AS/NZS ISO 19107:2020
ISO 19107:2019

AS/NZS ISO 19107:2020

Australian/New Zealand Standard™

Geographic information — Spatial schema

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

AS/NZS ISO 19107:2020

This Joint Australian/New Zealand Standard™ was prepared by Joint Technical Committee IT-004, Geographical Information/Geomatics. It was approved on behalf of the Council of Standards Australia on 18 March 2020 and by the New Zealand Standards Approval Board on 1 April 2020.

This Standard was published on 17 April 2020.

The following are represented on Committee IT-004:

ANZLIC — the Spatial Information Council
Australian Antarctic Division
Australian Bureau of Meteorology
Australian Maritime Safety Authority
CSIRO
Curtin University of Technology
Department of Agriculture and Water Resources (Australian Government)
Department of Defence (Australian Government)
Geoscience Australia
Science New Zealand
Services Australia (Australian Government)
Spatial Industries Business Association
University of Melbourne

This Standard was issued in draft form for comment as DR AS/NZS ISO 19107:2020.

Keeping Standards up-to-date

Ensure you have the latest versions of our publications and keep up-to-date about Amendments, Rulings, Withdrawals, and new projects by visiting:

www.standards.org.au

www.standards.govt.nz

ISBN 978 1 76072 819 9

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

AS/NZS ISO 19107:2020
ISO 19107:2019

Australian/New Zealand Standard™

Geographic information — Spatial schema

Originated as AS/NZS ISO 19107:2005.
Second edition 2020.

COPYRIGHT

© ISO 2020 — All rights reserved
© Standards Australia Limited/the Crown in right of New Zealand, administered by the New Zealand Standards Executive 2020

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher, unless otherwise permitted under the Copyright Act 1968 (Cth) or the Copyright Act 1994 (New Zealand).

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

Preface

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee IT-004, Geographical Information/Geomatics, to supersede AS/NZS ISO 19107:2005, *Geographic information — Spatial schema.*

The objective of this Standard is to specify conceptual schemas for describing the spatial characteristics of geographic entities, and a set of spatial operations consistent with these schemas. It treats "vector" geometry and topology. It defines standard spatial operations for use in access, query, management, processing and data exchange of geographic information for spatial (geometric and topological) objects. Because of the nature of geographic information, these geometric coordinate spaces will normally have up to three spatial dimensions, one temporal dimension and any number of other spatially dependent parameters as needed by the applications. In general, the topological dimension of the spatial projections of the geometric objects will be at most three.

This Standard is identical with, and has been reproduced from, ISO 19107:2019, *Geographic information — Spatial schema*.

As this document has been reproduced from an International Standard, the following applies:

Australian or Australian/New Zealand Standards that are identical adoptions of international normative references may be used interchangeably. Refer to the online catalogue for information on specific Standards.

The terms "normative" and "informative" are used in Standards to define the application of the appendices or annexes to which they apply. A "normative" appendix or annex is an integral part of a Standard, whereas an "informative" appendix or annex is only for information and guidance.

Documents
Document Sharing Hub

© Standards Australia Limited/Standards New Zealand 2020

Full document details: AS NZS ISO 19107-2020

Contents

Preface ii
Foreword viii
Introduction ix
1 Scope 1
2 Normative references 1
3 Terms and definitions 1
4 Symbols, notation and abbreviated terms 17
4.1 Presentation and notation 17
4.1.1 Unified Modeling Language (UML) 17
4.1.2 Naming conventions 17
4.2 Organization 18
4.3 Abbreviated terms and symbols 18
5 Conformance 19
5.1 Requirements class conformance targets 19
5.1.1 Conformance targets 19
5.1.2 Geometry metrics (geodesy) 22
5.1.3 Topological dimensionality 22
5.1.4 Interpolation schemes 22
5.1.5 Structural complexity 23
5.1.6 Functional complexity 24
5.2 Conformance classes 24
5.3 Requirements classes 25
6 Coordinates and core geometry 26
6.1 Semantics 26
6.2 Requirements Class Coordinate 27
6.2.1 Codelists to specify capabilities 27
6.2.2 Coordinate systems for Geometry — Semantics 27
6.2.3 GeometricReferenceSurface 31
6.2.4 Interface ReferenceSystem 35
6.2.5 Codelist ReferenceSystemTypes 36
6.2.6 Interface CompoundReferenceSystem 36
6.2.7 Interface HomogeneousCoordinateSystem 37
6.2.8 Interface GeometricCoordinateSystem 37
6.2.9 Datatype DirectPosition 42
6.2.10 Union Datatype RSID 44
6.2.11 Codelist Axis 44
6.2.12 Role metadata: AxisDescription 45
6.2.13 Datatype Axis Description 45
6.2.14 Codelist SpatialAxis 45
6.2.15 Codelist SphericalAxis 45
6.2.16 Codelist TemporalAxis 45
6.2.17 Codelist ParametricAxis 46
6.2.18 Codelist Datum 46
6.2.19 Datatype Parameter 47
6.2.20 Datatype Permutation, Projection 47
6.2.21 Interface ReferenceDirection 48
6.2.22 Datatype Bearing 48
6.2.23 Codelist Rotation 50
6.2.24 Codelist RelativeDirection 50
6.2.25 Codelist FixedDirection 50
6.2.26 Codelist CurveRelativeDirection 50
6.2.27 Datatype Vector 50

AVA Documents
Document Sharing Hub
Full document details: AS NZS ISO 19107-2020

6.2.28 Interface Envelope ... 52
6.2.29 Engineering coordinate systems, Tangent spaces and local interpolations ... 53
6.3 Requirements Class Coordinate Data ... 53
6.4 Requirements Class Geometry ... 54
6.4.1 Semantics ... 54
6.4.2 Interface TransfiniteSetOfDirectPositions ... 55
6.4.3 CodeList: BoundaryType ... 55
6.4.4 Interface Geometry ... 56
6.4.5 Datatype GeometryData ... 70
6.4.6 CodeList: GeometryType ... 70
6.4.7 Interface Encoding ... 70
6.4.8 Interface Query2D ... 71
6.4.9 Interface Query3D ... 74
6.4.10 Interface Empty ... 75
6.4.11 Interface Primitive ... 76
6.4.12 Datatype PrimitiveData ... 78
6.4.13 Interface Point ... 78
6.4.14 Datatype PointData ... 80
6.4.15 Interface Orientable ... 80
6.4.16 Datatype OrientableData ... 81
6.4.17 Datatype Knot ... 82
6.4.18 Interface Curve ... 83
6.4.19 DataType CurveData ... 93
6.4.20 Interface OffsetCurve ... 93
6.4.21 Datatype OffsetCurveData ... 94
6.4.22 Interface ProductCurve ... 95
6.4.23 ProductCurveData ... 96
6.4.24 CodeList: CurveInterpolation ... 96
6.4.25 Interface Surface ... 98
6.4.26 Datatype SurfaceData ... 101
6.4.27 CodeList: SurfaceInterpolation ... 101
6.4.28 Interface Solid ... 102
6.4.29 Datatype SolidData ... 105
6.4.30 CodeList: SolidInterpolation ... 105
6.4.31 Interface Collection ... 105
6.4.32 Role element: Geometry ... 106
6.4.33 DataType CollectionData ... 107
6.4.34 Interface Complex ... 107
6.4.35 Role Complex: generator: Primitive ... 110
6.4.36 Role Complex: superComplex and subComplex ... 110
6.5 Requirements Class Geometry Data ... 111

7 Interpolations for Curves ... 111
7.1 Requirements Class Line Curve ... 111
7.1.1 Semantics ... 111
7.1.2 Interface Line ... 111
7.1.3 DataType LineData ... 113
7.2 Requirements Class Line Data ... 113
7.3 Requirements Class Geodesic Curve ... 113
7.3.1 Semantics ... 113
7.3.2 Interface Geodesic ... 114
7.3.3 DataType GeodesicData ... 115
7.4 Requirements Class Geodesic Curve Data ... 115
7.5 Requirements Class Rhumb ... 115
7.5.1 Interface Rhumb ... 115
7.5.2 DataType RhumbData ... 115
7.6 Requirements Class Rhumb Curve Data ... 116
7.7 Requirements Class Polynomial Curves ... 116
7.7.1 Semantics ... 116

