Week 7 Tutorial 1

The purpose of this tutorial is to demonstrate the basics of plotting data on a graph.

Note: One nice feature of live scripts is it allows you to use the "Run section" button or "Ctrl+r" or "Cmd+r" to will run each section (differentiated by the blue line) and you can see the plot change as we add more lines of code to give more detail to the plot. If you use the "Run section" button or shortcut, be sure to run each section, you cannot skip to a particular section without first running the prevoius sections as the setup variables won't exist.

NOTICE: Here I've added the `close all` command, you'll want to add this to all of your programs just like you do with `clear` and `clc`.

```
% Always clear workspace variables before a new tutorial or program.
clear
clc
close all % Close any currently open plot figure windows
```

Edit the code below and update the variable named **`name`** with your name for this tutorial in the code below.

```
name="";
fprintf("Output for Tutorial_07_1 run by %s.\n\n", name)
```

Given the following data, let's create a plot. We've created tables before using transpose and then displaying them with the `disp()` function.

```
% A matrix can be used to display a table of values
angles=0:30:360;
table=[angles' sind(angles)' cosd(angles)' tand(angles)'];

% Display the data using disp()
disp('  Angle(Deg)      Sine        Cosine        Tangent')
disp(table)
```

Creating the Plot

Let's create a plot instead. With plots we can display much more data so let's recreate the `angles` variable as a vector from 0 to 360 in increments of 1 degree.

```
% Plotting the results
clear angles % clears variable angles
angles=0:360;
```

To create the plot of data, we'll add the values for each dataset. We have three datasets so we'll have three sets of (x, y) data which will all be passed in with the format `plot(x1,y1,x2,y2,...,xn,yn)`. Now, you may notice we're reusing the `angles` variable for x in all three datasets and that's fine, each set of y values is computed with the same set of x values so in this case, they will be the same x values for all three.

```
figure(1)
```

```
plot(angles,sind(angles),angles,cosd(angles),angles,tand(angles))
```

Labeling the Plot

We could leave the plot as is, but we should always properly label plots, here we'll add a title, and labels for the x and y axes.

```
% Adds title, axis label, and gridlines
title('Trigonometric Functions')
xlabel('Angle(Degrees)')
ylabel('sin, cos, tan')
```

More Configuration Options

Now, let's add a little more detail to our plot to make it real professional looking.

To add a grid, use

```
grid on
```

To set the x and y scales, use

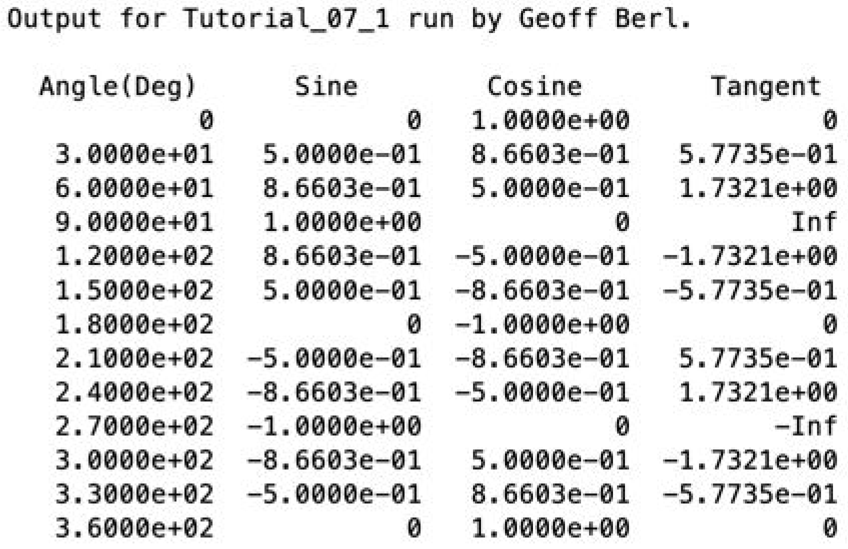
```
axis([0,360,-2,2]) % sets the xmin, xmax, ymin, ymax of a graph
```

To add a legend, use (in the same order as the dataset values)

```
legend('sin', 'cos', 'tan') % adds a legend to define each data set
```

Example Output:

