Homework 5 Foundations of Computational Math 1 Fall 2011

Problem 5.1

Consider the following numbers:

- 122.9572
- 457932
- 0.0014973

5.1.a. Express the numbers as floating point numbers with $\beta = 10$ and $t = 4$ using rounding to even and using chopping.

5.1.b. Express the numbers as floating point numbers with in single precision IEEE format using rounding to even. It is strongly recommended that you implement a program to do this rather than computing the representation manually.

5.1.c. Calculate the relative error for each number and verify it satisfies the bounds implied by the floating point system used.

Problem 5.2

(25 points)

Consider the function

$$f(x) = \frac{1.01 + x}{1.01 - x}$$

5.2.a. Find the absolute condition number for $f(x)$.

5.2.b. Find the relative condition number for $f(x)$.

5.2.c. Evaluate the condition numbers around $x = 1$.

5.2.d. Check the predictions of the condition numbers by examining the relative error and the absolute error

$$err_{rel} = \frac{|f(x_1) - f(x_0)|}{|f(x_0)|}$$
$$err_{abs} = |f(x_1) - f(x_0)|$$

with $x_0 = 1$, $x_1 = x_0(1 + \delta)$ and δ small.

Problem 5.3

Most recent machines use a binary base,$\beta = 2$, but the number of bits, t, may vary in a floating point system. The following algorithm is an attempt to determine t experimentally.

```
$x = 1.5$, $u = 1.0$,$t = 0$, $\alpha = 1.0$
while $x > \alpha$
    $u = u/2$
    $x = \alpha + u$
    $t = t + 1$
end
```

5.3.a. Assume that the floating point system has $\beta = 2$ and uses a hidden bit normalization. Does this algorithm find t? Does the method of rounding affect your answer?

5.3.b. Apply the algorithm to a machine that uses $\beta = 2$ in single precision and double precision. Do your observations from the output of the code agree with the IEEE floating point standard for single and double precision floating point numbers?

Problem 5.4

For this problem assume that the floating point system uses $\beta = 10$ and $t = 3$. The associated floating point arithmetic is such that $x \boxed{op} y = fl(x\ op\ y)$.

Let x and y be two floating point numbers with $x < y$ and consider computing their average $\alpha = (x + y)/2$.

Consider three algorithms for computing α. The parentheses indicate the order of the floating point operations.

- $\alpha_1 = ((x + y)/2.0)$
- $\alpha_2 = ((x/2.0) + (y/2.0))$
- $\alpha_3 = (x + ((y - x)/2.0))$

For the floating point values $x = 5.01$ and $y = 5.02$:

5.4.a. Evaluate α_1, α_2, and α_3 in the specified floating point system.

5.4.b. Explain the results.

5.4.c. Some algorithms produce a series of intervals by splitting an interval (x, y) into intervals (x, α) and (α, y) and choosing to process one of these two smaller intervals further in the next step of the algorithm. Could the behavior observed for the three average computations cause difficulties for such an algorithm?