Available online at www.sciencedirect.com
SCIENCE DIRECT®
The Journal of Logic and Algebraic Programming 64 (2005) 113–134
THE JOURNAL OF LOGIC AND ALGEBRAIC PROGRAMMING
www.elsevier.com/locate/jlap

Precise numerical computation ☆

Mark Sofroniou [a], Giulia Spaletta [b,*]

[a] *Research and Development, Wolfram Research, Champaign, IL 61820-7237, USA*
[b] *Mathematics Department, Bologna University, Piazza di Porta S. Donato, 5, 40127 Bologna, Italy*

Dedicated to Jerry B. Keiper (1953–1995) and Nicholas C. Metropolis (1915–1999)

Abstract

Arithmetic systems such as those based on IEEE standards currently make no attempt to track the propagation of errors. A formal error analysis, however, can be complicated and is often confined to the realm of experts in numerical analysis. In recent years, there has been a resurgence of interest in automated methods for accurately monitoring the error propagation. In this article, a floating-point system based on significance arithmetic will be described. Details of the implementation in *Mathematica* will be given along with examples that illustrate the design goals and differences over conventional fixed-precision floating-point systems.

AMS classification: 65G20; 65G30; 65G50

Keywords: Significance arithmetic; Interval arithmetic; Error analysis; Accuracy and precision; Condition number; Computer algebra systems; Computer aspects of numerical algorithms

1. Introduction

Knuth has expressed a desiderata of floating-point arithmetic as follows [19]:

> It would be nice if we could give our input data for each problem in an unnormalized form which expresses how much precision is assumed, and if the output would indicate just how much precision is known in the answer.

This enhancement would assist those who do not wish to undertake a rigorous analysis of computational error. Significance arithmetic is one approach to obtaining such a facility by providing local error monitoring. A floating-point system based on significance arithmetic dynamically adjusts the number of digits used in computations; it is therefore ideally

☆ Work partially supported by the grants 'CNR Agency 2000 Computational grids and applications' and GNCS 2002 'Analysis and development of parallel computational kernels for problem solving'.
* Corresponding author.

1567-8326/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jlap.2004.07.007

suited to a software implementation where the mantissa can readily change in size over a very wide range. A premise of *Mathematica*'s arbitrary precision significance arithmetic is that it is preferable that the dynamic control of digits in low-order operations should be based upon the precision of the operands rather than by considerations such as fixed format storage limitations.

Significance arithmetic is not new and has been around since the late 1950s [8,22,23]. Despite an initial period of enthusiasm, one of the reasons the model fell out of favor was inefficiency compared with alternative hardware implementations of fixed-precision arithmetic models, such as that which evolved into the IEEE 754 standard. Hardware implementations also imposed severe restrictions on the range of representable numbers. Furthermore error estimates, that were obtained using an integral value for the associated error, gave results that were frequently too pessimistic to be of practical value [40].

Recently there has been a resurgence of interest in automatic error control. Nowadays, many modern computational environments are equipped with facilities for carrying out arithmetic in software. Furthermore, there seems to be an increasing demand from the user community for tools which assist in a numerical study of problems, but at the same time give some guarantee of reliability of the solution.

There are some very detailed descriptions of IEEE arithmetic, such as [31], but significance arithmetic is less well known: details of the implementation in *Mathematica* are not widely available and indeed the possibility of using adaptive precision, with its associated error bounds, offered by this computer algebra system is sometimes overlooked entirely [31, pp. 92–93]. One goal here is to fill this void. A description of how errors are represented and combined in the implementation is given and some enhancements since the early days of development are outlined.

This article is organized as follows. Section 2 begins by introducing some standard notions and the terminology used in *Mathematica*. The error propagation model used in significance arithmetic is then explained in Section 3. In order to make this article as self-contained as possible, the relevant commands are introduced in Section 4 together with details of the implementation. Section 5 contains examples that are problematic for IEEE arithmetic and these are compared and contrasted with the results obtained using significance arithmetic. Some of the properties of significance arithmetic and functions that can be used to verify the implementation are also outlined. All computations in this article have been carried out using a pre-release version of *Mathematica* running on a Pentium IV processor machine with RedHat Linux 8.0.

2. Background

In this section, some standard definitions that are used in connection with floating-point arithmetic and error analysis are introduced, the ideas underlying significance arithmetic are described and the terminology that is used in *Mathematica* is explained.

2.1. Representation of numbers

Several definitions are now recalled to illustrate how real numbers are represented in a computer [20]. A base-β finite number X of length $n = k + m$ is an n-tuple with radix point between the k most significant digits and the m least significant digits: