Physics 309
Meeting *Mathematica*

Introduction

I've designed these exercises to help you become comfortable loading *Mathematica*, saving files, entering *Mathematica* commands, asking the program for help, and performing a few other basic chores.

Be careful to type in each command exactly as written here. In particular, be very careful to type in precisely corresponding capital and lowercase letters, special symbols, and space between characters.

As you work these exercises, keep your brain engaged! It's especially important at this early stage for you to *pay obsessive attention to the syntax of each command*. As you work, think about how *Mathematica* responds to your commands. Identify common elements of *syntax* in the commands. If you don't know what a command does, ask *Mathematica* using the on-line help. If something doesn't work, ask a fellow student or ask me!

Feel free to play around. Once you've executed and deciphered each command, try changing one or more arguments to produce different output. Or try using previous commands you've already explored together with the one you're working on at the moment. That's the best way to learn *Mathematica*.

First Exercises with *Mathematica*

1. It's easy to make *Mathematica* generate a list of numbers (or functions). Try

   `Table[ n, {n, 2, 17}]` and `Range[2,17].`

2. You can make *Mathematica* simplify complicated expressions, without making algebraic errors. Try

```
Simplify[(x + y)(x^2 - x*y + y^2)]
```

3. You can use *Mathematica* to evaluate definite and indefinite integration quickly and accurately. Try

```
Integrate[ x Cos[x^2], x]
Integrate[ x Cos[x^2], {x, 0, 4}]
N[ Integrate[ x Cos[x^2], {x, 0, 4}] ]
```

Note that *Mathematica* gives you an exact answer for the definite integral unless you ask it for a numerical value, using `N`.

4. *Mathematica* will even help you do double integrals. To evaluate

$$\int_0^{\pi/6}\int_0^{\pi/2}\left(y\sin x - x\sin y\right)dx\,dy$$

enter this command

```
Integrate[ y Sin[x] - x Sin[y], {x, 0, Pi/6}, {y, 0, Pi/2}]
```

Make *Mathematica* tell you the numerical value for this double integral.

5. You can use *Mathematica* to take the drudgery out of solving all sorts of equations—including transcendental equations. The first step in solving the transcendental equation

$$\cos x - x = 0$$

using *Mathematica* is to determine an initial guess at the solution by drawing a graph.

(a) First the graph

```
Plot[ Cos[x]-x, {x, 0, 1}]
```

(b) Now we look at the graph to determine a root—a value of x at which the left-hand-side of the equation is zero. Let's choose $x = 0.5$ as our initial guess. We tell *Mathematica* to find the solution of the equation (its root) using `FindRoot`:

```
FindRoot[ Cos[x] == x, {x, 0.5}]
```

(c) Using the graph, can you suggest a better initial guess than x = 0.5? Try your guess in `FindRoot`.

6. You can use *Mathematica* to evaluate infinite summations, analytically and numerically. Try

```
Sum[1/i^3, {i, 1, Infinity}]
Sum[1/i^3, {i, 1, Infinity}] // N
```

Note that the second command gets a numerical value by "piping" the output of the `Sum` command into the "number" command `N`. You can pipe the output of most *Mathematica* commands into most other *Mathematica* commands. This tactic makes it easy to understand your notebooks (and avoid errors.)

7. *Mathematica* makes it easy to apply functions to several arguments. We arrange the arguments into *Mathematica* expressions called lists and feed these lists into *Mathematica*'s built-in functions. Watch what happens when you try

```
Sqrt[{a, b, c, d}]
Expand[{(a + b)^2, (a + b)^3}]
D[{x^2, x^3, Exp[-x], Sin[x]}, x]
```

This amazingly powerful feature is called listability.

8. *Mathematica* has awesome three-dimensional graphics capabilities. Type in the following command to see a three-dimensional parametric plot of a familiar figure:

```
bagel = ParametricPlot3D[
      {Cos[s] (2 + Cos[t]), Sin[s] (2 + Cos[t]), Sin[t]},
      {s, 0, 2Pi}, {t, 0, 2Pi},
      Boxed->False, Axes->None]
```

Notice that I named the plot bagel.

This lets me use the plot in subsequent commands. For instance, to get a different view of our bagel, try

```
Show[bagel, ViewPoint->{3,1,1}]
```