ACM40290: NUMERICAL ALGORITHMS

Assignment 2: Floating point number systems

The purpose of this exercise is to familiarise yourself with MATLAB's floating-point arithmetic system, which uses IEEE double precision floating point arithmetic by default. It is important to know the parameters of the number system you are using and to be aware of its limitations.

Machine precision. The accuracy of a floating point system can be characterized by a quantity variously known as *machine precision*, *machine epsilon*, or *macheps*. For a given precision, its value — which we denote by ϵ_{mach} — is defined to be the difference between 1.0 and the next highest floating point number. For a floating point number system with base b and precision p (i.e. the number of digits in the mantissa), machine epsilon is given by $\epsilon_{\text{mach}} = b^{1-p}$.

A closely related quantity is the *unit roundoff*, which depends on the particular rounding rules used. With rounding by chopping, $u = \epsilon_{\text{mach}}$ whereas with rounding to nearest $u = \frac{1}{2}\epsilon_{\text{mach}}$. The unit roundoff is important because it determines the maximum possible relative error in representing a nonzero real number x in a floating point system. A characterization of the unit roundoff that you may sometimes see is that it is the smallest number ϵ such that $fl(1+\epsilon) > 1$ Here $fl(x)$ is the floating point approximation to x.

Note: Depending on where you look, sometimes machine epsilon and unit roundoff are defined slightly differently than we have defined them here. Sometimes they are even defined to be the same thing. For our purposes, we will stick with the definitions given here as they distinguish between the two important quantities to keep in mind for floating point numbers: (1) the spacing between numbers; and (2) the roundoff error in converting an exact number into a floating point number.

EXERCISES

(1) Write a Matlab function MachEps() that calculates machine epsilon. Briefly explain how you did the computation and your results.

(2) Matlab has various important constants and functions built in. Some of these are eps, realmax, realmin, inf, pi. Also available are date, ver, and version which are useful for annotating output. Determine what these are on your system.

(3) Explain why an alternating infinite series, such as

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

for $x < 0$ is difficult to evaluate accurately in floating point arithmetic. How would you then accurately calculate $\exp(x)$ for x negative?

(4) What happens when you evaluate the infinite series

$$\sum_{n=1}^{\infty} \frac{1}{n}$$

Explain what you would expect with exact arithmetic, and why summing the series in floating point arithmetic yields a finite sum.

(5) Determine the response of Matlab to the indeterminates

$$\frac{0}{0}, \frac{\infty}{\infty}, \infty * 0, 1^{\infty}, \infty - \infty, 0^0, \infty^0$$

(6) Calculate mathematically (by hand) the result of the following 4 statements:

$$a = \frac{4}{3}; b = a - 1; c = b + b + b; e = 1 - c;$$

Now use MATLAB to do the same calculations. Explain the result.

All answers should be submitted via Blackboard and should include MATLAB code along with a brief writeup.

Due: 5pm Friday, October 6th 2017