AVA Documents
Document Sharing Hub
Full document details: AS NZS ISO 19107-2020

7.7.2 Interface RealFunction ... 117
7.7.3 Interface FunctionArc ... 117
7.7.4 Association Role function ... 117
7.7.5 Interface FunctionCurve ... 118
7.7.6 Interface RealPolynomial ... 118
7.7.7 Interface PolynomialArc ... 119
7.7.8 Datatype PolynomialArcData ... 120
7.7.9 Interface PolynomialCurve ... 120
7.7.10 DataType PolynomialCurveData ... 120
7.8 Requirements Class Polynomial Curve Data ... 120
7.9 Requirements Class Conic Curves ... 121
7.9.1 Semantics ... 121
7.9.2 Interface Arc ... 122
7.9.3 Datatype ArcData ... 123
7.9.4 Interface Circle ... 124
7.9.5 Interface Conic ... 124
7.9.6 Interface EllipticArc, Datatype EllipticArcData ... 127
7.10 Requirements Class Conic Curve Data ... 127
7.11 Requirements Class Spiral Curve ... 127
7.11.1 Semantics, Mathematical background: curves and curvature ... 127
7.11.2 Interface Spiral Curves ... 133
7.11.3 Interface Clothoid Curve ... 135
7.11.4 Datatype SpiralData ... 135
7.12 Requirements Class Spiral Curve Data ... 135
7.13 Requirements Class Spline Curve ... 135
7.13.1 Semantics ... 135
7.13.2 CodeList: KnotType ... 136
7.13.3 CodeList: SplineCurveForm ... 137
7.13.4 Interface SplineCurve ... 137
7.13.5 Interface PolynomialSpline ... 140
7.13.6 Interface CubicSpline ... 141
7.13.7 Interface Bezier ... 142
7.13.8 Interface BSplineCurve (and NURBS) ... 143
7.13.9 DataType BsplineData ... 144
7.14 Requirements Class Spline Curve Data ... 144

8 Interpolations for Surfaces ... 144
8.1 Requirements Class Polygon Surface ... 144
8.1.1 Semantics ... 144
8.1.2 Interface Polygon ... 144
8.1.3 Datatype PolygonData ... 146
8.1.4 Interface PolyhedralSurface ... 146
8.1.5 Datatype PolyhedralSurfaceData ... 146
8.1.6 Interface Triangle ... 146
8.1.7 Datatype TriangleData ... 147
8.1.8 Interface TriangulatedSurface ... 147
8.1.9 Datatype TriangulatedSurfaceData ... 147
8.2 Requirements Class Polygon Surface Data ... 147
8.3 Requirements Class Parametric Curve Surface ... 147
8.3.1 Semantics ... 147
8.3.2 Interface ParametricCurveSurface ... 148
8.3.3 Datatype ParametricCurveSurfaceData ... 151
8.3.4 Interface BilinearGrid ... 151
8.3.5 Extensions of ParametricCurveSurface ... 152
8.4 Requirements Class Parametric Curve Surface Data ... 153
8.5 Requirements Class Conic Surface ... 153
8.5.1 Semantics ... 153
8.5.2 Interface Sphere ... 153
8.5.3 Interface Cone ... 154

AVA Documents
Document Sharing Hub
Full document details: AS NZS ISO 19107-2020

8.5.4 Interface Cylinder ... 154
8.6 Requirements Class Conic Surface Data ... 155
8.7 Requirements Class Spline Surface ... 155
8.7.1 Semantics ... 155
8.7.2 Interface BSplineSurface (and NURBS) ... 155
8.7.3 Codelist BSplineSurfaceForm ... 157
8.8 Requirements Class Spline Surface Data ... 157

9 Interpolations for Solids ... 157
9.1 Requirements Class Boundary Representation Solid ... 157
9.2 Requirements Class Boundary Representation Solid Data ... 158
9.3 Requirements Class Parametric Curve Solid ... 158
9.3.1 Interface ParametricCurveSolid ... 158
9.3.2 Interface BSolidSpline ... 159
9.3.3 Other interpolations ... 160
9.4 Requirements Class Parametric Curve Solid Data ... 160

10 Topology ... 160
10.1 Requirements Class Topology root ... 160
10.1.1 Semantics ... 160
10.1.2 Interface Topology ... 161
10.1.3 Interface Primitive ... 165
10.1.4 Interface DirectedTopo ... 167
10.1.5 Datatype TopologyData ... 170
10.1.6 DataType PrimitiveData ... 170
10.1.7 DataType ComplexData ... 170
10.1.8 Datatype Expression ... 171
10.1.9 Datatype ExpressionTerm ... 173
10.2 Requirements Class Topology Root Data ... 173
10.3 Requirements Class Node ... 173
10.3.1 Semantics ... 173
10.3.2 Interface Node ... 173
10.3.3 Interface DirectedNode ... 174
10.4 Requirements Class Edge ... 175
10.4.1 Interface Edge ... 175
10.4.2 Interface DirectedEdge ... 176
10.5 Requirements Class Face ... 176
10.5.1 Semantics ... 176
10.5.2 Interface Face ... 176
10.5.3 Interface DirectedFace ... 177
10.6 Requirements Class Topology Solid ... 177
10.6.1 Interface Solid ... 177
10.6.2 Interface DirectedSolid ... 178
10.7 Requirements Class Topological Complex ... 178
10.7.1 Semantics ... 178
10.7.2 Interface Complex ... 178
10.8 Requirements Class Derived Topological Relations ... 181
10.8.1 Introduction ... 181
10.8.2 Canonical form for Geometry ... 182
10.8.3 Boundary operators for aggregate objects ... 183
10.8.4 Boolean or set operators ... 184
10.8.5 Egenhofer operators ... 185
10.8.6 Full topological operators ... 186
10.8.7 Combinations ... 189

11 Special Requirements Classes ... 189
11.1 Requirements Class Simplicial geometry ... 189
11.1.1 Semantics ... 189
11.1.2 Datatype Simplex ... 190
11.1.3 DataType SimplicialTerm ... 192

Documents
Document Sharing Hub
Full document details: AS NZS ISO 19107-2020

11.1.4 DataType::SimplicialPolynomial 192
11.1.5 DataType::SimplicialComplex 192
11.2 Requirements Class Point Clouds 192
11.2.1 Semantics 192
11.2.2 Interface PointCloud 193

Annex A (normative) **Abstract test suite** **195**

Annex B (informative) **Examples for application schemas** **210**

Annex C (informative) **MiniTopo** **214**

Annex D (informative) **Crosswalk 19107:2003 to current version** **219**

Bibliography **222**

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing documents is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*.

This second edition cancels and replaces the first edition (ISO 19107:2003), which has been technically revised. The main changes compared to the previous edition are as follows:

— It now forms a logical subset of this second edition. In other words, this document is 100 % backwardly compatible with its previous version, ISO 19107:2003, except in a few areas (in NURBS) where the previous version contained technical errors that are corrected in this revision.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

Introduction

This document provides conceptual schemas for describing, representing and manipulating the spatial characteristics of geographic entities. Standardization in this area is the cornerstone for other geographic information design, specification and standardization.

"Vector" data consists of geometric primitives used to construct expressions of the spatial characteristics of geographic features. "Raster" data is based on the division of the extent covered into small units according to a tessellation of the space. This document deals only with vector data.

There is a hierarchy of complexity in the "geometry" of the underlying object used in various coordinate systems. These may use reference planes (map geometry – Euclidean), reference spheres (spherical geometry — using spherical trigonometry), reference ellipsoids (ellipsoidal geometry using Gaussian or Riemannian metrics) or more complex surfaces (usually using numeric approximations for calculation). The coordinates of a point locate it on, or in relation to, the reference geometry. With the exception of "map geometry," the usual Euclidean formulae for distance and area do not apply directly in the coordinate system.

