INTERNATIONAL STANDARD

ISO/IEC 10967-3

First edition
2006-05-01

Information technology — Language independent arithmetic —

Part 3: Complex integer and floating point arithmetic and complex elementary numerical functions

Technologies de l'information — Arithmétique indépendante des langages —

Partie 3: Arithmétique des nombres complexes entiers et en virgule flottante et fonctions numériques élémentaires complexes

Reference number
ISO/IEC 10967-3:2006(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

IISO/IEC 10967-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces.*

ISO/IEC 10967 consists of the following parts, under the general title *Information technology — Language independent arithmetic*:

— *Part 1: Integer and floating point arithmetic*

— *Part 2: Elementary numerical functions*

— *Part 3: Complex integer and floating point arithmetic and complex elementary numerical functions*

Introduction

The aims

Portability is a key issue for scientific and numerical software in today's heterogeneous computing environment. Such software may be required to run on systems ranging from personal computers to high performance pipelined vector processors and massively parallel systems, and the source code may be ported between several programming languages.

Part 1 of ISO/IEC 10967 specifies the basic properties of integer and floating point types that can be relied upon in writing portable software.

Part 2 of ISO/IEC 10967 specifies a number of additional operations for integer and floating point types, in particular specifications for numerical approximations to elementary functions on reals.

The content

The content of this document is based on part 1 and part 2, and extends part 1's and part 2's specifications to also cover operations approximating imaginary-integer and complex-integer arithmetic, imaginary-real and complex-real arithmetic, as well as imaginary-real and complex-real elementary functions.

The numerical functions covered by this document are computer approximations to mathematical functions of one or more imaginary or complex arguments. Accuracy versus performance requirements often vary with the application at hand. This is recognised by recommending that implementors support more than one library of these numerical functions. Various documentation and (program available) parameters requirements are specified to assist programmers in the selection of the library best suited to the application at hand.

The benefits

Adoption and proper use of this document can lead to the following benefits.

For programming language standards it will be possible to define their arithmetic semantics more precisely without preventing the efficient implementation of the language on a wide range of machine architectures.

Programmers of numeric software will be able to assess the portability of their programs in advance. Programmers will be able to trade off program design requirements for portability in the resulting program. Programs will be able to determine (at run time) the crucial numeric properties of the implementation. They will be able to reject unsuitable implementations, and (possibly) to correctly characterize the accuracy of their own results. Programs will be able to detect (and possibly correct for) exceptions in arithmetic processing.

Procurers of numerical programs will find it easier to determine whether a (properly documented) application program is likely to execute satisfactorily on the platform used. This can be done by comparing the documented requirements of the program against the documented properties of the platform.

Finally, end users of numeric application packages will be able to rely on the correct execution of those packages. That is, for correctly programmed algorithms, the results are reliable if and only if there is no notification.

Information technology — Language independent arithmetic —

Part 3: Complex integer and floating point arithmetic and complex elementary numerical functions

1 Scope

This part of ISO/IEC 10967 specifies the properties of numerical approximations for complex arithmetic operations and many of the complex elementary numerical functions available in a variety of programming languages in common use for mathematical and numerical applications.

An implementor may choose any combination of hardware and software support to meet the specifications of this document. It is the computing environment, as seen by the programmer/user, that does or does not conform to the specifications.

The term *implementation* (of this part) denotes the total computing environment pertinent to this part, including hardware, language processors, subroutine libraries, exception handling facilities, other software, and documentation.

1.1 Inclusions

The specifications of part 1 and part 2 are included by reference in this part.

This document provides specifications for properties of complex and imaginary integer datatypes and floating point datatypes, basic operations on values of these datatypes as well as for some numerical functions for which operand or result values are of imaginary or complex integer datatypes or imaginary or complex floating point datatypes constructed from integer and floating point datatypes satisfying the requirements of part 1 (ISO/IEC 10967-1). Boundaries for the occurrence of exceptions and the maximum error allowed are prescribed for each specified operation. Also the result produced by giving a special value operand, such as an infinity, or a NaN (not-a-number), is prescribed for each specified floating point operation.

This document provides specifications for:

a) Basic imaginary integer and complex integer operations.

b) Non-transcendental imaginary floating point and Cartesian complex floating point operations.

c) Exponentiation, logarithm, radian trigonometric, and hyperbolic operations for imaginary floating point and Cartesian complex floating point.

