Ph3 Mathematica Homework: Week 2

Eric D. Black
California Institute of Technology

v1.0

1 Plotting

1.1 The Plot command

To plot a basic function, use the `Plot` command. This command takes a minimum of two options, the function you want to plot and the range over which you want to plot it.

Exercise 1: Plot a sine function over one period.

```
Plot[Sin[x], {x, 0, 2*Pi}]
```

Set the horizontal and vertical ranges of the plot with the `PlotRange` option. Make the arrow just like it looks, by typing a minus sign followed by a greater-than symbol. A single pair of number sets the vertical range. Type this in and execute it.

```
Plot[Sin[x], {x, 0, 2*Pi}, PlotRange->{-2, 2}]
```

Set both the horizontal and vertical ranges like this. Type this in, execute it, and see if the result is what you expect.

```
Plot[Sin[x], {x, 0, 2*Pi},
 PlotRange -> {{-Pi, 3*Pi}, {-2, 2}}]
```

Exercise 2: Label the axes with `AxesLabel`.

```
Plot[Sin[x], {x, 0, 2*Pi},
 PlotRange -> {{-0.5*2*Pi, 1.5*2*Pi}, {-2, 2}}, AxesLabel->{"x", "y"}]
```

Dress your plot up with a frame and more elaborate labels.

```
Plot[Sin[x], {x, 0, 2*Pi},
 PlotRange -> {{-0.5*2*Pi, 1.5*2*Pi}, {-2, 2}}, Frame->True,
 FrameLabel->{"phase", "amplitude"}]
```

There are *a lot* of options you can use with `Plot`, but these will do for now.

1.2 Plot3D

Here's a cute one you should know about. We won't use it much (if at all) in this class, but I can't leave this topic without showing it to you.

Exercise 3: Type the following into a cell, and run it.

```
Plot3D[Cos[Sqrt[x^2 + y^2]]*Exp[-Sqrt[(x^2 + y^2)]/5], {x, -5*Pi,
  5*Pi}, {y, -5*Pi, 5*Pi}, PlotPoints -> {100, 100}, PlotRange -> All]
```

Once you have your plot, click and drag on a corner to rotate it around.

1.3 VectorPlot

Another one I can't resist showing you is `VectorPlot`. Just like it sounds, this one plots vector fields. If for example you want to plot the function

$$-y\hat{\imath} + x\hat{\jmath}$$

from -5 to 5 on both axes, you would type

```
VectorPlot[{-y, x}, {x, -5, 5}, {y, -5, 5}]
```

Exercise 4: Plot the following vector fields using `VectorPlot`.

1. $x\hat{\imath} + y\hat{\jmath}$

2. $y\hat{\imath} + x\hat{\jmath}$

3. $-x\hat{\imath} + y\hat{\jmath}$

1.4 ListPlot

Up to now we've been plotting mathematical functions, but in this class (and in experimental science in general) you will plot lists of data far more often. The command to do this is `ListPlot`, and its argument is, not surprisingly, a list.

Exercise 5: Remember last week, when you generated a list of the first ten integers and their square roots? Plot that list using `ListPlot`. If you named your list `firstten`, the syntax would be

```
ListPlot[firstten]
```

Note that you do not have to specify the domain (the range of the x-values), as you would with `Plot`, since that information is contained in the list. All of the other options you learned for `Plot` will also work with `ListPlot`, *e.g.* `PlotRange`, `AxesLabel`, etc.

1.5 Multiple plots: the Show command

You can overlay multiple plots on the same graph using the `Show` command. Its arguments are plots.

Exercise 6: Put the graphs of the square-root function and your list on the same plot using the following command.

```
Show[ListPlot[firstten], Plot[Sqrt[x], {x, 0, 10}]]
```

1.6 Storing plots in variables

Remember how I told you you could store either numbers or lists in variables? Well, you can store plots too. In fact, you can store just about anything in a variable. The last exercise would work just as well if it were written as

```
a = ListPlot[firstten]
b = Plot[Sqrt[x], {x, 0, 10}]
Show[a, b]
```

Storing plots in variables gives you the option of calling them back up later without having to reevaluate (or retype) them. It's a useful practice, and we'll revisit it more later.

1.7 Plotting data with error bars

`ListPlot` will accept arguments other than simple numbers. If you give it data with uncertainties using `Around`, as we discussed last week, it will automatically include and display those error bars.

Exercise 7: Make the plot from Taylor, Exercise 2.21a, in Mathematica.