RUNNING A NUMERICAL CODE ON NEW HARDWARE TECHNOLOGIES: IS SINGLE PRECISION FLOATING POINT ARITHMETIC SUFFICIENT?

Christophe Denis
School of EEE and Computer Science
The Queen’s University of Belfast, U.K.
Laboratoire d’Informatique de Paris 6, UPMC Univ Paris 06
email C.Denis@qub.ac.uk

Many companies like banks are interested by new hardware technologies (for example FPGAs and GPUs) to improve the performance of their numerical codes for a fixed cost and power consumption. In order to study the capability of these new architectures we are currently investigating the implementation of a financial program code based on Monte Carlo computations (using a GPU) and a physics program requiring a very large number of two dimensional integrals (using FGPAs).

Unfortunately, on these architectures the use of the double precision floating point arithmetic can be expensive or even unavailable. It is then crucial to carry out a rigorous numerical verification of the single precision computation to give confidence in the accuracy of the computed results.

The aim of the talk is to present the numerical investigation performed on these two codes running in single precision by using a stochastic and an interval arithmetic approach. These two approaches will be presented during the talk and compared in terms of performance, easy to use and their effects on the original source code.

The CADNA library is an implementation of discrete stochastic arithmetic for code written in Fortran, C and C++ [1]. Several libraries implementing the interval arithmetic exist: we use the MPFI library for programs written in C[3]. For all numerical results in this abstract, only the significant digits are printed.

The code computing the two dimensional integrals is written in C. Two versions of this code have been written respectively with the CADNA and the MPFI libraries. A numerical instability has been detected during the computation of the two dimensional integral $I$ in the CADNA version and not in the MPFI version. With the single precision, the value $I$:

- computed with CADNA is $I = 1.648755 \times 10^{-2}$;
- computed with MPFI is $I = [1.64871253 \times 10^{-2}, 1.64879803 \times 10^{-2}]$;
The exact value given by a benchmark is $I = 1.6487550218 \times 10^{-2}$ so the number of significant digits of $I$ is 7.

The code computing the MonteCarlo calculations is written in C. It returns the value $v$ of the portfolio of swaptions and the vector of derivatives $L_b$ of $v$ with respect to the vector of initial forward interest rates. The use of the CADNA library has detected an accumulation of roundoff error when averaging the payoffs with a classical summation from a very large number of paths. The classical summation has been replaced by the binary tree summation defined [2] to avoid these roundoff errors. For a number of paths equal to 96000 and using the single precision, the values:

- computed with CADNA are $L_b = 2.134813 \times 10^1$, $v = 2.243232 \times 10^2$
- computed with MPFI are $L_b = [2.128467941 \times 10^1, 2.137287140 \times 10^1]$, $v = [2.240517425 \times 10^2, 2.250145875 \times 10^2]$

The interval results are converted to double which is the center of these intervals rounded to the nearest double. We obtain $v = 2.243233 \times 10^2$, $L_b = 2.134817 \times 10^1$. The results obtained indicate that the number of significant digits is 5 for $L_b$ and for $v$. After these numerical improvements, the single precision floating point is sufficient for the two numerical codes.

**Acknowledgments**

The author wish to thank M. Giles and C.J. Gillan for providing their numerical codes and N.S Scott for his useful suggestions and comments.

**References**

