,1)
plot(x, sin(x)); xlabel('sin(x)');
title('Subplot 1 of Sine')

subplot(2,1,2)
plot(x, cos(x)); xlabel('cos(x)');
title('Subplot 2 of Cosine')
```



Plotting in MATLAB with scatter()

The **scatter()** command plots a set of discrete points. The input is two vectors, one for each axis.

```
%% Scatter Example

X = [0.039 0.078 0.116 0.155 0.194 0.233 0.271 0.310 0.349 0.388 0.426 0.465];
Y = [0.79 1.51 2.17 2.75 3.27 3.72 4.11 4.42 4.67 4.85 4.96 5.00];

scatter(X,Y,'*m')

title('Scatter Plot of XY Data')
xlabel('X')
ylabel('Y')
legend('Data','Location','best')
```



Acknowledgement

The lab slides you see are not made by one person. All the TA/TFs served for this course have contributed their effort and time to the slides. Below are the leading TFs for each semester:

- 2021 FA - Karla Negrete (GTA)
- 2022 SP - Justin Grahovac
- 2022 FA - Kelli Boyer, Yisrael Wealcatch, Noelle Ray (GTA)
- 2023 SP - Mahamoudou Bah and Matt Moeller
- 2024 SP - Mohammad Riyaz Ur Rehman and Michael Mullaney