Lesson 05: Plotting

We've toyed a lot with how to perform calculations, but it's not awfully useful if we can't visualize them. Thankfully, MATLAB offers us a wide array of plotting functionality to help us visualize our data.

Graphic Objects

MATLAB organizes graphical objects into a hierarchical structure[1]:

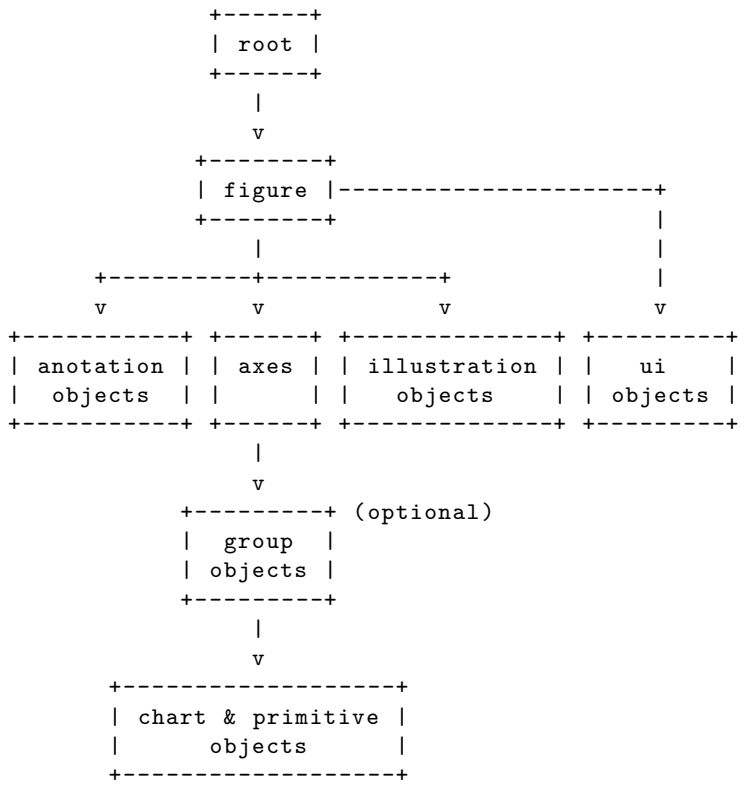


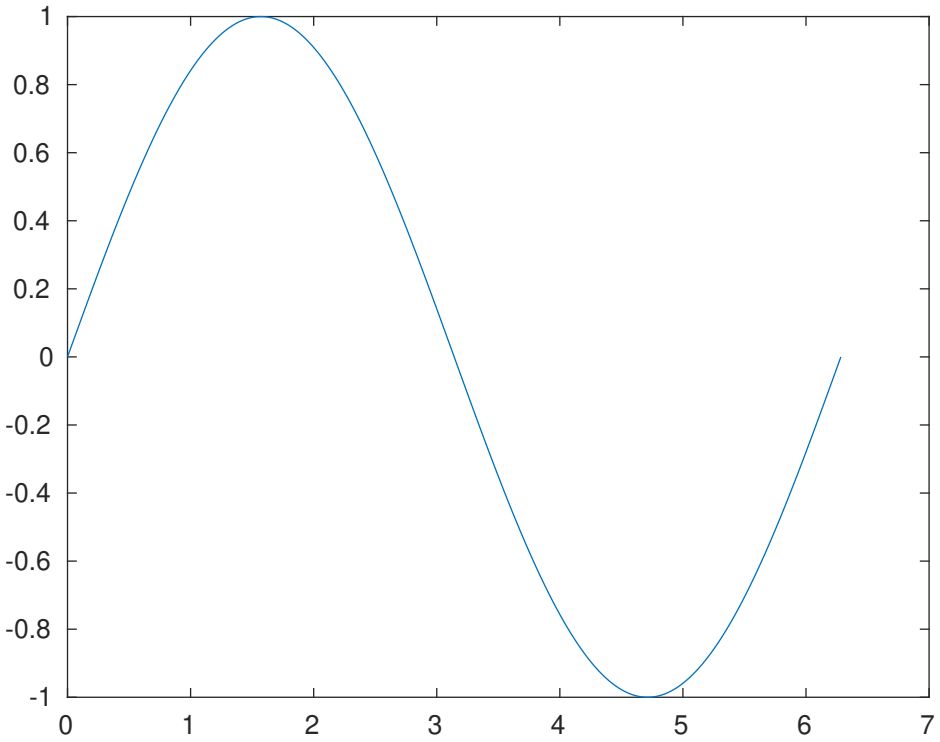
Figure 1: Organization of Graphics Objects

