Spring, 2017

Physics 6810: Assignment #2

This assignment is a follow-up to Sessions 3 and 4. It is intended to give you additional practice in the basic tasks from class (like coding algorithms), that we'll need repeatedly and build upon, and to verify to me that you can do them. Please ask questions! It is due at the end of the day (midnight) on Friday, February 17.

Please use BuckeyeBox to "hand in" the assignment. Put your homework inside a subfolder of your main folder named `PS_2` or `Assignment2` or similar.

Include in your repository makefiles (one for each program), C++ programs (with the answers to any questions in the comments at the top) and a postscript file (other formats are also ok) of the log-log plot (made with an included gnuplot plot file). *It is required that your code have appropriate comments.* Comment your codes with your name, email, AND revision history, as in the example codes from class, as well as appropriate comments answering questions and explaining the code.

The problems here are parts of the same computational problem of calculating integrals. You can combine them in any way that is convenient (i.e., you don't need separate main programs for each sub-section). Note that the session notes have lots of help info and that a sample code for GSL integration is included in the session 4 zip archive. *Check the 6810 webpage for suggestions and hints.* Please ask questions and give feedback early and often.

1. **Follow-up/Completion of Session 3**

   (a) Write routines to do integration of an integrand function (specified in the code by the user) over a specified (finite) interval in *double precision* using Simpson's rule, the Milne integration rule (see the Session 3 notes), and an *appropriate* integration routine from the GSL. (You may choose to base your Simpson's rule routine on the one from class. Check the hints for other suggestions.) Test your routines with a non-trivial test integral (*choose an integrand we haven't used and that is much more complicated than a polynomial over your integration range*). The test program and routines should be in separate files and you should create a makefile to compile them.

   (b) Extend your program to generate an appropriate data file and use it to make a plot (with a gnuplot plot file that makes a color postscript file) to do an error analysis of the integration methods from the first part. (This includes fitting and discussing slopes in different regions.)

(c) Find *graphically* (i.e., by *looking* at the plot from the last section) the optimum number of points to use for the Milne integration rule in double precision and explain how this makes sense analytically. That is, build on the discussion of errors in the notes to make a *quantitative* estimate of what the optimum number should be and compare to your empirical result. (Remember that you shouldn't expect perfect agreement.)

(d) (*BONUS to get a plus*) Evaluate one of the singular integrals from the handout (1094 Session 4) directly using Simpson's or Milne and a GSL routine, and then compare to the answer found by applying a method discussed in the handout or session notes (or in Numerical Recipes). Briefly discuss your results and why the method works.

(e) (*Alternative BONUS to get a plus*) **Empirical error analysis.** In real-life research problems, we may not have an exact answer to test our code with. How could you analyze your integration results if you didn't know the exact answer? Apply the method from the section on "Empirical Error Analysis" in the Session 4 background notes to analyze one of the integration rules (with your choice of rule and a non-trivial choice of integral!) to find the approximation error.