This document also provides specifications for:

d) The results produced by an included floating point operation when one or more operand values include IEC 60559 special values.

e) Program-visible parameters that characterise certain aspects of the operations.

1.2 Exclusions

This document provides no specifications for:

a) Datatypes and operations for polar complex floating point. This part neither requires nor excludes the presence of such polar complex datatypes and operations.

b) Numerical functions whose operands are of more than one datatype, except certain imaginary/complex combinations. This part neither requires nor excludes the presence of such "mixed operand" operations.

c) A complex interval datatype, or the operations on such datatypes. This part neither requires nor excludes such datatypes or operations.

d) A complex fixed point datatype, or the operations on such datatypes. This part neither requires nor excludes such datatypes or operations.

e) A complex rational datatype, or the operations on such datatypes. This part neither requires nor excludes such datatypes or operations.

f) Matrix, statistical, or symbolic operations (on suitable datatypes). This part neither requires nor excludes such operations or datatypes.

g) The properties of complex arithmetic datatypes that are not related to the numerical process, such as the representation of values on physical media.

h) The properties of integer and floating point datatypes that properly belong in programming language standards or other specifications. Examples include:

   1) the syntax of numerals and expressions in the programming language,
   2) the syntax used for parsed (input) or generated (output) character string forms for numerals by any specific programming language or library,
   3) the precedence of operators in the programming language,
   4) the rules for assignment, parameter passing, and returning value,
   5) the presence or absence of automatic datatype coercions,
   6) the consequences of applying an operation to values of improper datatype, or to uninitialised data.

Furthermore, this part does not provide specifications for how the operations should be implemented or which algorithms are to be used for the various operations.

2 Conformity

It is expected that the provisions of this document will be incorporated by reference and further defined in other International Standards; specifically in programming language standards and in binding standards.

A binding standard specifies the correspondence between one or more of the arithmetic datatypes, parameters, and operations specified in this document and the concrete language syntax of some programming language. More generally, a binding standard specifies the correspondence between certain datatypes, parameters, and operations and the elements of some arbitrary computing entity. A language standard that explicitly provides such binding information can serve as a binding standard.

When a binding standard for a language exists, an implementation shall be said to conform to this document if and only if it conforms to the binding standard. In case of conflict between a binding standard and this document, the specifications of the binding standard take precedence.

When a binding standard requires only a subset of the imaginary or complex integer or imaginary or complex floating point datatypes specified in this document, an implementation remains free to conform to this document with respect to other datatypes independently of that binding standard.

When a binding standard requires only a subset of the operations specified in this document, an implementation remains free to conform to this document with respect to other operations, independently of that binding standard.

When no binding standard between a language and some datatypes or operations specified in this document exists, an implementation conforms to this document if and only if it provides one or more datatypes and one or more operations that together satisfy all the requirements of clauses 5 through 8 that are relevant to those datatypes and operations. The implementation shall then document the binding.

Conformity to this document is always with respect to a specified set of datatypes and set of operations. Conformity to this document implies conformity to part 1 and part 2 of ISO/IEC 10967 for the integer and floating point datatypes and operations used. Under certain circumstances, conformity to IEC 60559 is implied by conformity to part 1 of ISO/IEC 10967.

An implementation is free to provide arithmetic datatypes and arithmetic operations that do not conform to this document, or that are beyond the scope of this document. The implementation shall not claim or imply conformity to this document for such datatypes or operations.

An implementation is permitted to have modes of operation that do not conform to this document. A conforming implementation shall specify how to select the modes of operation that ensure conformity.

NOTES

1 Language bindings are essential. Clause 8 requires an implementation to document a binding if no binding standard exists. See Annex **??** for an example of a conformity statement, and Annex C for suggested language bindings.

2 A complete binding for this document will include (explicitly or by reference) a binding for part 2 and part 1 as well, which in turn may include (explicitly or by reference) a binding for IEC 60559 as well.

3 This document does not require a particular set of operations to be provided. It is not possible to conform to this document without specifying to which datatypes and set of operations (and modes of operation) conformity is claimed.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60559:1989, *Binary floating-point arithmetic for microprocessor systems.*

ISO/IEC 10967-1, *Information technology – Language independent arithmetic – Part 1: Integer and floating point arithmetic.*

ISO/IEC 10967-2, *Information technology – Language independent arithmetic – Part 2: Elementary numerical functions.*