Physics 250 Computational Methods in Physics 2009
Homework 2 ALL (both) problems due this time, Monday January 26 2009

1) In class we talked about the use of the midpoint integration routine for a variety of functions and made plots of how the error in these integrals when to zero with increasing numbers of points. Do this again, for yourself using my program (posted on blackboard) for the integral of $x^{-1/2}$ over two ranges, $x_1 = 0, x_2 = 1$ and $x_1 = .03, x_2 = 1$. Note that this function is infinite at the edge of the range so that neither the Riemann sum nor the trapezoid rule will work, so you will need to modify my m file to disable them. Use 2^n points in this integration for $n = 1,2,3,4,5,6,7$ and make (appropriately labeled) plots of the base 2 logarithm of the error in the calculated integral as a function of n. Discuss these results, as appropriate, noting that the function does not have a finite derivative, at least in some of the range of one of these integrals. Put the results in the digital drop box. The preferred format in the digital drop-box is a "zipped" file/folder with the title Your_name_#.zip where Your_name is (well) your name and # is the homework number (in this case 1). Put appropriate m files and files containing plots in a zip file / zip directory. Multiple zip files (per homework) are allowed, but discouraged.

2) The MatLab programs in Garcia's book are available freely at http://www.mathworks.com/matlabcentral/fileexchange/2270. Either by downloading these programs or by typing them in get the interpolation programs interp and interpf. Read Garcia's discussion of these ideas and programs. Modify interp so that it, instead of inputting data from the keyboard it takes an integer number of points $n_{points} > 3$, from a range $x_{\min} < x < x_{max}$ from a known function, func (remember what we did in class for integration). This should result in points at $x_i = x_{\min} + (i-1)(x_{\max} - x_{\min}) / (n_{points} - 1)$ with $i = 1,2,3,...n_{points}$. Then, arrange interpf so that it plots *both* the interpolated value of the function and the *difference* between the "actual" curve of the real function and the interpolated curve, at all the "actual" points, and at n_{interp} points between them. Arrange this so that the

interpolated function and the difference are on *different* plots. For any x base the interpolated curve on the points of the function at. x_i, x_{i+1}, x_{i+2} where you choose i so that $x_i < x < x_{i+1}$. For simplicity, when you are between the last two points e.g. when $i = n_{points} - 1$, use the value of the "real" function at $x_{n_{points}+1} = x_{\min} + (n_{points})(x_{\max} - x_{\min}) / (n_{points} - 1)$ as the last point used by the interp program. Having written this program, run it for the sine function with $x_{\min} = 0, x_{\max} = \pi$, $n_{interp} = 3$ and $n_{points} = 3, 10, 30$. Remark on what you would think being presented with these plots. How quickly do you think the error in the interpolation routine goes to zero with increasing n_{points}? Make a guess for smooth functions e.g. based on a Taylor expansion, the first term that seems to be necessarily left out and how the expected size of that term depends in n_{points} (which it does indirectly.) Turn in the programs / m files containing your subroutines and appropriately label plots of the output to the digital drop-box, as discussed in problem 1. The preferred format in the digital drop-box is a "zipped" file/folder with the title Your_name_#.zip where Your_name is (well) your name and # is the homework number (in this case 1).