[1] `https://www.mathworks.com/help/matlab/graphics-objects.html`

2-D Plots

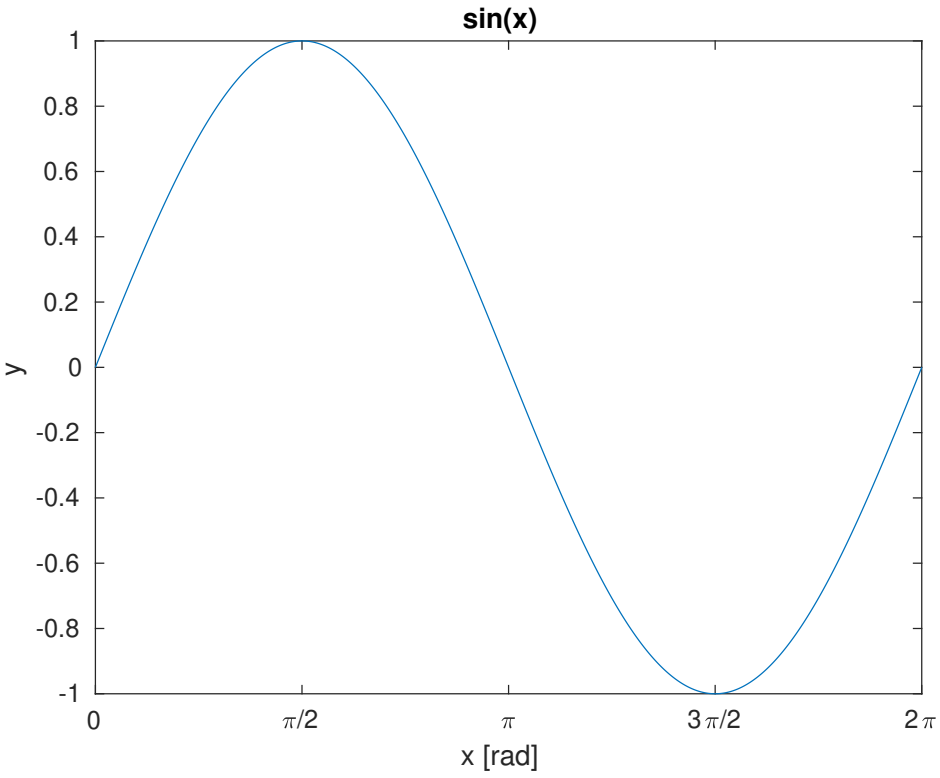
There's not much point in explaining. Let's jump right into an example.

```
x = linspace(0, 2 * pi, 1e3);
y = sin(x);

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



Here `figure` creates a new figure that `plot` can interact with. Although we successfully plotted our sine wave, it's quite ugly. Let's take the liberty to make it a little more pretty.


```
x = linspace(0, 2 * pi, 1e3);
y = sin(x);

figure;
plot(x, y);
title('sin(x)');
xlabel('x [rad]');
xlim([x(1), x(end)]);
xticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi'});
xticks([0, pi / 2, pi, 3 * pi / 2, 2 * pi]);
ylabel('y');
```

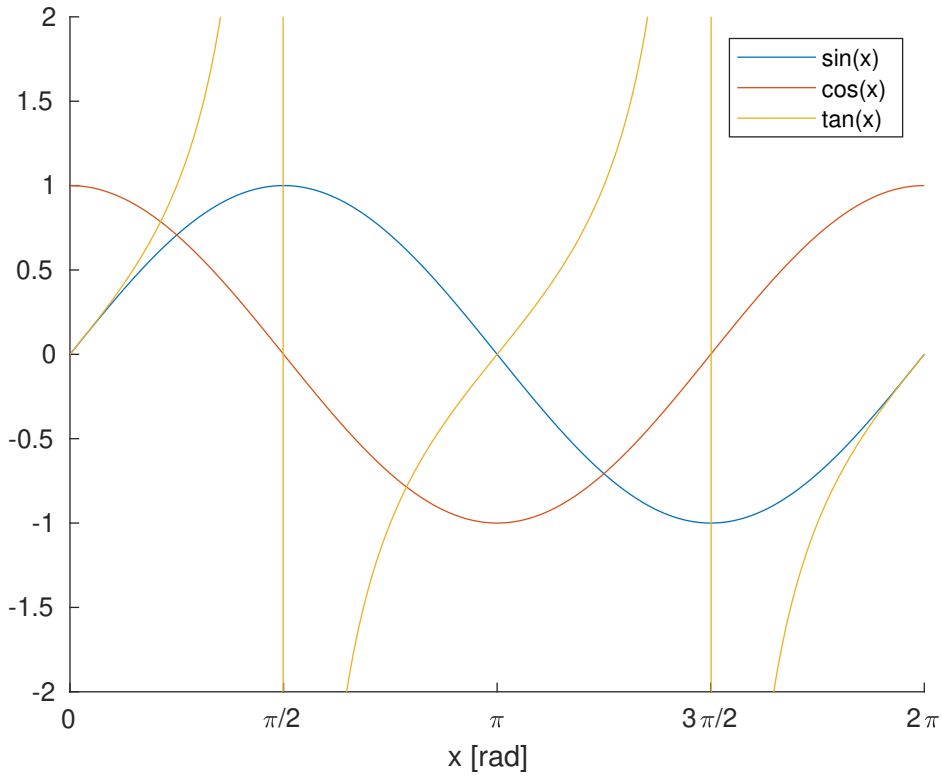


Retaining Plots

Say we wanted to plot multiple functions on the same plot:


