Chapter 15

Conclusion and Perspectives

With the recent IEEE standard update resulting in IEEE 754-2008, computer arithmetic will soon see important changes: the fused multiply-add (FMA) instruction will probably be available on most processors, and correctly rounded functions (at least in some domains) and decimal arithmetic will be provided on most systems. Also, it should be easier to specify whether one wishes the implicit intermediate variables of an arithmetic expression to be computed and stored in the largest available format (to improve the accuracy of the calculations) or in a format clearly specified in the program (to enhance software portability and provability).

We hope that future language standards will allow the users to easily and efficiently take advantage of the various possibilities offered by IEEE 754-2008.

Some instructions might ease the task of programmers. For instance, if, instead of having to use algorithms such as 2Sum (Algorithm 4.4, page 130) or 2MultFMA (Algorithm 5.1, page 152), the corresponding operations (i.e., the exact error of a floating-point addition or multiplication) were implemented on the floating-point units, we could much more efficiently compute accurate sums of many numbers or implement multiple-precision arithmetic. This should not complicate the existing architectures too much, since, to guarantee correctly rounded results, a conventional floating-point adder or multiplier already has to do a large part of the task. Also (but this might be slightly more complex to implement in hardware), having instructions for $\circ(x+y+z)$ or $\circ(xy+zt)$ would make very accurate functions much easier to program.

Concerning the Table Maker's Dilemma, the current methods will make it possible to know the worst cases for most univariate functions in the binary64 or decimal64 formats. For much wider formats (typically, binary128 or decimal128), unless new algorithms are found, we have no hope of determining the worst cases in the foreseeable future. And yet, methods based on the Lenstra–Lenstra–Lovász (LLL) algorithm (see the Appendix) might soon allow us to prove information of the kind "we do not know the hardness to round, but it is less than 400."

J.-M. Muller et al., *Handbook of Floating-Point Arithmetic*, DOI 10.1007/978-0-8176-4705-6_15,