Run this tutorial from the **Command Window** and ensure your output matches the following.



```
Output for Tutorial_07_1 run by Geoff Berl.

  Angle(Deg)      Sine        Cosine       Tangent
           0           0   1.0000e+00            0
  3.0000e+01  5.0000e-01   8.6603e-01   5.7735e-01
  6.0000e+01  8.6603e-01   5.0000e-01   1.7321e+00
  9.0000e+01  1.0000e+00            0          Inf
  1.2000e+02  8.6603e-01  -5.0000e-01  -1.7321e+00
  1.5000e+02  5.0000e-01  -8.6603e-01  -5.7735e-01
  1.8000e+02           0  -1.0000e+00            0
  2.1000e+02 -5.0000e-01  -8.6603e-01   5.7735e-01
  2.4000e+02 -8.6603e-01  -5.0000e-01   1.7321e+00
  2.7000e+02 -1.0000e+00            0         -Inf
  3.0000e+02 -8.6603e-01   5.0000e-01  -1.7321e+00
  3.3000e+02 -5.0000e-01   8.6603e-01  -5.7735e-01
  3.6000e+02           0   1.0000e+00            0
```

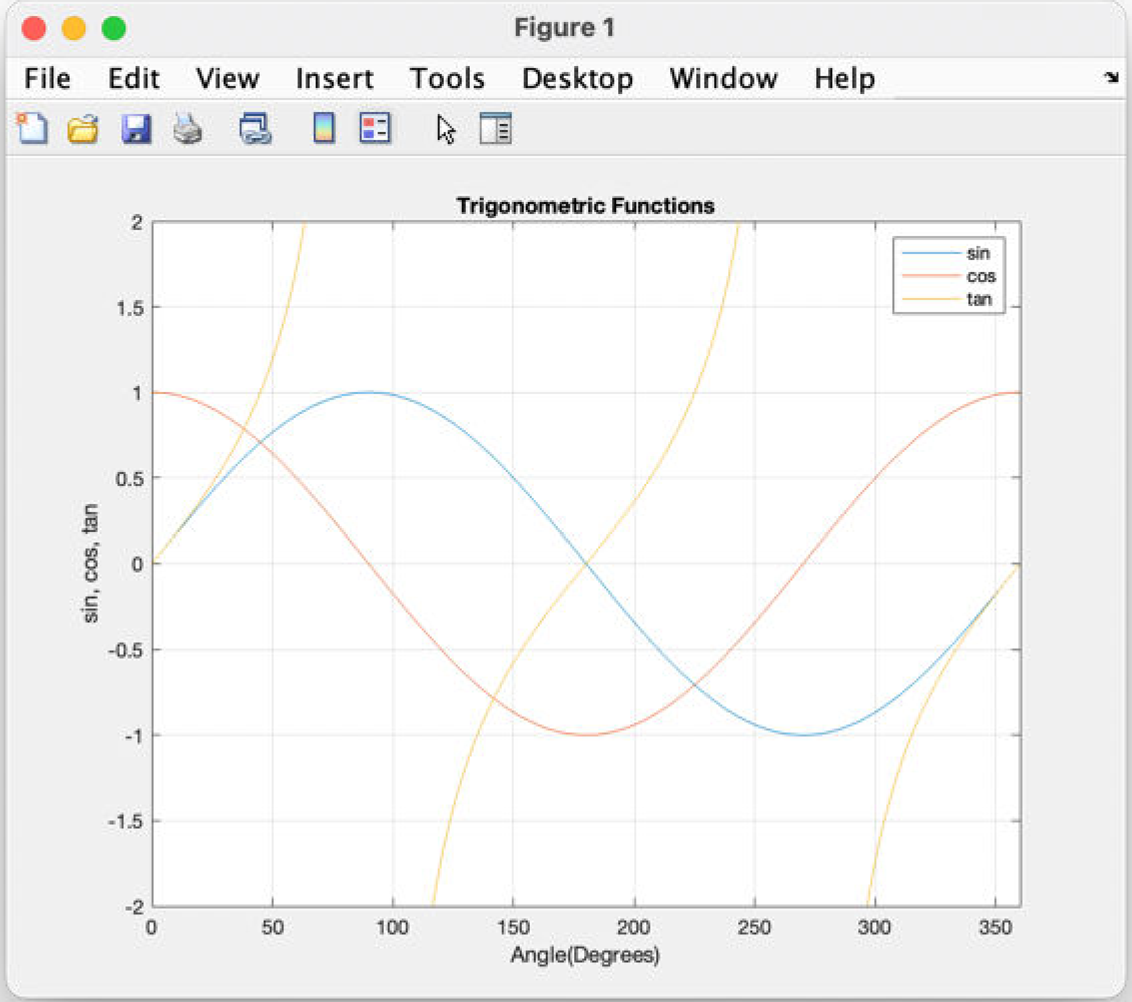


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Trigonometric Functions
sin
cos
tan
sin, cos, tan
Angle(Degrees)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
50
100
150
200
250
300
350