```
x = linspace(0, 2 * pi, 1e3);
f = {@sin, @cos, @tan};

figure;
hold on;
for i = 1:numel(f)
    plot(x, f{i}(x));
end
xlabel('x [rad]');
legend('sin(x)', 'cos(x)', 'tan(x)');
xlim([x(1), x(end)]);
xticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi'});
xticks([0, pi / 2, pi, 3 * pi / 2, 2 * pi]);
ylim([-2, 2]);
```

Aside— `hold on` can be a back stabber! Sometimes you may experience buggy behavior if you call it before the first plot.

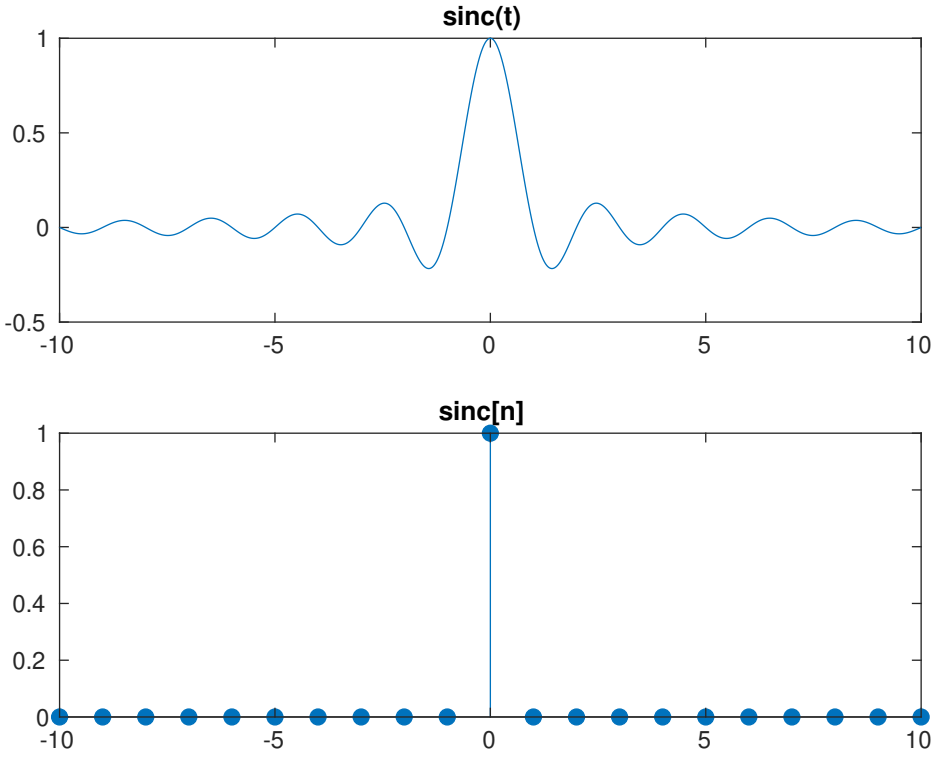


Subplots

We can have multiple plots in a figure by doing the following:

```
t = linspace(-10, 10, 1000);
n = -10:10;

figure;

subplot(2, 1, 1);
plot(t, sinc(t));
title('sinc(t)');

