ASSIGNMENT 3, PHYS 3437

1. Obtain a copy of `hermstart.f` from the website. This program performs an Hermite interpolation on the Bessel function $J_1(x)$, and writes two datafiles to disc: `herraw.dat` and `herint.dat`. The former contains the raw data for $J_1(x)$ as given in the data statements near the beginning of the program, and the latter contains the Hermite-interpolated data on a grid ten times finer. Compile the code, run it, and within gnuplot, issue the following command:

```
plot [0:10] "herraw.dat" u 1:2 w lines, "herint.dat" u 1:2 w lines, 0 w lines
```

It should be obvious which plots represent the raw and interpolated data. You should also note that at the location of the raw data points, the interpolated plot is smooth because of the requirement of the Hermite interpolation scheme that the first derivative be continuous. In case you are wondering, the first derivatives are computed from the functions $J_0(x)$ and $J_2(x)$ (also tabulated in the data statements of `hermstart.f`) using the well-known recursion relation for Bessel functions:

$$J_1'(x) = \frac{J_0(x) - J_2(x)}{2}. \tag{1}$$

Note that the first root of $J_1(x)$ after $x = 0$ lies somewhere between $x = 3$ and $x = 5$. By importing your hybrid bisection/secant root finding subroutine from Assignment 2 into `hermstart.f` and making any other changes that may be necessary, modify `hermstart.f` so that you can find the first root past $x = 0$ to six significant figures.

For this problem, email me a copy of your modified hermite interpolation program (it's probably best to do this along with the answers to the other questions) and on paper provide the first root for $J_1(x)$ past $x = 0$ to six significant figures with an error estimate. Does this result necessarily give the actual root of $J_1(x)$ to within the error quoted? Why or why not?

2. Write a program to solve an $n \times n$ tridiagonal system of equations, where n can be specified by the user. The guts of a tridiagonal solver are in the subroutine `trisolve` which is on the website - you should feel free to use this subroutine. You will have to devise your own way of allowing the user to input the actual matrix and source terms and reporting the final results.

Use your program to solve the following system of equations:

$$\begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}. \tag{2}$$

For this problem, again email a copy of your solution and on paper provide the solution to equations (2) above as determined by your solver (*i.e.*, values of x_1, x_2, *etc.*).

3. Write a cubic spline interpolation program which allows the user to choose between a natural and clamped spline. While there are a number of differences between the Hermite and cubic spline schemes, the program `hermstart.f` for problem 1 will still be useful as a possible template for the layout of your program.

a) Starting with the data in the data statements of `hermstart.f`, plot $J_1(x)$ for $0.0 \le x \le 10.0$ for $\Delta x = 0.1$ using the clamped spline. Values for the derivatives $J_1'(0.0)$ and $J_1'(10.0)$ can be computed from equation (1) above and the data for $J_0(x)$ and $J_2(x)$ in `hermstart.f`.

b) Repeat part a), this time creating six plots (all plotted on the same graph) with $J_1'(0.0) = -2.0$, -1.0, 0.0, 1.0, 2.0, and 3.0, with $J_1'(10.0)$ the same as in part a). Can you see what causes the "ringing" in the interpolations? That is, how do the "bogus" values for the derivative specified at the left point of the left-most interval corrupt the interpolation into other intervals?

c) Now plot $J_1(x)$ for $0.0 \le x \le 10.0$ and for $\Delta x = 0.1$ using the natural spline.

d) What is your best advice to the novice if they don't know the derivatives at the end points: to use the natural spline, or to use the clamped spline with "stabs in the dark" for the first derivatives at the end point(s)? Conversely, if you do know the derivatives at the end points, does it make *that* much difference to use the natural spline instead [at least for $J_1(x)$]?

e) Why do you suppose it is called the "natural" spline? That is, what is so natural about setting the second derivatives to zero at the end points?

For this problem, email me your cubic spline interpolator, and provide all graphs and written answers to the questions from parts a) to e) on paper.