Physics 651: Assignment 0

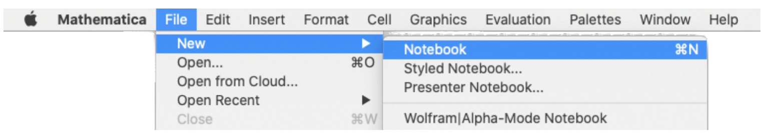
(to be submitted by Tuesday, September 3, 2024, but ungraded)

I invite you to attempt Assignment 0 and to turn in your work for Questions 4, 6, and 8. Your submission should take the form of a single Wolfram Notebook, sent as an attachment to `kbeach@olemiss.edu`. Please follow the naming convention `Phys651-A0-webid.nb`, and be sure to include the subject line `Phys651-Fall2024-webid Assignment 0 Submission`. You are welcome to annotate your Notebook in whatever way you think would be helpful to me; I suggest including a Format > Style > Section marker to indicate the start of each question and using the Format > Style > Text mode to add other descriptive comments.

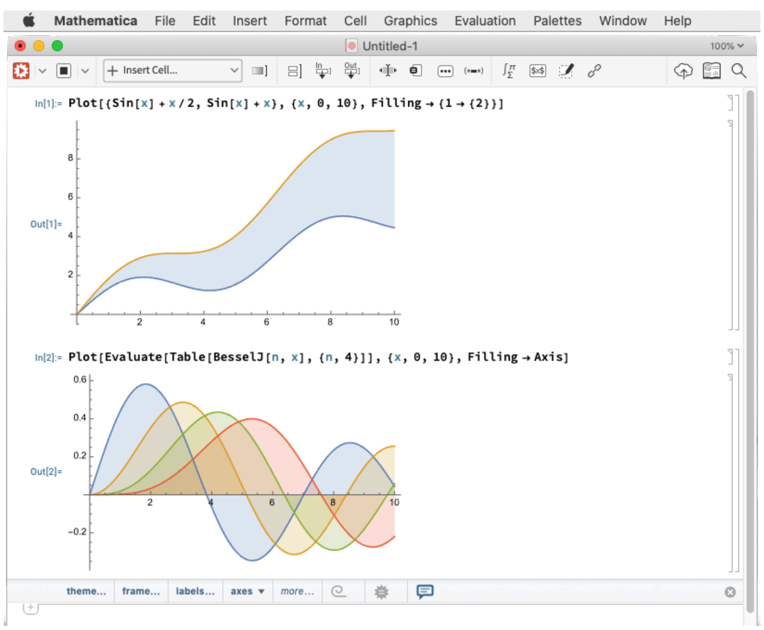
1. Installation instructions for *Mathematica* are provided on the class website.

2. Start up the *Mathematica* front-end application. This is the interface that sits atop the *Mathematica Kernel*, which is the underlying computational engine.

3. Open a new *Notebook*.



4. Try creating some simple graphics with the `Plot` function. Note that pressing enter or return on your keyboard will produce just a bare line-feed/carriage-return. To get *Mathematica* to interpret and execute a line of code, you will have to type shift + return.



5. If you are completely new to this, have a look at the videos Getting Started in Mathematica and 2D Plotting. There is also a gallery of screencasts on the Wolfram website and a large collection of course videos at Wolfram U.

6. Note that while *Mathematica* supports a more conventional procedural style (what you're used to in python, C++, etc), it's really designed for so-called functional programming. This summary page might be useful for you. Functional programming is powerful, expressive, and compact. It's sometimes hard to parse if you're not used to it. For instance, the following produces the bifurcation diagram of the logistic map:

```
Logistic[x_, r_] := r x (1 - x)
LogisticLast[x0_, r_, iters_] := Nest[Logistic[#, r] &, x0, iters]
ListPlot[Table[LogisticLast[k/7.0, #, 1000] & /@ (Range[400]/100.0), {k, 1, 6}]]
```

In a few weeks, this will look much less mysterious!

7. Here we plot two polynomials $f(x)$ and $g(x)$, each of order 3, that have roots at $x = 2, 5, 6$ and $x = 2, 3, 6$, respectively.

```
f[x_] := (x - 2) (x - 5) (x - 6);
g[x_] := (x - 2) (x - 3) (x - 6);
Plot[{f[x], g[x]}, {x, 1, 8}, Frame -> True,
 FrameLabel -> {"Independent variable," x, "Dependent variable"},
 LabelStyle -> Directive[Larger, Darker[Green, 0.8]],
 PlotLabel -> "Roots of two functions compared",
 PlotLabels -> Placed[Automatic, Right]]
```

Observe that function arguments in *Mathematica* are enclosed in (square) brackets rather than (rounded) parentheses, as in conventional mathematical notation. Lists and sets are delineated by (curly) braces. The trailing underscore on a variable is necessary when defining a pattern; e.g., `f[x_]` indicates that a value for `x` will be provided when the function `f` is called, according to what appears on the right-hand-side of the `:=` assignment operator. Also, it suffices to write `(x - 2)(x - 5)(x - 6)` for the product of monomials; we don't need a connective asterisk, i.e., `(x - 2)*(x - 5)*(x - 6)`, as in most other programming languages.

8. Consider the 4th order polynomials $p(x) = \prod_{j=1}^{4}(x - 2j + 1)$ and $q(x) = \prod_{j=1}^{4}(x - 2j)$, having roots at $x = 1, 3, 5, 7$ and $x = 2, 4, 6, 8$. Make a plot showing the two functions and their product (drawn as three distinct lines, properly labelled). Then compute the integrals

$$\int_0^9 dx\, p(x), \quad \int_0^9 dx\, q(x), \quad \int_0^9 dx\, p(x)q(x), \quad \text{and} \quad \int_0^9 dx\, \frac{p(x)q(x)}{\prod_{j=1}^{8}(x - j)}.$$

You might find it helpful to read the instructions brought up by the prompts `?Integrate` and `?Product`.