Topology expressions provide qualitative descriptions of the spatial relations between geometry objects. Topology deals with the characteristics of geometric figures that remain invariant if the space is deformed elastically. Topological properties do not change when information is transformed from one coordinate system to another, usually including the coordinate function that map from R2 or R3 to the reference geometry. Topological properties in the domain of the coordinate system will be identical to those on the geographic surface; but the metric properties may change significantly (e.g. distance, area, direction).

Spatial operators are functions and procedures that use, query, create, modify or delete spatial objects. This document defines the taxonomy of some of the more important operators, their definitions and implementations. The goals are to:

— Define spatial operators unambiguously, so that different implementations will yield comparable results within the limitations of accuracy and resolution.

— Use these definitions to define a set of standard operations that will form the basis of compliant systems and thus act as a test-bed for implementers and a benchmark set for validation of compliance.

— Define an operator algebra that will allow combinations of the base operators to be used predictably in the query and manipulation of geographic feature data.

Standardized conceptual schemas for spatial characteristics will increase the ability to share geographic information between applications. These schemas will be used by geographic information system and software developers and users of geographic information to provide consistently understandable spatial data structures and functions.

This document is technical because geometry is a technical topic. Euclid was speaking of a simpler form of geometry to the most powerful man in his world when he said:

There is no royal road to geometry (μή εἶναι βασιλικήν ἀτραπόν ἐπί γεωμετρίαν).

Euclid to Ptolemy I Soter (General with Alexander the Great, Pharaoh of Egypt) —

Attributed by Proclus (412–485 AD) in Commentary on the First Book of Euclid's Elements

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

NOTES

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

Australian/New Zealand Standard

Geographic information — Spatial schema

1 Scope

This document specifies conceptual schemas for describing the spatial characteristics of geographic entities, and a set of spatial operations consistent with these schemas. It treats "vector" geometry and topology. It defines standard spatial operations for use in access, query, management, processing and data exchange of geographic information for spatial (geometric and topological) objects. Because of the nature of geographic information, these geometric coordinate spaces will normally have up to three spatial dimensions, one temporal dimension and any number of other spatially dependent parameters as needed by the applications. In general, the topological dimension of the spatial projections of the geometric objects will be at most three.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

ISO 19103, *Geographic information — Conceptual schema language*

ISO 19108, *Geographic information — Temporal schema*

ISO 19109, *Geographic information — Rules for application schema*

ISO 19111, *Geographic information — Spatial referencing by coordinates*

ISO/IEC 11404:2007, *Information technology — General-Purpose Datatypes (GPD)*

ISO/IEC 19505-2:2012, *Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 2: Superstructure*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 11404, ISO 19103, ISO/IEC 19505-2 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

NOTE Common words from geometry, such as point, curve, line, surface solid, etc., take the common meanings unless they are used as classifier names (usually interfaces), in which case they are a digital representation of the geometric concept. Common mathematical terms that are not defined here take on their common meanings in mathematics (see [15], [10], ISO/IEC 11404 or a standard text on the topic, such as the "N. Bourbaki"[1)] series currently published by Springer Verlag, in French, English and German). Care should be taken since mathematical terms can be context sensitive, and can easily be confused with common words. For example, "open" set, "closed" curve, "closed" set, "rational" function, "boundary," "interior," "closure", "exterior", "function" and others from common language but have very specific meanings in mathematics and in this document. Where necessary to prevent confusion, existing definitions have been elaborated to make their intent in this document explicit. Mathematical terms include common vocabulary from geometry, topology, calculus, geodesy and differential geometry. Many of these terms can be sufficiently common that inclusion is not necessary. They are included here to prevent confusion especially for terms like the ones listed above that have another meaning in another context.

1) N. Bourbaki is the pen-name for the "Association des collaborateurs de Nicolas Bourbaki" (Association of Collaborators of Nicolas Bourbaki) of mathematicians first published in 1935 and dedicated to "formalizing" mathematics. The group has an office at the "École Normale Supérieure" in Paris. See https://en.wikipedia.org/wiki/N.

© Standards Australia Limited/Standards New Zealand 2020

Document Sharing Hub
Full document details: AS NZS ISO 19107-2020

3.1
abstract root
<programming> common root classifier of a category which is a superclass of any other classifier in the category

Note 1 to entry: The class Any in some programming languages is the abstract root of all classes. Thus, it is the *de facto* union of all classes. In this document, Geometry is the (named and explicit) abstract root for all geometry objects. In the package Geometry and any of its subpackages (including those in its Requirements Classes), any interface will be a subtype of Geometry either directly or transitively.

3.2
arc
<geometry> *segment* (3.83) of a curve

3.3
barycentric coordinates
<coordinate geometry> point in a **n-dimension** coordinate system using n+1 numbers, $[u_0, u_1, u_2, u_3, \ldots, u_n] \ni \left[[0 \le u_i \le 1] \wedge \sum u_i = 1{,}0\right]$, in which the location of a point of an **n-simplex** (of any dimension) is specified by a weighted centre of mass of equal masses placed at its vertices using vector algebra of the $\mathbb{R}^n$ used in the **coordinate reference system**

Note 1 to entry: Even though there are n+1 coordinates in a barycentric coordinate system, the topological dimension is n, since the restriction (sums to 1,0) loses 1 degree of freedom (once you have n ordinates, the remaining one is determined such as $u_n = 1{,}0 - \sum_{i=0}^{n-1} u_i$. The coordinates for the simplex are all non-negative, but the system can be extended outside of the simples by using negative numbers. If the ordinates are all positive, then the point is inside (interior to) the **n-simplex**. If one of them is 1,0 and the other 0, this is a corner of the simplex. If one of them is zero and the others still each greater than or equal to zero, the point is on the n–1-simplex opposite the vertex zeroed out. If any are negative, the point is outside of the simplex. The coordinates are dependent on the underlying coordinate reference system of the source data.

3.4
bearing
horizontal angle, tangent or direction at a point

Note 1 to entry: This definition (as opposed to the one in ISO 19162:2015) is required for this document because the concept is used in other definitions, such as first geodetic problem and second geodetic problem. The two definitions are nearly equivalent because the tangent of a curve on a surface is a tangent to the surface and does specify a direction. Usual 2D measure of bearing can be an angle equivalently measured from North clockwise, or a unit tangent vector. If the coordinate system is spatially 3D, the horizontal bearing angle may also need to a vertical altitude angle to be complete. If a reference curve (as used in ISO 19162) is parameterized by arc length, then the "derivative" is a unit vector. If another parameterization "*t*" is used, then the derivative should be normalized $\left(\vec{\tau} / \|\vec{\tau}\|; \dot{c}(t) = \vec{\tau};\right)$. This is useful, since parameterization by arc length can be computationally difficult. The numeric representation of a vector depends on the coordinate system. The bearing is not dependent on a coordinate system, but it can be represented in any reasonable system. The bearing is not dependent on its various representations.

3.5
bicontinuous
<mathematics> invertible, continuous and with a continuous inverse

3.6
boundary
set that represents the limit of an entity

Note 1 to entry: Boundary is most commonly used in the context of geometry, where the set is a collection of points or a collection of objects that represent those points. In other arenas, the term is used metaphorically to describe the transition between an entity and the rest of its domain of discourse.

Nicolas Bourbaki. Their books are held in high regard by the mathematical community.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.7
buffer
geometric object containing all points and only those points whose distance from a specified geometric object is less than or equal to a given distance use in its construction

3.8
closure
union of the interior and boundary of a topological object or geometric object

3.9
coboundary
set of topological primitives of higher topological dimension associated with a particular topological object, such that this topological object is in each of their boundaries

Note 1 to entry: If a node is on the boundary of an edge, that edge is on the coboundary of that node. Any orientation parameter associated with one of these relations would also be associated with the other. The coboundary of a node can be called a "node star".

3.10
conformal, adj.
angle-preserving

Note 1 to entry: Some projections are conformal. For example, a Mercator preserves the angle between curves, so that if two curves in a Mercator projected plane cross at a 90°, then the preimage curves on the ellipsoid also cross at 90°, such as lines of constant latitude and lines of constant longitude.

