Assigned: Monday, March 31, 2014
Due: Week of April 7, 2014

MTH 371: Group Project 2
Spline Interpolation

GENERAL GROUP PROJECT GUIDELINES:

- Group project assignments should be a collaborative effort. All should participate in discussion and solution writing.
- Each week, your group must meet with Dr. Vidden to discuss your findings. All members must be present. Your grade will be assigned at the end of the meeting.
- Each student should keep group project solutions in a dedicated notebook. Bring this notebook to your weekly meeting to discuss your findings. For coded solutions, bring a laptop to your weekly meeting. Have the laptop ready before the start of the meeting.

1. Write a Scilab `.sci` file for the following functions.
   (a) `x = trisolver(u,h,b)`

   This function solves the system $A\vec{x} = \vec{b}$ where A is a symmetric tridiagonal matrix with main diagonal `u`, lower diagonal `h`, and upper diagonal `h` where `u,h` are vectors of appropriate length. Code this function to be as efficient as possible. That, as we discussed in class, modify ideas from Gaussian elimination to solve special systems of this form.

   (b) `P = natCubicSpline(x,f)`

   This function computes the natural cubic spline approximation to a function using equally spaced intervals. `x` gives the interpolation nodes and `f` gives the function values at these nodes. The returned array `P` contains the Scilab cubic polynomial for each interval. The above `trisolver` function should be used.

2. Write a Scilab `.sce` file which uses the scripts from problem 1 to compute the natural cubic spline approximation of function $f(x) = \sin(x)\sin\left(\frac{2x+5}{10}\right)$ on the interval $[-5, 5]$ using nodes $-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5$. Plot this function and your spline approximation on the same graph. Clearly indicate the location of the interpolation points. Label your graph and include a legend.

3. BONUS QUESTION!
   (a) Repeat question 2 using a complete cubic spline. You can use the `trisolver` function from 1, but you will need to write a new function, `compCubicSpline`.
   (b) Use the Scilab function `splin` to plot as many types of splines (natural, complete, and more!) as you can. Do this again for problem 2. Plot all approximations of the same graph. Research the background and purpose of each type.

4. BONUS QUESTION!
   Create a random set of function values using the Scilab command `[x,f]=rand(100,2)`and fit a natural cubic spline to these 100 points. Feel free to modify code from problem 1 to do this. Plot these points and your spline approximation together.