Preface

Floating-point arithmetic is by far the most widely used way of approximating real-number arithmetic for performing numerical calculations on modern computers. A rough presentation of floating-point arithmetic requires only a few words: a number $x$ is represented in radix $\beta$ floating-point arithmetic with a sign $s$, a significand $m$, and an exponent $e$, such that $x = s \times m \times \beta^e$. Making such an arithmetic reliable, fast, and portable is however a very complex task. Although it could be argued that, to some extent, the concept of floating-point arithmetic (in radix 60) was invented by the Babylonians, or that it is the underlying arithmetic of the slide rule, its first modern implementation appeared in Konrad Zuse’s 5.33Hz Z3 computer.

A vast quantity of very diverse arithmetics was implemented between the 1960s and the early 1980s. The radix (radices 2, 4, 16, and 10 were then considered), and the sizes of the significand and exponent fields were not standardized. The approaches for rounding and for handling underflows, overflows, or “forbidden operations” (such as $5/0$ or $\sqrt{-3}$) were significantly different from one machine to another. This lack of standardization made it difficult to write reliable and portable numerical software.

Pioneering scientists including Brent, Cody, Kahan, and Kuki highlighted the relevant key concepts for designing an arithmetic that could be both useful for programmers and practical for implementers. These efforts resulted in the IEEE 754-1985 standard for radix-2 floating-point arithmetic, and its follower, the IEEE 854-1987 “radix-independent standard.” The standardization process was expertly orchestrated by William Kahan. The IEEE 754-1985 standard was a key factor in improving the quality of the computational environment available to programmers. It has been revised during recent years, and its new version, the IEEE 754-2008 standard, was released in August 2008.

By carefully specifying the behavior of the arithmetic operators, the 754-1985 standard allowed researchers to design extremely smart yet portable algorithms; for example, to compute very accurate sums and dot products, and to formally prove some critical parts of programs. Unfortunately, the subtleties of the standard are hardly known by the nonexpert user. Even more worrying, they are sometimes overlooked by compiler designers. As a consequence, floating-point arithmetic is sometimes conceptually misunderstood and is often far from being exploited to its full potential.
This and the recent revision of the IEEE 754 standard led us to the
decision to compile into a book selected parts of the vast knowledge on
floating-point arithmetic. This book is designed for programmers of numer-
ical applications, compiler designers, programmers of floating-point algo-
rithms, designers of arithmetic operators, and more generally the students
and researchers in numerical analysis who wish to more accurately under-
stand a tool that they manipulate on an everyday basis. During the writing,
we tried, whenever possible, to illustrate by an actual program the described
techniques, in order to allow a more direct practical use for coding and
design.

The first part of the book presents the history and basic concepts of
floating-point arithmetic (formats, exceptions, correct rounding, etc.), and
various aspects of the IEEE 754 and 854 standards and the new revised stan-
dard. The second part shows how the features of the standard can be used
to develop smart and nontrivial algorithms. This includes summation algo-
rithms, and division and square root relying on a fused multiply-add. This
part also discusses issues related to compilers and languages. The third part
then explains how to implement floating-point arithmetic, both in software
(on an integer processor) and in hardware (VLSI or reconfigurable circuits).
The fourth part is devoted to the implementation of elementary functions.
The fifth part presents some extensions: certification of floating-point arith-
matic and extension of the precision. The last part is devoted to perspectives
and the Appendix.

Acknowledgements

Some of our colleagues around the world and students from École Normale
Supérieure de Lyon and Université de Lyon greatly helped us by reading
preliminary versions of this book: Nicolas Bonifas, Pierre-Yves David, Jean-
Yves l’Excellent, Warren Ferguson, John Harrison, Nicholas Higham, Nicolas
Louvet, Peter Markstein, Adrien Panhaleux, Guillaume Revy, and Siegfried
Rump. We thank them all for their suggestions and interest.

We have been very pleased working with our publisher, Birkhäuser
Boston. Especially, we would like to thank Tom Grasso, Regina Gorenshtein,
and Torrey Adams for their help.

Jean-Michel Muller, Nicolas Brisebarre  Lyon, France
Florent de Dinechin, Claude-Pierre Jeannerod  July 2009
Vincent Lefèvre, Guillaume Melquiond
Nathalie Revol, Damien Stehlé
Serge Torres
Handbook of Floating-Point Arithmetic
Muller, J.-M.; Brisebarre, N.; de Dinechin, F.; Jeannerod, C.-P.; Lefèvre, V.; Melquiond, G.; Revol, N.; Stehlé, D.; Torres, S.
2010, XXIV, 572 p., Hardcover
A product of Birkhäuser Basel