3.11
connected
property of a topological space implying that only the entire space or the empty set are the only subsets which are both open and closed

Note 1 to entry: The formal definition of connected is that any pair of locally open sets whose union is the entire space must have a non-empty intersection.

a topological space T is connected if and only if

$$[\forall X, Y \subset T \ni X \cup Y = T] \Rightarrow [X \cap Y \neq \varnothing] \quad (1)$$

This formal definition is difficult to use. The term *path connected* (3.75), defined below is equivalent for the purposed of this document. The use or “finite precision” coordinates makes sets which are connected but not path connected impossible to represent. In all cases “connected” is used, but “path connected” is easier to test and to visualize.

3.12
connected node
<topology> node that starts or ends one or more edges

3.13
control point
<coordinate geometry> point used in the construction of a geometry that partially controls its shape but does not necessarily lie on the geometry

Note 1 to entry: A centre of an arc is a control point; poles in b-spline curves are control points.

3.14
convex
<geometry> containing all points on a "line" joining two interior points

Note 1 to entry: The definition of convex requires a definition of line. For coordinate systems, this is the usual linear interpolated arc, but in context, the "line" on a geometric reference surface will be a "geodesic arc". The default in this document is the linear interpolate.

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.15
convex hull
<geometry> smallest convex set containing a given geometric object

Note 1 to entry: "Smallest" is the set theoretic smallest, not an indication of a measurement. The definition can be rewritten as "the intersection of all convex sets that contain the geometric object". Another definition in a Euclidean space $\mathbb{E}^n$ is the union of all lines with both end points in the given geometric object.

$$C = A.convexHull \Leftrightarrow$$
$$[C.convex = TRUE] \wedge [A \subset C] \wedge [[B.convex = true, A \subset B] \Rightarrow [A \subseteq C \subseteq B]] \quad (2)$$

3.16
coordinate
one of a sequence of numbers designating the position of a point

Note 1 to entry: In a coordinate reference system, the numbers are qualified by units. The number of offsets (generally called "ordinates") in a coordinate is not the dimension. If there is a constraint, the dimension can be smaller. See *coordinate dimension* (3.17).

3.17
coordinate dimension
<coordinate geometry> number of separate decisions needed to describe a position in a coordinate system

Note 1 to entry: The coordinate dimension represents the number of choices made, and constraints can restrict choices. A barycentric coordinate which has (n+1)-offsets, but the underlying space is dimension n. Homogeneous coordinates (*wx*, *wy*, *wz*, *w*) are actually 3 dimensional because the choice of "*w*" does not affect the position, i.e. (wx, wy, wz, w) = (x y, z, 1) → (x, y, z) which is not affected by w. The dimension will be at most the count of the numbers in the coordinate, but it can be less if the coordinates are constrained in some manner.

3.18
coordinate reference system
<differential geometry, geodesy> coordinate system that is related to an object by a datum

[SOURCE: ISO 19111:2007, 4.8, modified — Notes 1 and 2 to entry have been added]

Note 1 to entry: The original definition of coordinate reference system (CRS) uses a geometric object (a geodetic datum) which is referenced to the real world by a Datum. ISO 19111 can extend this to any "parameter" which essentially can be represented as the graph of the parameter relation. This graph is in the cross product of the Datum, the domain space of the parameter function, and the parameter space (which in turn may be multi dimensional). Thus, every coordinate system used in this document is logically a CRS. Since this was not the original intent of ISO 19111, this document will use CRS for coordinate systems associated to a geodetic datum, and use the more general term "coordinate system" for anything else that is either more or less. Here, CRS means spatial coordinates. In every case where the underlying datum (surface) is not flat, the coordinate system is not Euclidean, and the metric is not Pythagorean.

Note 2 to entry: In the event that later versions of ISO 19111 change the definition as copied above, this document should not be affected. In the case that dynamic datums are used, the measurements made by the geometry operations of the objects in this document would only be valid at the time used for the datum and the associated coordinates. If the datum varies, the measure may also vary, but the definitions of the measures remains the same, but the actual values may be a function of time.

3.19
curvature vector
<differential geometry> second derivative of a curve parameterized by arc length, at a point

Note 1 to entry: If $c(s) = (x(s), y(s), z(s))$ is a curve in 3D Cartesian space ($\mathbb{E}^3$), and s is the arc length along $c(s)$, then the unit tangent vector is $\dot{c}(s) = (\dot{x}(s), \dot{y}(s), \dot{z}(s))$, i.e. the derivative of the coordinate values of "c" with respect to "s". The curvature vector is $\ddot{c}(s) = (\ddot{x}(s), \ddot{y}(s), \ddot{z}(s))$. The curvature vector can be approximated by the inverse of the radius of a circle through any 3 nearby points on the curve (pointed from the curve to towards the centre of the circle)

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub
Full document details: AS NZS ISO 19107-2020

3.20
cycle
<geometry, topology> bounded spatial object with an empty boundary

Note 1 to entry: Cycles are used to describe boundary components. A cycle usually has no boundary because it closes on itself, but it is bounded (i.e., it does not have infinite extent). A circle or a sphere, for example, has no boundary (i.e., its boundary is empty), but is bounded.

3.21
data point
<coordinate geometry> point that lies on the geometry

Note 1 to entry: The vertices in a line string are data points, the points used to construct a polynomial spline are data points. Data points can be used as control points, but are often derived after the geometry is constructed.

3.22
diameter
<metric> maximum distance between two points in the set of points

3.23
directed edge
<topology> directed topological object that represents an association between an edge and one of its orientations

Note 1 to entry: A directed edge that is in agreement with the orientation of the edge has a + orientation, otherwise, it has the opposite (–) orientation. Directed edge is used in topology to distinguish the right side (–) from the left side (+) of the same edge and the start node (–) and end node (+) of the same edge and in computational topology to represent these concepts.

3.24
directed face
<topology> directed topological object that represents an association between a face and one of its orientations

Note 1 to entry: The orientation of the directed edges that compose the exterior boundary of a directed face will appear positive from the direction of this vector; the orientation of a directed face that bounds a topological solid will point away from the topological solid. Adjacent solids would use different orientations for their shared boundary, consistent with the same sort of association between adjacent faces and their shared edges. directed faces are used in the coboundary relation to maintain the spatial association between face and edge.

3.25
directed node
<topology> directed topological object that represents an association between a node and one of its orientations

Note 1 to entry: Directed nodes are used in the coboundary relation to maintain the spatial association between edge and node. The orientation of a node is with respect to an edge, "+" for end node, "–" for start node. This is consistent with the vector notion of "result = end – start".

3.26
directed solid
<topology> directed topological object that represents an association between a topological solid and one of its orientations

Note 1 to entry: Directed solids are used in the coboundary relation to maintain the spatial association between face and topological solid. The orientation of a solid is with respect to a face, "+" if the upNormal is outward, "–" if inward. This is consistent with the concept of "up = outward" for a surface bounding a solid.

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.27
distance
<geometry, metric spaces> minimal length of a curve that joins the two points or geometries

Note 1 to entry: The usual distance function for two points in a coordinate space assumes an underlying plane and is a Euclidean distance. If the underlying Reference Surface is not a plane, then distance is defined by this minimum length of all curves between the two points. These surfaces are prime examples of non-Euclidean geometry, where the parallel postulate in Euclid's Elements does not hold. In mathematical terms, distance is the "greatest lower bound" of the length of the curves. The word minimum is sometimes used, but there should be no expectation that an instance of that minimum actually occurs, only that any larger number will have a length in the set that is smaller.

3.28
domain
<mathematics> well-defined set

Note 1 to entry: Domains are used to define the "domain and range" of operators and functions. If a function "f" maps values from the source domain "X" to values in the target domain "Y," then the notation is "f:X→Y." The relationship between the function and its source domain is referred to as its "domain." The relationship between the function and its target domain is referred to as its "range", "image" or alternatively "codomain". See the definition of "function" and "mapping" in [15].

3.29
edge
<topology> 1-dimensional topological primitive

Note 1 to entry: The geometric realization of an edge is a curve. The boundary of an edge is the set of one or two nodes associated with the edge within a topological complex.

