Prof. Ming Gu, 861 Evans, tel: 2-3145
Office Hours: WF 2:00-3:000PM
Email: mgu@math.berkeley.edu
http://www.math.berkeley.edu/∼mgu/MA128A2010F

Math128A: Numerical Analysis

Programming Assignment #1, Due 9/29/2010

Consider the cubic equation

$$ax^3 + bx^2 + cx + d = 0, \tag{1}$$

where $a \neq 0$, b, c, and d are random constants.

1. Compute the roots using the exact formulas for the roots of equation (1). For this write a function .m file with the format

   $$[\texttt{roots}] = \texttt{cubic}(a, b, c, d),$$

   where a, b, c, d are the coefficients of the cubic polynomial input. This function should return a vector with the three roots of the cubic, sorted using the Matlab `sort()` function.

2. To solve equation (1) with numerical methods, for each of the bisection method, Newton's method, and the Muller's method, you should

   (a) First compute a root of equation (1) using each of the methods listed. Do this by creating function .m files with the formats

   $$[\texttt{root}] = \texttt{bisection}(a, b, c, d),$$

   $$[\texttt{root}] = \texttt{newton}(a, b, c, d),$$

   $$[\texttt{root}] = \texttt{muller}(a, b, c, d),$$

   where a, b, c, d are the coefficents of the cubic. Use relative change in iteration, $|p_{n+1} - p_n|/|p_n|$ as a measure of error, and tolerance 10^{-6}, except in bisection method, where you should use $|f(p_n)|$ instead. Be sure to use the function .m file template and include some comments about how your function works.

   (b) Use deflation procedure to reduce equation (1) to a second order equation. Do this by creating a function $[A, B, C] = \texttt{deflate}(a, b, c, d, \texttt{root})$, where a, b, c, d are the coefficients of an input cubic, `root` is a root of the cubic, and the outputs A, B, C are coefficients of a quadratic polynomial with the same roots as the input cubic, except `root`.

(c) Solve the second order equation. To do this make a function

$$[\texttt{roots}] = \texttt{quadratic}(a, b, c),$$

that finds the roots of the quadratic $ax^2 + bx + c = 0$ using the quadratic equation.

3. Add more input arguments as necessary (such as initial interval for bisection and initial guess for other methods.) Your initial guesses do not have to be very robust for this assignment.

4. Compare the accuracy of roots computed using the exact root formulas and the above numerical methods with those obtained with the `roots` matlab command. Do this for 100 random cubic polynomails, and report the average and maximum differences between your roots and those of `roots`.

   Write a script .m file called `testrootfinders.m` that calls your functions `cubic, bisection, newton, muller, deflate, quadratic` as well as the Matlab command `roots` and returns the results requested above. To generate random coefficients a, b, c, d you can use the `rand()` function. To check the difference between approximate and exact roots, you can use

   $$\texttt{norm}(\texttt{exactroots} - \texttt{approximateroots}).$$

   Have the script file print each of the results in such a way that we can tell what they are. Make sure that your vectors of exact roots and approximate roots are both sorted when you compare them!

Important: The names, inputs, and outputs of all the functions and scripts you write must be exactly as written above. This includes capitalization of .m file names and order of inputs and outputs. If not, we cannot grade your assignment!