Math 329 Unit 1 Fall 2014

Written Exercises:

1. Use Newton's method on the equation $x^2 = N$ to derive the following iterative algorithm for obtaining the square root of N:

$$x_{i+1} = \frac{1}{2}\left(x_i + \frac{N}{x_i}\right).$$

2. Similarly derive algorithms to find the third and fourth roots of N.

3. Expand $f(x)$ in a Taylor Series about the point $x = a$ (include at least up to cubic terms in the series). Let a stand for the initial guess for a root and x represent the root you wish to find. Use the Taylor series to derive Newton's method. What is the largest error term in the limit of small $(x - a)$?

4. What starting values cause Newton's method to fail completely for $f(x) = (x+1)^3(x-1)$? Explain.

5. Find a function $f(x)$ that is discontinuous on a bracketing interval (a, b) but for which bisection starting with that interval still does find the root.

6. Suppose that $f(x)$ has five simple roots within (a, b) labeled x_1 to x_5 in increasing order. Which of these five roots are possible outcomes of bisection if it begins with the bracket (a, b) and never lands exactly on a root? Are there any roots that bisection will not find unless you land exactly on those roots? Explain.

7. Suppose a polynomial has a double root at $x = 1$ and a triple root at $x = 3$. Ignoring the chance of landing exactly on a root, is it possible to find all the roots using bisection? What about Newton's method?

Computer Exercises:

Zip together the files you create and email them to me. The files should include: bisection.m, test_bisection.m, newton.m, slowfunction.m, unit1.m, unit1_output.txt where unit1.m runs the code to answer questions 9-11 and unit1_output.txt shows the results of running unit1.m. Try to make unit1.m print out your answers to the "briefly explain" parts of the questions [use the disp command].

8. Write a Matlab function (bisection.m) for rootfinding using the bisection method. The function should be written as generally as possible, requiring inputs for the function to evaluate and two starting values which bracket the root. Include checks that the input values are reasonable (i.e. that the root is bracketed). Also accept optional inputs (with default value 1e-10) for absolute tolerance levels in both x and f. Allow an optional input to turn on or off printing (default on) of the latest x and f values at each iteration. Print using code like: `fprintf('%2i:  c=%18.12e, f=%18.12e\n',count,c,fc)`
Your function should be well documented with comment lines and should include comments

at the top for the help command to display. (Usually this includes instructions for how to use the script and a brief description of what the script does.)

Write a script to test your code. It should define functions `f=x.^2-2, g=1e13*(x.^2-2)` and `h=1e-13*(x.^2-2)` and find the root starting with bracket [1,2]. In the same script test the slow function you created for the in-class tutorial (which pauses for 1 second before returning $f(x)$ above). Use commands tic and toc to time your rootfinding routine for the slowfunction while setting a large tolerance of 1e-3 for both tolerances. If your routine takes more than 20 seconds for the slowfunction, rewrite it to minimize function calls until it takes 20 seconds or less.

Think about it: For each function tested figure out whether the result is within the tolerances tolx and tolf. What feature of the function determines which tolerance stops the routine? (no need to hand in this.)

9. Use your bisection function to find where the graphs of the functions $y = x^2 - 2$ and $y = e^x$ intersect. Get the intersection value correct to 5 decimal places.

10. The goal here is to find the smallest positive root to 7 decimals of accuracy. Write a routine (hint: while loop) to step slowly from zero until you find a bracket. Then use your bisection routine for that bracket to find the root.

    a) $x^4 - 2x - 1 = 0$
    b) $\cos(3x) + 1 = e^{x^2}$
    c) $e^{x-1} = x^3 + 2$
    d) $e^{-x} = -\sin(2x)$

11. Use the supplied script (newton.m available at http://math.colgate.edu/math329/newton.m) which uses Newton's method to find the roots of a function when the derivative is known. Answer the following.

    a) Explore the convergence rate for the different roots of $f(x) = (x+1)^3(x-1)$. Notice that this function has roots $x = -1$ and $x = 1$. Use starting values ± 0.9 and ± 1.1 and compare the number of iterations it takes to find the roots within 0.0001 of the correct value. What is the difference between the convergence rates and why do you think this difference exists?

    b) Matlab routines that work with real numbers often work with complex numbers as well if you start with a complex valued initial point like $(1 + i)$. The following equations may have complex solutions. Apply Newton's method to find **all** the solutions of the following equations. Briefly explain for each how (and whether) you know you have found them all.

        i) $x^2 = -2$
        ii) $x^3 - x^2 - 1 = 0$
        iii) $x^4 - 3x^2 + x + 1 = 0$
        iv) $x^2 = (e^{-2x} - 1)/x$