3.30
edge-node graph
<topology> graph embedded within a topological complex composed of all of the edges and connected nodes within that complex

Note 1 to entry: The edge-node graph is a subcomplex of the complex within which it is embedded.

3.31
ellipsoid
<geodesy> geometric reference surface embedded in 3D Euclidean space represented by an ellipsoid of revolution where the rotation is about the polar axis

Note 1 to entry: For the Earth the rotation is about the polar axis. This results in an oblate ellipsoid with midpoint of the foci located at the nominal centre of the Earth.

Note 2 to entry: The two usual algorithms for latitude on an ellipsoid and on a sphere (such as used in spherical coordinates) are only equivalent if the ellipsoid is a sphere, having all radii equal in all directions. The problem is that a radial line from the centre of a general ellipsoid does not always cross the surface of the ellipsoid orthogonally. In general, planar slices through the centre do not intersect the surface orthogonally, and therefore the curves that correspond to the great circles of a sphere are not geodesics on the ellipsoid.

Note 3 to entry: The topology of the ellipsoid is inherited from the $\mathbb{E}^3$ space in which it is embedded. The difference is that metrics such as distance and direction on the ellipsoid are restricted to curves wholly on the surface and vectors tangent to the surface.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.32
empty set
∅
<mathematics> set without any elements

Note 1 to entry: Sets are equal if they contain exactly the same elements. Since any two empty sets would share exactly the same contained elements (by definition none), they are, by definition, equal. The empty set (∅) can be considered a geometric entity, because all the elements it contains are points. This is a vacuous statement since the set ∅ contains no elements, and therefore the "for all" statement has nothing to test and is thus true in each of its non-existent cases. There are a lot of true but vacuous statements in proofs about ∅. This confuses some programmers since many systems use type safe sets, in which the class of the entities determines a class for the container set. The math does not care about "class" and only sees sets; so that an empty set of aardvarks and an empty set of zebras in mathematics are (is?) the same set. The other confusion is that ∅ is not the database Null introduced by Codd and used in relational and other query languages in 3 valued logic. Null means unknown and many statements involving Null are undecidable (neither provably true nor provably false). The empty set is not "lack of knowledge" but certainty in the nonexistence of elements in the set. Most statements beginning "for all elements in ∅" are true, but vacuous. Most statements beginning "there exist an element in ∅" are always categorically false. It is almost impossible to construct an undecidable statement about ∅. Null and ∅ are not related. "Void" can mean "invalid" or "completely empty."

3.33
end node
<topology> node in the boundary of an edge that corresponds to the end point of that edge

3.34
end point
<geometry, topology> last point of a curve

3.35
error budget
<metric> statement of or methodology for describing the nature and magnitude of the errors which affect the results of a calculation

Note 1 to entry: In the most usual case, error budgets in this document describe metric calculations using representational geometry objects to estimate real-world metrics, such as distance and area.

3.36
exponential map
<differential geometry> function that maps tangent vectors at a point to end point of geodesic beginning at that point with an exit bearing equal to that of the vector and a length equal to that of the vector

Note 1 to entry: See first geodetic problem for an explanation of the process of calculating this mapping.

3.37
exterior
<geometry, topology> difference between the universe and the closure

Note 1 to entry: The concept of exterior is applicable to both topological object and geometric object.

3.38
face
<topology> 2-dimensional topological primitive

Note 1 to entry: The geometric realization of a face is a surface. The boundary of a face is the set of directed edges within the same topological complex that are associated with the face via the boundary relations. These can be organized as rings.

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.39
first geodetic problem
direct geodetic problem
<differential geometry, geodesy> problem that given a point on a surface and the direction and distance from that point to a second point along a geodesic, determines that second point

Note 1 to entry: This "problem" defines a mapping from the vector space at a point (each vector given by a direction and a length) to points of the Figure of Earth that satisfy the problem for that direction and distance. For example, if we fix the distance "r" and take all directions, the resultant geometry is the circle centred at the original point of radius "r". This document will make heavy use of this mapping; see exponential map and the second geodetic problem.

3.40
footprint
2D extent or projection of a 3D object on a horizontal surface

3.41
function
<mathematics, programming> rule that associates each element from a domain ("source domain," or "domain" of the function) to a unique element in another domain ("target domain," "co-domain," or "range" of the function)

3.42
free function
<mathematics, programming> function in an object-oriented programming language not associated to any object class

3.43
geodesic
geodesic line
<differential geometry, geodesy> curve on a surface with a zero-length tangential curvature vector

Note 1 to entry: A geodesic's curvature vector is perpendicular the surface thus has the minimum curvature of any curve restricted to the surface. This is often defined as a minimal distance curve between two points, but this does not always suffice, since some points (especially on ellipsoids and spheres) are often joined by more than one geodesic. For example, on an ellipsoid the points with $(\varphi, \lambda) = (0, 0)$ and $(0, 180)$ are joined by four separate geodesic [2 polar (the shorter) and 2 equatorial]. The exponential map is only guaranteed to be one-to-one for a small area (depending on where the centre is and how the surface is curved).

3.44
geodesic circle
<differential geometry, geodesy> set of points an equal distance from a given point (on the datum)

Note 1 to entry: The geodesic circles centred on a pole (either one) are the lines of constant latitude. Circles in a tangent space centred on the origin (corresponding to the point of tangency) map to geodesic circles by the exponential map on the geometric reference surface centred on the point of tangency.

3.45
geometric aggregate
collection of geometric objects that has no internal structure

Note 1 to entry: No assumptions about the spatial relationships between the elements can be made.

3.46
geometric boundary
boundary represented by a set of geometric primitives that limits the extent of a geometric object

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.47
geometric complex
set of disjoint geometric primitives where the boundary of each geometric primitive can be represented as the union of other geometric primitives of smaller dimension within the same set

Note 1 to entry: The geometric primitives in the set are disjoint in the sense that no point is interior to more than one geometric primitive. The set is closed under boundary operations, meaning that for each element in the geometric complex, there is a collection (also a geometric complex) of geometric primitives that represents the boundary of that element. Recall that the boundary of a point (the only 0D primitive object type in geometry) is empty. Thus, if the largest dimension geometric primitive is a solid (3D), the composition of the boundary operator in this definition terminates after at most three steps. It is also the case that the boundary of any object is a cycle.

3.48
geometric dimension
<geometry, topology> largest number *n* such that each point in a set of points can be associated with a subset that has that point in its interior and is topologically isomorphic to $\mathbb{E}^n$, Euclidean *n*-space

Note 1 to entry: Curves, because they are continuous images of a portion of the real line, have geometric dimension 1. Surfaces cannot always be mapped to R^2 in their entirety, but around each point position, a small neighborhood can be found that resembles (under continuous functions) the interior of the unit circle in R^2, and are therefore 2-dimensional. In this document, most surfaces (instances of Surface) are mapped to portions of R^2 by their defining interpolation mechanisms.

3.49
geometric object
<geometry> spatial object representing a geometric set

Note 1 to entry: A geometric object consists of a geometric primitive, a collection of geometric primitive, or a geometric complex treated as a single entity. A geometric object may be the spatial representation of a feature object.

3.50
geometric primitive
<geometry> geometric object representing a single, connected, homogeneous (isotropic) element of space

Note 1 to entry: Geometric primitives are non-decomposed objects that present information about geometric configuration. They include points, curves, surfaces, and solids. Many geometric objects behave like primitives (supporting the same interfaces defined for geometric primitives) but are actually composites composed of some number of other primitives. General collections may be aggregates and incapable of acting like a primitive (such as the lines of a complex network, which is not connected and thus incapable of being traceable as a single line). By this definition, a geometric primitive is topological open, since the boundary points are not isotropic to the interior points. Geometry is assumed to be closed. For points, the boundary is empty.

3.51
geometric realization
<geometry, topology> geometric complex where the geometric primitives are in a 1-to-1 correspondence to the topological primitives of a topological complex, such that the boundary relations in the two complexes agree

Note 1 to entry: In such a realization, the topological primitives are considered to represent the interiors of the corresponding geometric primitives even though the primitives themselves are closed.

