Homework 1. Due Friday Sept. 6 by 10am in my mailbox

Always provide complete, well written solutions.

1. **Floating point representation**

   Have a look at [1], in particular read the *Exponent* paragraph.

   (a) What term is used in [1] to call the mantissa (used in [2]) of a floating point number?

   (b) Why is the representation of zero not straightforward? What is the IEEE solution to represent zero? Is the representation of zero compatible with the representation of denormalized numbers?

   (c) What is the IEEE solution to represent infinity? and NaN?

   (d) Explain in your own words what are the biased and unbiased exponents.

2. **Invalid operations**

   (a) What are the values of the expressions $\infty/0$, $0/\infty$, and ∞/∞? Justify your answer.

   (b) For what nonnegative values of a is it true that a/∞ equals zero?

   (c) The formula $R_1R_2/(R_1+R_2)$ is equivalent to $\frac{1}{1/R_1+1/R_2}$ if R_1 and R_2 are both nonzero. Does it deliver the correct answer using IEEE arithmetic if R_1 or R_2,or both, are zero?

3. **[Ref. [2], Chapter 2, Problem 3]**

   We consider the sequence

   $$\forall k \geq 0, \forall j \geq 0, f_{jk} := \sin(x_0 + (j-k)\pi/3),$$

   as well as $\widehat{f}_{jk}$, for all $k \geq 0$ and $j \geq 0$, defined from implementing in double precision floating point:

   $$\begin{cases} \widehat{f}_{j0} = \sin(x_0 + j\pi/3), \forall j \geq 0, \\ \widehat{f}_{j,k+1} = \widehat{f}_{j,k} - \widehat{f}_{j+1,k}, \forall k > 0, \forall j \geq 0. \end{cases}$$

   (a) Show that f_{jk} satisfies the recurrence relation

   $$f_{j,k+1} = f_{j,k} - f_{j+1,k}\ . \tag{1}$$

   We will now consider this as a formula that computes the f values on level $k+1$ from the f values on level k.

   (b) Assuming that $\left|\widehat{f}_{jk} - f_{jk}\right| \leq \epsilon$ for all j, show that $\left|\widehat{f}_{j,k+1} - f_{j,k+1}\right| \leq 2\epsilon$ for all j. Thus, if the level k values are very accurate, then the level $k+1$ values still are pretty good.

(c) Write a matlab program that computes $e_k = \widehat{f}_{0k} - f_{0k}$ for $1 \leq k \leq 60$ and $x_0 = 1$. Note that f_{0n}, a single number on level n, depends on $f_{0,n-1}$ and $f_{1,n-1}$, two numbers on level $n-1$, and so on down to n numbers on level 0. Print the e_k and see whether they grow monotonically. Plot the e_k on a linear scale and see that the numbers seem to go bad suddenly at around $k = 50$. Plot the e_k on a log scale. For comparison, include a straight line that would represent the error if it were exactly to double each time.

4. **[Ref. [2], Chapter 2, Problem 7]**

   Assuming that `x, y, z, w` are floating point numbers in single precision, we do lots of arithmetic on the variables `x, y, z, w`. In each case below, determine whether the two arithmetic expressions result in the same floating point number (down to the last bit) as long as no `NaN` or `inf` values or denormalized numbers are produced.

   (a)

```
( x * y ) + ( z - w )
( z - w ) + ( y * x )
```

   (b)

```
( x +   y ) + z
  x + ( y   + z )
```

   (c)

```
  x * oneHalf + y * oneHalf
( x + y ) * oneHalf
```

   (d)

```
  x * oneThird + y * oneThird
( x + y ) * oneThird
```

References

[1] David Goldberg, What Every Computer Scientist Should Know about Floating-Point Arithmetic.

[2] Bindel and Goodman, Principles of scientific computing