MATH 437: Principles of Numerical Analysis

Prof. Wolfgang Bangerth // Blocker 507D // bangerth@math.tamu.edu
TA: Youli Mao // Blocker 505E // youlimao@math.tamu.edu

Homework assignment 1 – due Thursday 9/5/2013

Problem 1 (Continuous vs. discrete). Functions $f(x)$ are usually defined over an entire domain $x \in I = (a, b) \subset \mathbb{R}$ and – if interesting – take values in an image $f(I) \subset \mathbb{R}$. Both domain and image are sets with infinitely many elements. On the other hand, computers can only represent numbers using a finite number of bits, most often as 32-bit (`float`, or `REAL*4`) or 64-bit (`double`, or `REAL*8`) IEEE floating point numbers, which store numbers in the form $\pm m 2^e$, where $0 \le m < 1$ is the mantissa

$$m = b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \cdots + b_M 2^{-M} \tag{1}$$

and e is the exponent and has the form

$$e = \pm(u_0 2^0 + u_1 2^1 + u_2 2^2 + u_3 2^3 + \cdots + u_E 2^E). \tag{2}$$

The coefficients b_i, u_i are single-bit numbers, i.e., either 0 or 1. In the binary system, floating point numbers can therefore be written as $\pm 0.b_1 b_2 b_3 \ldots \times 2^{\pm u_E u_{E-1} u_{E-2} \ldots u_0}$. The total number of bits needed for the representation are M bits for the mantissa, $E+1$ bits for the exponent, and 2 bits for the two signs.

Obviously, not all elements of I and $f(I)$ can be represented. Write a short program to find

a) an approximation to the smallest and largest positive numbers that can be represented in `float` and `double` precision;

b) the smallest `float` and `double` floating point number you can add to 1 such that the result is different from 1.

c) In exact arithmetic, the system of linear equations

$$\begin{aligned} x_1 + x_2 &= 2, \\ x_1 + 10^{20} x_2 &= 1 + 10^{20} \end{aligned}$$

has the solution $x_1 = x_2 = 1$. Are there corresponding floating point numbers for x_1, x_2 that when plugged into the left hand side of the equations yields the exact values on the right hand side? If so, which? If not, is this a problem?

Problem 2 (Floating point vs real numbers). Let ε be the smallest floating point number in double precision such that in computer arithmetic $1 + \varepsilon \neq 1$ (you determined ε in Problem 1b). What are the floating point values of $(1 + \frac{\varepsilon}{2}) - 1$, $1 + (\frac{\varepsilon}{2} - 1)$, and $(1 - 1) + \frac{\varepsilon}{2}$? In what important way do exact and floating point arithmetic therefore differ?

Problem 3 (Taylor series). Derive the first four terms and integral remainder term of the Taylor series of

a) $f(x) = \sin x$ when expanded around $x_0 = 0$;

b) $f(x) = x \sin x$ when expanded around $x_0 = \pi/2$;

c) $f(x) = 4(x-3)^2(x+2)$ when expanded around $x_0 = 1$. What happened to the remainder term and what does this mean for the accuracy of the Taylor expansion with only four terms?

d) $f(x) = x^x$ when expanded around $x_0 = 1$. (Note: You will first have to figure out how to differentiate $f(x)$. Use the identity $a^b = e^{b \ln a}$.)

You may use a computer algebra system like Maple to compute derivatives of $f(x)$, but not to generate the entire Taylor series.