3.52
geometric reference surface
<geometry> surface in some Euclidean space, usually $\mathbb{E}^3$, that represents an approximation to the surface of the Earth possibly restricted to a small area but often covering the entire globe

Documents
Document Sharing Hub
© Standards Australia Limited/Standards New Zealand 2020
Full document details: AS NZS ISO 19107-2020

3.53
geometric set
<geometry> set of points

Note 1 to entry: This set in most cases is infinite, except where the set consists of a list of point locations. Curves, surfaces and volumes being continuous are infinite sets of points. Some systems will define 'degenerate' curves, which are actually points.

3.54
inner product
<vector geometry> bilinear, symmetric function from pairs of vectors $\left\langle \vec{v}_1, \vec{v}_2 \right\rangle \to \mathbb{R}$ to a real number such that $\left\langle \vec{v}, \vec{v} \right\rangle = \left\| \vec{v} \right\|$ and $\left\langle \vec{v}_1, \vec{v}_2 \right\rangle = \left\| \vec{v}_1 \right\| \left\| \vec{v}_2 \right\| \cos(\theta)$ where "θ" is the angle between $\vec{v}_1$ and $\vec{v}_2$.

Note 1 to entry: Inner products in differential geometry are used on the differentials that make up the local tangent spaces. In this document, this will usually be vectors tangent to a datum surface embedded a geocentric $\mathbb{E}^3$ Euclidean/Cartesian space.

3.55
isolated node
<topology> node not related to any edge

Note 1 to entry: Isolated nodes are usually not discussed in most algebraic or combinatorial topology. In a network, they are not interesting because they cannot be a part of any path. Algebraic topology always uses topological systems there the boundary of any object of dimension" n" is always a union of a set of objects of dimension "n–1". Which is why the dimension of the empty set is "–1".

3.56
isomorphic
<mathematics> having an isomorphism

3.57
isometry
<mathematics> mapping between metric spaces that preserves the metric

Note 1 to entry: The formal definition is $f : X \to Y \ni \forall x, y \in X . distance(x, y) = distance(f(x), f(y))$.

3.58
isomorphism
<mathematics> relationship between two domains (such as two complexes) such that there are 1-to-1, structure-preserving functions from each domain onto the other, and the composition of the two functions, in either order, is the corresponding identity function

Note 1 to entry: A geometric complex is isomorphic to a topological complex if their elements are in a 1-to-1, dimension- and boundary-preserving correspondence to one another.

$$[A, B \ isomorphic] \Leftrightarrow [\exists f : A \to B, g : B \to A \ni f \circ g = Id_A, g \circ f = Id_B] \quad (3)$$

3.59
homomorphism
<mathematics> relationship between two domains such that there is a structure-preserving function from one to the other

Note 1 to entry: A homomorphism is distinct from a isomorphism in that no inverse function is required. In an isomorphism, there are two homomorphisms that are functional inverses of one another. Continuous functions are topological homomorphisms because they preserve "topological characteristics". The mapping of topological complexes to their geometric realizations preserves the concept of boundary and is therefore a homomorphism.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub
Full document details: AS NZS ISO 19107-2020

3.60
maximum
max
least upper bound
lub
<mathematics> smallest element larger than or equal to all elements of a set contained in an ordered domain (≤)

Note 1 to entry: $\left[\forall a \in A \Rightarrow \max(A) \geq a\right] \Rightarrow \left[\forall b \ni \left[\left(b \ni \left[\forall a \in A \Rightarrow \mathrm{b} \geq a\right] \Rightarrow \left[\max(A) \leq b\right]\right]\right.\right.$. Any number is an upper bound of ∅ (empty set) as a set of numbers, because any given number is greater than any number in ∅ (an admitted vacuous statement since there is no number in ∅, but true nonetheless). This means that the max (∅) must be smaller than any number; thus "−∞".

3.61
metric operation
measure
operations associated to measurements

Note 1 to entry: Generically, a metric is a standard of measure of any kind. The Greek word metron, meaning "measure," giving us "metr". Latin metricus and Greek metrikós "of, relating to measuring". See *metric unit* (3.62).

3.62
metric unit
unit of measure

Note 1 to entry: See *metric operation* (3.61).

3.63
minimum
min
greatest lower bound
glb
<mathematics> largest element smaller than or equal to all elements of a set contained in an ordered domain (≤)

Note 1 to entry: $\left[\forall a \in A \Rightarrow \min(A) \leq a\right] \Rightarrow \left[\forall b \ni \left[\left(b \ni \left[\forall a \in A \Rightarrow \mathrm{b} \leq a\right] \Rightarrow \left[\min(A) \geq b\right]\right]\right.\right.$ Any number is a lower bound of ∅ considered as a set of numbers, because any given number is less than any number in ∅ (an admitted vacuous statement since there is no number in ∅, but true nonetheless). This means that the min(∅) must be greater than any number; thus "+∞".

3.64
monotonic
<mathematics> never increasing or never decreasing

Note 1 to entry: An increasing sequence never gets smaller. A strictly increasing sequence always gets larger. A decreasing sequence never gets larger. A strictly decreasing sequence always gets smaller.

$$\begin{aligned}
[x(i) \text{ is increasing}] &\Leftrightarrow [i < j \Rightarrow x(i) \leq x(j)] \\
[x(i) \text{ is strickly increasing}] &\Leftrightarrow [i < j \Rightarrow x(i) < x(j)] \\
[x(i) \text{ is decreasing}] &\Leftrightarrow [i < j \Rightarrow x(i) \geq x(j)] \\
[x(i) \text{ is strickly decreasing}] &\Leftrightarrow [i < j \Rightarrow x(i) > x(j)] \\
[x(i) \text{ is decreasing}] \vee [x(i) \text{ is increasing}] &\Leftrightarrow \left[x(i) \text{ is monotonic}\right]
\end{aligned} \tag{4}$$

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.65
n-disc
<topology, geometry> geometry isomorphic to the set of points X in $\mathbb{E}^n$ such that $\|X\| \leq 1$; set of all points in $\mathbb{E}^n$ less than or equal to one-unit distance from the origin

Note 1 to entry: The 0-disc is a point. The 1-disc is a line. The 2-disc is a circle and its interior. The 3-disc is a sphere and its interior (a ball).

3.66
n-simplex
<geometry, topology> convex hull of n+1 points in general position in a space of dimension at least n, or a topologically isomorphic image of such a geometry

Note 1 to entry: An n-simplex is n-dimensional, and is topological isomorphic to an n-disc. A 0-simplex is a point. A 1-simplex is a curve (usually a line) with two 0-simplexes (points) on its boundary; a 2-simplex is a triangular surface with three 1-simplices on its boundary. In general, an n-simplex boundary will have n (n–1)-simplexes. An n-simplex and an n-disc are topologically isomorphic. The boundary of an n-simplex is isomorphic to an (n–1)-sphere. In all cases, the number prefix represents the topological dimension of the object.

3.67
n-sphere
<geometry, topology>geometry isomorphic to the set of points X in $\mathbb{E}^{n+1}$ such that $\|X\| = 1$; set of all points in $\mathbb{E}^{n+1}$ one-unit distance from the origin

Note 1 to entry: An n-sphere is isomorphic to the boundary of an n+1 disc.

3.68
neighborhood
<topology, metric spaces> open set of points containing a specified point in its interior

3.69
node
<topology> 0-dimensional topological primitive

Note 1 to entry: The boundary of a node is the empty set.

3.70
normal curvature vector
<differential geometry, geodesy> projection of the curvature vector of the curve perpendicular to the tangent plane to the surface at the point

Note 1 to entry: The normal curvature vector of a curve is dependent only on the curve and the surface (which is a constraint on the curve). The normal curvature vector of a curve restricted to a surface is parallel to the normal vector of the surface. If the normal curvature vector is equal to the curvature vector of the curve everywhere, then the curve is a geodesic.