subplot(2, 1, 2);
stem(n, sinc(n), 'filled');
title('sinc[n]');
```

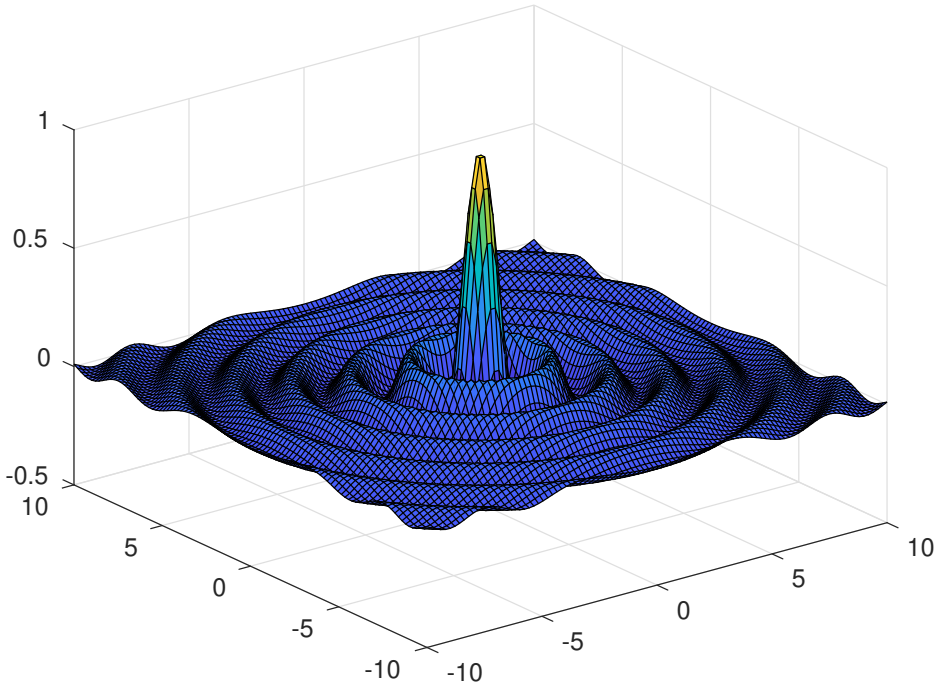
Aside— Notice the zero-crossing property of the sinc function!



3-D Plots

MATLAB is also capable of making 3-D plots:

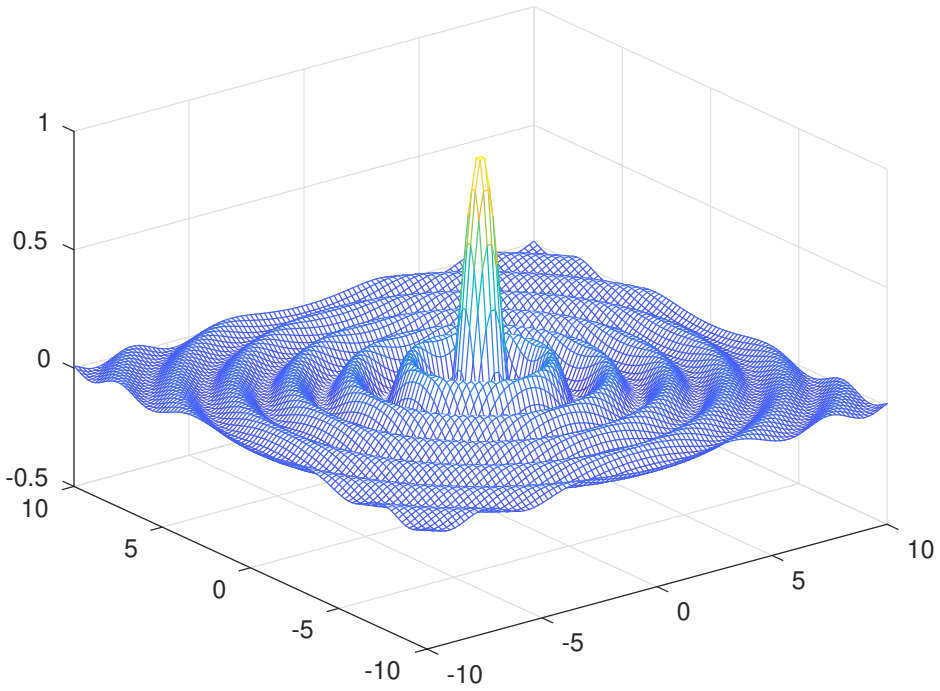
```
x = linspace(-10, 10, 100);
y = x;

[X, Y] = meshgrid(x, y);
R = sqrt(X.^2 + Y.^2);

figure;
surf(X, Y, sinc(R));
```



You can think of `meshgrid` as the Cartesian product of the two input vectors split over two output matrices. Having these matrices allows us to evaluate functions over a 2-D plane.

We can also use a mesh instead:

```
figure;
mesh(X, Y, sinc(R));
```



More Plotting Functionality

This barely scratches the surface of MATLAB's extensive plotting functionality. Once again, the online documentation is a great starting point for the interested reader: `https://www.mathworks.com/help/matlab/2-and-3d-plots.html`

