Math 297 **Homework 4** **Due Oct. 23, 2008**
Grady Wright

1. In this problem you will use radial basis function (RBF) interpolation with thin plate splines to reconstruct the topograhy of a region from scattered topographic samples. Download the MATLAB m files `rbffit` and `rbfval`, and the data file `topodata.mat` from the course web page and save them to your working directory.

   The data file contains 1000 topographic samples of a two-ravine drainage from upper state New York. To view the raw data issue the following commands:

   ```
   load topodata;
   scatter3(x,y,z,40,z,'.');
   colorbar
   ```

   Assuming that x and y are the independent variables and z (the elevation) is the dependent variable, use the `rbffit` function to construct an interpolant to the data. Next, use the MATLAB function `meshgrid` to generate a rectangular grid of points for evaluating the interpolant. In the x-direction the grid should contain 100 equally spaced points in the interval $[0, 261]$, and in the y-direction the grid should contain 153 equally spaced points in the interval $[0, 399]$. The function `rbfval` should be used to do the evaluation. Produce a surface plot of the interpolant from these grided data values using the `surf` command. Include the original scattered data values in the plot and make sure you include labels on the coordinate axes and a title. Once the surface plot has been generated, experiment with the functions `shading`, `colormap`, `lighting`, and `camlight` to change the appearance of your plot. The following sequence of those commands should generate a very sharp looking plot

   ```
   shading interp;
   colormap(autumn);
   lighting phong;
   camlight right;
   ```

   Finally, generate a contour plot of the surface using the command `contour`. Make your plot contain 14 contours from 0 to 260 in increments of 20. Include the original (x, y) sample locations on your plot, and labels.

2. The almost universally used algorithm to compute $\sqrt{a}$ is the recursion

   $$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right), \tag{1}$$

   easily obtained by means of Newton's method for the function $f(x) = x^2 - a$.

   (a) Let $a = 4$ and $x_n = x$ in the right hand side of (1), then plot the resulting function together with the line $y = x$. Using the graphical fixed-point iteration process discussed in class, estimate in which intervals around the root $x = 2$ the above iteration will converge. Verify you prediction with numerical results.

   (b) Assume you are working with a very simple processor that only supports addition, subtraction, multiplication and halving (a subraction of one in the exponent of a base-2 number), but not a general divide. Devise a fast algorithm using Newton's method for this processor to directly approximate $\frac{1}{\sqrt{a}}$. You can then use this algorithm to compute $\sqrt{a}$ by simplity multiply with the result of the algorithm by a.

3. NCM, problem 4.10
4. NCM, problem 4.14
5. NCM, problem 4.15
6. NCM, problem 4.16
7. Steffensen's method is a quadratically convergent algorithm for computing the solution to the general fixed point problem:

$$x = g(x).$$

   Unlike Newton's method, which is also quadratically convergent, Steffensen's method does not require any information about the derivative of $g(x)$. Given an initial guess x_0 for the solution, Steffensen's method attempts to improve upon that guess according to the equation:

$$x_{n+1} = x_n - \frac{(g(x_n) - x_n)^2}{g(g(x_n)) - 2g(x_n) + x_n}, \quad n = 1, 2, \ldots$$

   (a) Implement Steffensen's method in MATLAB . Your function should take as input the function $g(x)$ in the fixed point iteration, an initial guess x_0 for the solution, and input for any problem specific parameters for $g(x)$ (see the `varargin` MATLAB command and its use in the `fzerotx` function from the NCM bood). Your function should output a vector containing all the iterates until the termination criteria (or failure to terminate) is met. Use the machine epsilon in your stopping criteria. Avoid unnecessary function evaluations—you should not need to use more than two function evaluations per iteration. Turn in a listing of your code.

   (b) Use your routine to find the zero of the equation

$$x + \text{erf}(2(x - a)) = x + \frac{2}{\sqrt{\pi}} \int_0^{2(x-a)} e^{-t^2} dt = 0$$

   which has exactly one real-valued solution (use the MATLAB function `erf` to calculate the integral). Let $a = 1$, and start with an initial guess $x_0 = 0$. Report the successive iterates from your function in a 'nice' table so that you can see how convergence proceeds.