3.71
normal
<differential geometry, geodesy> vector perpendicular (orthogonal) to the geometric object (curve or surface) at the point

Note 1 to entry: The normal will be parallel to the "upward" normal for surfaces in $\mathbb{E}^3$ defined in the surface interface in 6.4.17.2.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.72
normal section curve
<differential geometry, geodesy> plane curve segment containing the normal at one of its terminal points

Note 1 to entry: The usual construction begins by choosing one of the end points, the normal to the surface at that end point, and the location of the other end point and creating a plane in $\mathbb{E}^3$, which is then intersected with the datum. This curve is a normal section curve between these two points, containing the normal to the surface at the first point. On the sphere, this is the geodesic, and the two normal section curves are equal. On an ellipsoidal datum, the same is true for two points on the equator, or two points on the same meridian plane, but false in general. The reason is that in these cases the plane through the two points and the centre of the ellipsoid are planes of symmetry for the ellipsoid dividing the ellipsoid into symmetric "mirror" image halves. In general, the geodesic between the two points lies between the two normal section curves.

3.73
open set
<metric, topology, geometry> containing a metric or topologically open neighborhood of each of its points

Note 1 to entry: In a metric space, a set X us open if each point "x" in the set is contained in some small ball which is a subset of X: $[X \text{ is open}] \Leftrightarrow [x \in X] \Rightarrow [\exists \varepsilon > 0 \ni \text{distance}(x, y) < \varepsilon] \Rightarrow [y \in X]$. A topology is a set of subsets of a space which are considered open.

3.74
partition of unity
<mathematics> set of real-valued functions all over the same domain whose arithmetic sum at every domain value is 1

3.75
path connected
property of a geometric object implying that any two points on the object can be placed on a curve that remains totally within the object

Note 1 to entry: There are more general geometries that are connected but not path connected, but they are not representable as collections of “digital” geometries defined in this document. No geometric objects constructible by the methods in this document on a standard digital computer can be connected but not path connected. See the notes at the term "connected".

3.76
plane curve segment
plane curve
<geometry> curve in $\mathbb{E}^3$ that is contained in a plane

Note 1 to entry: In a 2-dimensional coordinate system, the test for plane curve to be valid is done in the 3-dimensional space of the geometric reference surface or coordinate reference system datum.

3.77
planar topological complex
<topology> topological complex that has a geometric realization that can be embedded in Euclidean 2 space ($\mathbb{E}^2$)

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.78
Pythagorean metric
<Euclidean geometry> distance measure on a $\mathbb{E}^n$ coordinate space using a root-mean sum of the differences between the individual coordinate offsets

Note 1 to entry: $P=(p_i), Q=(q_i), distance(P,Q)=\sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}$. The proofs of the Pythagorean metrics all depend on the local "flatness" of the space. Cartesian coordinate space which have Pythagorean metrics are called Euclidean spaces ($\mathbb{E}^n$). In the realm of coordinate reference systems, only "Engineering Coordinate Systems" are Euclidean. Any CRS using a curved Datum are by definition non-Euclidean, and cannot "truthfully" use Pythagorean metrics except for approximation. These approximations are valid for topological statements, but not for real world measures without adjustments.

3.79
rhumb line
loxodrome
<geometry, navigation> curve which crosses meridians of longitude at a constant bearing

3.80
range
co-domain
<mathematics> acceptable target values of a function

3.81
row-major form
<mathematics, computer science> storage mechanism for multidimensional array in linear memory, organized such that each row is stored in consecutive locations and such that the complete rows are the stored one after the other and continuing on is a similar fashion of each additional index

Note 1 to entry: If the indexes are (i, j) with the number of rows "r" and columns "c", then the mapping between the multidimensional locations to the linear storage locations is given by:

$$\begin{gathered}[i,j\in\mathbb{Z}\ni 1\le i\le r;1\le j\le c]\Rightarrow[(i,j)\to(i-1)c+j] \\ [i,j,k\in\mathbb{Z}\ni 1\le i\le r;1\le j\le c;1\le k\le f]\Rightarrow[(i,j,k)\to((i-1)c+j-1)f+k]\end{gathered} \tag{5}$$

Note 2 to entry: The matrix $\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$ in row major form is stored as [1 2 3 4 5 6 7 8 9]. For higher dimensions, the same pattern is applied recursively.

3.82
second geodetic problem
inverse geodetic problem
<differential geometry> problem that given two points, determines the initial direction and length of a geodesic that connects them

Note 1 to entry: See the first geodetic problem, which reverses this operation.

3.83
segment
<topology, geometry> minimal subpart of a geometry, usually as part of a composite

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.84
simple
<topology, geometry> homogeneous (all points have isomorphic neighborhoods) and with a simple boundary

Note 1 to entry: The interior is everywhere locally isomorphic to an open disc in a Euclidean coordinate space of the appropriate dimension $D^n = \{P \| \|P\| < 1,0\}$. The boundary is a dimension one smaller. This essentially means that the object does not intersect nor touch itself. Generally used for a curve that does not cross not touch itself with the possible exception of boundary points. Simple closed curves are isomorphic to a circle.

3.85
spatial dimension, adj
<topology, geometry> number of independent decisions in a coordinate system required to locate a position

Note 1 to entry: This definition is logically equivalent to the topological dimension of spatial projection of the CRS. It describes the space as a target for geometry. Grammatically, the term can be a noun but used to describe space, as "the spatial dimension of CRS84 is 2".

3.86
spatial dimension, noun
<topology, geometry> any of the independent decisions made in a coordinate system to locate a position

Note 1 to entry: "A Euclidean space with a spatial dimension of 3, $\mathbb{E}^3$, usually uses axis names 'x', 'y', and 'z', Its first spatial dimension is 'x', its second is 'y' and its third is 'z'." In the context of the space, the adjective use describes the space but the singular noun can use a name for an axis separately.

3.87
spatial object
<topology, geometry> object used for representing a spatial characteristic of a feature

3.88
spatial operator
<topology, geometry> function or procedure that has at least one spatial parameter in its domain or range

Note 1 to entry: Any UML operation on a spatial object would be classified as a spatial operator as are the query operators in Clause 8 of this document.

3.89
start node
<topology, graph theory> node in the boundary of an edge that corresponds to the start point of that edge as a curve

3.90
start point
<geometry> first point of a curve

3.91
subcomplex
<topology, geometry> complex all of whose elements are also in a larger complex

Note 1 to entry: Since the definitions of geometric complex and topological complex require only that they be closed under boundary operations, the set of any primitives of a particular dimension and below is always a subcomplex of the original, larger complex. Thus, any full planar topological complex contains an edge-node graph as a subcomplex.

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.92
tangent
<differential geometry, calculus> direction indicating the instantaneous direction of a curve at a point

Note 1 to entry: The tangent is usually calculated by differentiation of a functional representation of a curve but it may be approximated by a secant (double intersection) line from the point passing through another nearby point on the curve. The closer the second point is to the first, the better the approximation of the tangent's direction.

3.93
tangential curvature vector
geodesic curvature vector
<differential geometry, geodesy> projection of the curvature vector of a curve onto the tangent plane to the surface at the point

3.94
tangent space
tangent plane
collection of tangent vectors for curves passing through the point

3.95
tangent vector
first derivative of a curve parameterized by arc length

Note 1 to entry: If $c(s)$ = [($x(s)$, $y(s)$, $z(s)$]is a curve in a 3D Cartesian space ($\mathbb{E}^3$), and s is arc length along c, then the tangent is $\vec{\tau}(s)=\dot{c}(s)=(\dot{x}(s),\dot{y}(s),\dot{z}(s))$, i.e. the derivative of the coordinate values of $\boldsymbol{c}$ with respect to $\boldsymbol{s}$. The curvature vector is $\kappa(s)=\ddot{c}(s)=(\ddot{x}(s),\ddot{y}(s),\ddot{z}(s))$.

3.96
topological boundary
<geometry, topology> boundary represented by a set of oriented topological primitives of smaller topological dimension that limits the extent of a topological object, or by the geometric realization of such a set

Note 1 to entry: The boundary of a topological complex is the boundary of the geometric realization of the topological complex.

