Math 160S: Mathematical Numerical Analysis

Homework 1: Computer Arithmetic and Bisection Method

Due: *Thursday, January 22, 2009*

1. Read Burden and Faires Chapters 1 and 2.1.

2. Write a `matlab` program to determine ϵ_{mach}. Turn in your program for (a) and answer (b).

   (a) Declare two variables: `x`, `epsilon`. Initialize $\epsilon = 1.0$. Create a loop, each time through the loop, assign $x = 1.0 + \epsilon$ and print-out the values of $x - 1.0$ and ϵ. Then, each time through the loop, also halve the value of ϵ. Repeat until the value of $x - 1.0$ is no longer positive.

   (b) Use the result of (a) to determine $\boxed{\text{number of bits in the mantissa}}$[1] of `matlab` data types.

3. Write a `matlab` program to determine the *overflow* level:

   (a) First, determine the largest possible variables: Declare a variable r and initialize $r = 1.0$. Create a loop, each time through the loop, print-out the values of r and $\log 2(r)$. (What is the meaning of the second quantity?) Then, each time through the loop, also double the value of r. Repeat until the value of r becomes infinite.
   $\boxed{\text{Find the value } k \text{ for the largest } 2^k \text{ before overflow.}}$

   (b) Use the result of (a) to determine $\boxed{\text{number of bits in the exponent}}$ of `matlab` data types, and $\boxed{\text{the overall total number of bits}}$ based on this question and on (2).

4. Use Taylor approximations to avoid the loss-of-significance error in the following computations for small x:

$$\text{(a)} \qquad f(x) = \frac{e^x - e^{-x}}{2x}$$

$$\text{(b)} \qquad f(x) = \frac{\ln(1-x) + xe^{x/2}}{x^3}$$

   Box your final answers. In both cases, what is $\boxed{\lim_{x\to 0} f(x)}$? No programming is required for this problem.

5. Write programs to find the zeros of the function

$$f(x) = \tan(x/4) - 1$$

   within the interval $a = 0$ and $b = 4$, using the bisection method. The exact solution is $x = \pi$.

   (a) Write a version of bisection method that stops after taking exactly n steps. Call the routine `bisect1`. Make a $\boxed{\text{linear-log plot}}$ of the absolute value of the error in the solution as a function of the number of steps for $n = 1, 2, 3, \ldots, 200$. What is your $\boxed{\text{minimum achievable error}}$ in the solution? How does it relate to ϵ_{mach}? How many steps $\boxed{n}$ did it take? What happens if you take more steps than that?

[1] The $\boxed{\text{boxed}}$ items in homework questions are the final answer for each part—please box them on your problem sets to make things easier to grade.

(b) Write a version of bisection method that stops when the error in the problem reaches a set tolerance, i.e., $|f(c)| < \delta y$. Call the routine `bisect2`. Keep track of the number of times that you use the function `f(x)` using a variable `fcn_count`. (Be sure to re-zero this counter each time you start another bisection.) Make a log-linear plot of the `fcn_count` value needed to reach a desirable tolerance δy for $\delta y = 1, 0.5, 0.25, 2^{-3}, \ldots, 2^{-100}$. What happens if δy is "too small"? What is the minimum value of δy that you can reach?

(c) Write a version of bisection method that stops when the error in the solution reaches a set tolerance $|x^* - c| < \delta x$. Call the routine `bisect3`. Explain how you estimate the value $|x^* - c|$. What is the minimum achievable error in the solution?

What is your most accurate solution? Turn in print-outs of your programs.