Faculté des Sciences de Base
Master in mathematics
Master in computational science and engineering
Scientific computing applied to linear algebra
February 22, 2011

Serie 1

This serie consists in experimenting some types of error on the computer

For having the standard IEEE floating point behavior, use the gnu compiler (gcc) with the option -ffloat-store.

Exercise 1 (Integer overflow) *Write a C program for computing* $n!$ *with the recurrence* $n! = n * (n-1)!$. *Print the result for* $n = 1, 2, ..., 20$.

Exercise 2 (Accumulation) *One consider the series:*

$$s = \sum_{n=1}^{\infty} \frac{1}{n^3} \simeq 1.2020569.$$

Within a C program, compare, using simple precision real numbers (`float`), for $N = 2000$, *the two following approximation methods:*

$$s \simeq s_N = 1 + \frac{1}{2^3} + ... + \frac{1}{(N-1)^3} + \frac{1}{N^3}, \tag{1}$$

$$s \simeq S_N = \frac{1}{N^3} + \frac{1}{(N-1)^3} + ... + \frac{1}{2^3} + 1. \tag{2}$$

Exercise 3 (Cancellation) *We want to get the value of the function*

$$x \mapsto f(x) = \frac{1 - \cos(x)}{x^2}, \tag{3}$$

at the point $x = 1.2e - 8$, *with a C code with real numbers of type `double`. With the help of the relationship* $\cos(x) = 1 - 2\sin^2(x/2)$, *we see that*

$$f(x) = \frac{1}{2}\left(\frac{\sin(x/2)}{x/2}\right)^2, \tag{4}$$

which implies the bound $f(x) \leq 1/2, x \geq 0$. *Compare the evaluation obtained with the formulae (3) and (4).*

Exercise 4 (Instability) *We look for an approximation of* e *by the formula*

$$e = \lim_{n\to\infty}\left(1 + \frac{1}{n}\right)^n, \tag{5}$$

with the help of

$$e_N = \left(1 + \frac{1}{N}\right)^N, \; N = 10^i, i = 1, 2, ..., 9. \tag{6}$$

Print the results with a C code using real numbers in simple precision.

Exercise 5 *The precision of floating point numbers on a computer is measured by* ***the machine epsilon number****, which is the largest positive real floating point number* ϵ *such that 1.0+*$\epsilon/b = 1.0$, *where* b *is the basis used for representing the floating point numbers. On most computers* $b = 2$ *and one can represent this number as* $\epsilon = 2^{1-p}$. *Write a C progam for computing* ϵ *for numbers of type `float` and `double`.*