# First Group Project

This will count for 20 points of the first test. There are two parts and you can split up the work, but everyone should read over the final version to make sure you agree with the results.

1. **Finding all roots of an equation.** The problem here is to find all solutions to an equation $f(x) = 0$ defined on an interval. To be concrete let us assume the interval is $[0, 10]$. Assume that the function (which will be stored in `fcn.m`) has the following properties
   (a) It is continous and has continuous first and second derivatives. (But you are not to assume that you have a method of computing the derivative directly.)
   (b) Any two roots of $f(x) = 0$ are at a distance at least .1 apart. (But it is possible that $f(x)$ has no roots at all.

   You are to write the following:
   (a) A MatLab script (to be called `group.m`) that will find all the roots of $f(x) = 0$ accurate to 6 decimal places or say that there are no roots. For each root it should also give a "safe graded interval" about the roots where the function $f(x)$ changes sign and which is short enough that it insures that you answer is accurate to the required amount. Moreover this script should satisfy
      (i) I can tell how to run it by just typing `help group`.
      (ii) It gives the roots from smallest to largest.
      (iii) Anything that is printed to the screen should have an explanation as to what it is. (That is "the first root is ____ and we know that this is accurate to within $10^{-6}$ and the function changes sign on the interval ____.)
      (iv) It tells how many roots there are.
      (v) It does not print out too much stuff other than the roots, the safe graded intervals and the number of roots.
   (b) Documentation for the method used. This should include:
      (i) What methods were used (Bisection, Modified Newton, or a combination of these) and brief explanations of why these methods are appropriate.
      (ii) Why the method used should get all of the roots.
      (iii) A little discussion of how many steps the method should require.

   Things that will cost you points:
   - I will test the program on several functions and you will lose points if the program misses any roots of gives error message such as `NaN` as output.
   - There is no discussion of what "stopping criterion" being used.

2. **Convergence analysis of the modified Newton's method.** Let $h$ be a small positive number and let $f(x)$ be a function and define

$$g(x) = x - \frac{2h}{f(x+h) - f(x-h)} f(x).$$

Then the modified Newton's method is to start with an initial guess $x_0$ and define a sequence $x_0, x_1, \ldots$ by

$$x_{n+1} = g(x_n).$$

Now assume that $r$ is a root of $f(x) = 0$ (that is $f(r) = 0$) and assume $f'(r) \neq 0$. Then

(a) Show that $g(r) = r$.

(b) Let $x$ be a number close to $r$. Then by the mean value theorem

$$g(x) - r = g(x) - g(r) = g'(\xi)(x - r)$$

where $\xi$ is between $x$ and $r$ (and thus $\xi$ is also close to $r$). Use this fact to argue that

$$g(x) - r \approx g'(r)(x - r)$$

and thus if $e_n = x_n - r$ is the error at the $n$-th step then

$$|e_{n+1}| \approx |g'(r)||e_n|.$$

(c) Now compute $g'(r)$ and show that $g'(r) \approx Ch^2$ where $C$ is a constant (which you should compute) depending on $f$ and its derivatives at $r$. (This may involve using Taylor's theorem $f(r+h)$ and $f(r-h)$.)

(d) Put this all together to get that the errors satisfy

$$|e_{n+1}| \approx Ch^2|e_n|.$$

(With the same $C$ as above.) Can you use this to give rules for choosing $h$ to get good convergence?