ISO/IEC JTC1/SC22
Languages
Secretariat: CANADA (SCC)

ISO/IEC JTC1/SC22
N 1109

FEBRUARY 1992

TITLE: Comments from IFIP WG 2.5 (Numerical Software) on ISO/IEC CD 10967: Language Compatible Complex Arithmetic

SOURCE: Secretariat ISO/IEC JTC1/SC22

WORK ITEM: JTC1.22.28

STATUS: New

CROSS REFERENCE: CD 10967

DOCUMENT TYPE: Comments from Liaison Organization

ACTION: For information to SC22 Member Bodies.
For action by WG11.

Address reply to: ISO/IEC JTC1/SC22 Secretariat
J.L. Côté
Treasury Board Secretariat
140 O'Connor St., 10th Floor, Ottawa, Ontario, Canada, K1A 0R5
Tel.: (613)957-2496 Telex: 053-3336 Fax: (613)996-2690

Comments on
Version 3.1 of Draft ISO/IEC 10967:1991
Language Compatible Arithmetic

by members of IFIP Working Group 2.5 (Numerical Software)
31 Jan 1992

An extremely encouraging development of the past several years for those concerned with constructing portable numerical software has been the increasingly widespread use of hardware that conforms to the IEEE standard for binary floating-point arithmetic (ISO/IEC 559:1989 and ANSI/IEEE 754-1985). This was designed with great care and has many features that assist the construction of software that is robust and executes rapidly.

The proposed new standard specifies many desirable features for a floating-point type in a loose enough manner to encompass most of today's important hardware, though not Cray computers. Thus its adoption would put some pressure upon Cray Research to offer an alternative, but the pressure is there anyway since the workstation is becoming more and more the preferred mode of access and it is clearly desirable to be able to run small test cases on the workstation and see the same results for these cases on the supercomputer.

The authors clearly have a problem in deciding how loose the requirements should be, but it is really very hard to understand why they regard as acceptable the VAX D-format with its very restricted range (about 10^{-38} to 10^{+38}) for its precision (about 17 decimals) and the IBM double precision format whose range (about 10^{-76} to 10^{+76}) is little better.

The case that we wish to make is that the requirements should be very tight. The effect of a loose requirement is that software written to be robust in execution on all present and future machines that conform to a loose standard will be very verbose and probably slow in execution. An example is provided by Kahan (1991, Fig. 3). More likely is that programmers will not bother to add all these extra tests and the code will fail unexpectedly when moved to different hardware. An example of this is provided by Tydeman (1991), who considers the computation

$$x/\sqrt{x^2+y^2}$$

which might reasonably be expected never to yield a value greater than one. Such a tight specification is already provided by the IEEE standard for binary floating-point arithmetic, or the IEEE radix-independent standard for floating-point arithmetic (IEEE 854-1987), which has the great merit of permitting radix 10. Our suggestion is that any new standard be firmly rooted on these standards.

A further very serious defect of the proposed new standard lies in its treatment of exceptions, for which 'notification' is required. Notification may consist either of alteration of the control flow of the program (the authors say this is their preferred choice) or the output of a message in a 'hard-to-ignore' manner. Alteration of the control flow is becoming an increasingly unrealistic choice on today's hardware - the run-time penalty on vector or parallel hardware may be prohibitive. The alternative of millions of lines of output messages is even more unacceptable. What is needed is a mechanism that records the event without interupting the execution flow. Corrective action can then be taken when it is needed, perhaps by repeating the calculation with a different algorithm or in a higher precision, without penalizing the normal case. An excellent foundation has been provided for such an approach in the IEEE standards, which reinforces the value of basing any new standard on these standards.

Overall, one has the feeling that the proposed standard seeks to make the best of a bad situation (except that Cray arithmentic is regarded as beyond the pale). In fact, the situation has been improving steadily over the last few years as more and more computers with IEEE arithmetic come into use. The IEEE standard simplifies the problem of comparing and analyzing the results of computations. Computations performed on different machines can be compared and interpreted more easily, numerical software can be expected to produce consistent results on different systems, and so forth. The proposed new standard represents a turn away from a strict arithmetic standard, making such comparisons and interpretations far more difficult, and is thus detrimental to progress in scientific computation.

References:

Kahan, W. [1991]. Analysis and refutation of the LCAS. ACM SIGNUM Newsletter, 26, 2-15.

Tydeman, F. [1991]. Comments on LCAS draft 3.1. Presented to X3T2 in July 1991.