Annex B
(informative)

Implementation guidelines

B.1 Overview

This informative annex provides advisories relative to the implementation of the spatial operations contained in this document. Implementations may introduce errors of various kinds. Since the term error has many different meanings, depending on the application, a brief description of each of the various types of error referenced in this document is included in this annex. This discussion is intended to clarify the meaning of the types of errors as they relate to compliance.

B.2 Error types considered in this document

The term error has many meanings in common usage of the language. A dictionary definition might contain definitions such as:

a) the failure of a computer program to produce an anticipated result, such as a result not falling within an expected range,

b) a variation between the true value of a mathematical quantity and a calculated or measured value, or

c) a mistake as in an implementation or in the use of an implementation.

These are the error terms that are the most important to this document. The term error is often defined in terms of words that themselves have alternative meanings. When used in a scientific or technical sense a modifying adjective is often used for specificity. In this document, modifying adjectives are used to provide this specificity. In most cases the definitions of such terms are defined where used.

B.2.1 Measurement and modelling error

In many applications and in particular in geodesy, statistical models are often used to define and characterize the error in developing reference models. This process is quite detailed, but it suffices to provide a simplified example. Measurements taken on an appropriate set of points are used to develop the reference model. This process utilizes an assumed mathematical model for the shape of the Earth, usually an oblate ellipsoid or portion thereof, formulated in terms of a geocentric coordinate system with its origin at the centre of mass of the Earth. Free parameters are adjusted in the model to provide a minimum variance fit to the nominal surface of the real Earth. In this way most of the local Earth reference models (or datums), such as ORM EUROPE_1950, are developed. The root-mean-square difference between the measured points and the points computed from the reference model is termed the residual error or standard error. Other expressions of measurement error such as tolerance or maximum error or error interval are also in use.

In this document the reference models used are taken to be exact, that is, to have zero residual error. However, when specifying such reference models residual error values may be given with the reference model parameters for completeness. It is emphasized that errors associated with functional conformance in this document do not include residual errors or tolerance.

B.2.2 Implementation error

Conformance compliance in this document is focused on the notion of implementation error. Implementation error consists primarily of:

a) use of an incorrect mathematical formulation,

b) coding error such that a user error is not detected,

c) coding errors by which the mathematical formulation is incorrectly implemented,

d) excessive round-off error in the implementation of a mathematical formulation,

e) approximations used to speed up computations that cause excessive approximation error,

f) a formulation or implementation does not compensate for singularities or near singularities at some points in the valid domain of the formulation, or

g) results that lie outside a valid range not detected by the implementation.

The process of evaluating implementation errors is, itself, subject to user error including:

a) user error such as selecting the wrong Earth reference model,

b) user error in trying to employ the software outside an applicable region, and

c) user error in trying to test the software outside a valid conformance region.

B.2.3 Finite precision

It is generally not possible to exactly implement theoretical formulations on a digital computer due to limitations in representing real numbers on a finite word length computer. If x is a real number, its representation on a digital computer can be denoted as x_c. The difference between x and x_c is termed *digitization error*. There are some real numbers that can be exactly represented, but generally the digital representation is only good to a prescribed number of bits depending on the precision of the floating-point representation of the computer system used. Implementation of spatial operations can involve relatively large numbers. Loss of significance can occur in computing the differences of numbers with large absolute values and the products of relatively small numbers with large numbers.

Finite precision also can lead to excessive round-off error. The round-off error usually depends on the algorithm employed. Sometimes the round-off error can be minimized by a different algorithm design.

EXAMPLE Using single precision arithmetic for SRFs associated with the Earth may lead to a loss of precision on the order of half a metre even when the application is for the near Earth region.

NOTE To mitigate loss of precision, it is advisable to employ double precision (see ISO/IEC/IEEE 60559) arithmetic for floating-point operations.

B.2.4 Approximation error

The replacement of theoretical formulations with approximations made to increase computational efficiency introduces an error. The difference between the true value x and the approximation value x_a is the *approximation error*. The implementation of an approximation using a double precision representation includes both the digitization and approximation errors. The combination of these errors is termed the *computational error.* However, the magnitude of the approximation error usually dominates that of the digitization error and therefore the digitization error may generally be ignored.

The acceptable computational error is application dependent. Increased capabilities of real-world measurement systems and improved SRF models have led to increased requirements for more stringent error tolerances. In high-resolution simulation applications the requirement is to keep the computational error in position as small as 1 millimetre. Increased system accuracy requirements coupled with efficiency requirements place a considerable premium on development and use of efficient algorithms. Given the variability in computer system characteristics and application domain accuracy requirements there is no single solution that fits all cases. Subsequent clauses provide a set of general guidelines for algorithm designers and software developers that are intended to broaden their conceptual approach to implementations. These guidelines are specific to Earth-related spatial operations but most of them are applicable to the more general case.

B.3 General observations on implementations

In many application domains computational efficiency is very important. Some examples of such applications include: embedded systems with real time control feed-back, the processing of large numbers of very large environmental data files, real time graphics display of geographic data and large-scale simulations involving hundreds of thousands of interacting objects. Historically, computational assets were much less capable than those currently available. As a result, much research over the last century has been devoted to reducing the computational complexity for the type of spatial operations contained in this document. Many of the techniques currently used were developed for hand computation or in the context of more rudimentary computational systems. Implementers have been slow to adapt to the capabilities provided by computational systems that currently exist. Concomitant with the increased computational capabilities there have been significant technical advances in the field of computational mathematics. New methods have emerged along with better strategies for exploiting the current computational capabilities. These advances in computational mathematics have generally not been exploited for the types of spatial operations within the scope of this document.

The strategy for selecting algorithms for implementation is dependent on the intended application. For a general service system, where an interactive user needs a few spatial operations computed, efficiency is becoming much less important. Current machines are so fast that humans cannot perceive the difference between very fast machines and very slow ones. For such application domains the choice of algorithm is not critical as long as it is accurate, reliable and covers the domain of interest.

For computationally intense applications most of the mathematical formulations contained in this document are not appropriate for direct implementation. Some of the closed-form solutions may be unacceptably inefficient and may be replaced by various approximate methods.

EXAMPLE Most implementations of the inverses for map projections are implemented with finite series methods in order to avoid using potentially inefficient iterative methods.

B.4 Guidelines for algorithm development for spatial operations

B.4.1 Overview

Many computational algorithms have been developed for spatial operations processing for a wide range of applications. Many of these are not appropriate for efficient processing using current computer system environments. If an application domain does not require efficient processing, any accurate algorithm for computing spatial operations may be employed. In such cases, it is recommended that closed-form solutions be employed when available, and iterative procedures otherwise.

This clause includes a set of guidelines or advisories for use in designing *efficient* algorithms. While the target environment is generally a computer system with a super-scalar architecture, many of these advisories are applicable to legacy computer systems and specialized systems used for embedded processing. Most of the advisories are applicable to spatial operations processing for celestial bodies other than the Earth.

B.4.2 The computational environment

The properties of the computational environment should be taken into account. In recent decades a significant improvement in computational capabilities has occurred. Yet, in some application domains, algorithms that were