3.97
topological complex
<geometry, topology> collection of topological primitives that is closed under the boundary operations

Note 1 to entry: Closed under the boundary operations means that if a topological primitive is in the topological complex, then its boundary objects are also in the topological complex.

3.98
topological expression
<geometry, topology> collection of oriented topological primitives which is operated upon like a multivariate polynomial

Note 1 to entry: Topological expressions are used for many calculations in computational topology.

3.99
topological object
<geometry, topology> spatial object representing spatial characteristics that are invariant under continuous transformations

Note 1 to entry: A topological object is a topological primitive, or a topological complex.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

3.100
topological primitive
<geometry, topology> topological object that represents a single, homogeneous, non-decomposable element

Note 1 to entry: A topological primitive corresponds to the interior of a geometric primitive of the same dimension in a geometric realization.

3.101
topological solid
<geometry, topology> 3-dimensional topological primitive

Note 1 to entry: The boundary of a topological solid consists of a set of directed faces.

3.102
type coercion
<programming> conversion of one type of value to a value of a different type with similar content

Note 1 to entry: Point and DirectPosition are informationally identical (related to each other in a one to one fashion) in this context. Given a DirectPosition, a Point can be constructed. Given a Point, a DirectPosition can be derived from of its coordinates. If coercion is supported, a Point may be used where a DirectPosition is requested, and vice versa. Most programming languages use coercion, but others use “cast” operators (a type of constructor) requiring the programmer to initiate the coercion. “Strong substantiality” is related in that a subtype instance can always be coerced to any of its supertypes.

4 Symbols, notation and abbreviated terms

4.1 Presentation and notation

4.1.1 Unified Modeling Language (UML)

In this document, conceptual schemas are presented in the Unified Modeling Language (UML). ISO 19103 conceptual schema language presents the specific profile of UML used here. In general, UML 2 is used and most of the classifiers from ISO 19107 which use the stereotype «type» are stereotyped «interface» in this document. This document does not mandate any specific implementation system or language. Implementations do not have to use "class specification" and may use alternate implementation mechanism if they can create digital "objects" instances that meet the logical intent of the requirements stated in this document.

4.1.2 Naming conventions

Where possible, when a classifier represents the common behavior of a set of defined things from the terms defined in Clause 3, the UML classifier will generally use the defined terms as its name. Since classifier names are capitalized and contain no space, and the defined term may contain several words, the classifier name will separate words using upper-camel-case concatenations (no spaces but each word beginning with a capital with all other letters in lowercase). Similarly, the name may be some simplified key phrase. This "UpperCamelCase" rule is generally followed but may be violated if clarity or consistency with other standards is improved by minor violations. For example;

— Point values may be represented by the datatype DirectPosition or the interface Point.

— Instances of other geometry primitives will realize the interface Primitive in the package Geometry, and other interfaces for their specific dimension and interpolation mechanism. (Curve, Surface, Solid, PolynomialCurve, Bspline, etc.

— The abstract root interface of the classes is also called Geometry.

— Instances of curves will realize the interface Curve.

— Any class name referenced from another standard will retain its original format.

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

Other names in the UML models may similarly use key phrases in lowerCamelCase (same as UpperCamelCase, but the first word begins in a lowercase letter). For example, the boundary, dimension, endPoint and startPoint are all used as names. Operations that create instance of the named classifier will use UpperCamelCase to parallel the classifier name that it instantiates.

4.2 Organization

The clauses in this document are organized in terms of UML packages. Inter-package dependencies define the criterion for viable application schemas; an application schema that contains an implementation of any package defined from this document will also contain implementations all its dependencies.

The most common keyword for a package in this document is «requirementsClass». Such packages will correspond to a conformance class in the sense that each conformance test in a conformance class will test one or more requirement referencing classifiers in the corresponding requirements class stated in this document. Keywords comparisons in UML are case-insensitive, but to remain consistent, all keywords and stereotypes are used in lowercase.

4.3 Abbreviated terms and symbols

The following abbreviated terms, symbols and names are important key to understanding the normative or informational content to follow. They are in general known publicly, and readers of this document will need to understand their meaning. The following are useful in understanding this document:

∅	Empty Set, the unique un-typed set containing no elements
≅	Logically equivalent
$(\ )^n$	Implies a cross product space of "n" copies of the base, n = 0, 1, 2, ...; n=o implies a zero-length tuple; the cardinality of $\mathbb{R}^n$ for n=0 is one, the empty tuple
« »	Limits of a UML stereotype or keyword, either in the text or in a diagram, the symbols are called "guillemets"
2D	2-dimensional (meaning the geometric dimension of spatial portion of the CRS)
3D	3-dimensional (meaning the geometric dimension of spatial portion of the CRS)
API	Application Programming Interface
ATS	Abstract Test Suite
C++	Programming language based on C with object-oriented extensions, "c plus plus"
CAD/CAM	Computer Aided Design, Drafting or Manufacture (usually geometry based design systems)
CRS	Coordinate Reference System, usually a reference to an instance of the classifiers in ISO 19111
CS	Coordinate System
DMS	Degrees (°), minutes (′), seconds (″) – representation of an angle such as 34° 32′ 15″
$\mathbb{E}^n$	Euclidean n-space (n dimensional Euclidean geometry space including the Pythagorean metric, usually 2 or 3 dimensional, spatial)
ECEF	Earth Centred, Earth-Fixed (an $\mathbb{E}^3$ space containing the ellipsoid of choice)
ECR	Earth Centred Rotational, alternate name for ECEF

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

JavaScript	Prototype-based scripting language with objects, dynamic typing and first-class functions
JSON	JavaScript Object Notation
LISP	Programming language based on LISt Processing. The standard is Common LISP.
MBR	Minimum Bounding Region (sometimes Rectangle)
OCL	Object Constraint Language (as defined by the UML standards)
Per	Permission
$\mathbb{R}$	The real numbers
Rec	Recommendation
REQ	Requirement
$\mathbb{R}^{n}$	Tuples of "n" Real numbers, possibly non-Euclidean, non-Cartesian
$\mathbb{R}^{n,1}$	$\mathbb{R}^{n}$ in homogeneous coordinates $\left[(x,y)\in\mathbb{R}^{2}\rightarrow(wx,wy,w)\in\mathbb{R}^{2,1}\ if\ w\neq 0\right]$; although w is an ordinate the dimension of $\mathbb{R}^{n,1}$ is "n" since $\forall w:(x,y,1)=(wx,wy,w)$
RSID	Reference System Identifier, or Reference System Identity
SQL 3	Common name for SQL 99 during its development
SQL 99	SQL language adopted in 1999, which includes object-oriented datatypes
SQL/MM	SQL Multi Media extensions for SQL, see ISO/IEC 13249 3
TEN	Tetrahedron Irregular Network (no specific constructor is implied)
TIN	Triangulated Irregular Network (no specific constructor is implied)
UML	Unified Modeling Language
URI	Uniform Resource Identifier (IETF RFC 3986)
WKB	Well-Known Binary (data represented as standardized structured binary)
WKT	Well-Known Text (data represented as standardized structured text)
$\mathbb{Z}$	The integers
$\mathbb{Z}+$	The positive integers, often used as an index of ordinal numbers, i.e. 1st, 2nd, 3rd, …
$\mathbb{Z}^{n}$	Tuples of "n" integers

5 Conformance

5.1 Requirements class conformance targets

5.1.1 Conformance targets

This document presents conceptual schemas for describing the spatial characteristics of geographic features. This schema defines conceptual interfaces that may be realized in application schemas,

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

profiles and implementation specifications. The document concerns ONLY externally visible interfaces and places no restriction on the underlying implementations other than what is needed to satisfy the interface specifications. Application schema structural examples are given in Annex B.

Standardization targets (things which are testable against the Conformance Classes) determine the depth to which tests are made. This document determines tests for three classifications of target types. These are:

— Data structures for data interchange, messaging or transfers using encoding similar to those defined in ISO 19118[3]. These formats may be text or binary based. Text formats include any use or well-known text formats, or textual encodings such as XML, or JSON. Binary transfer can be associated to the in-core storage formats of any sufficiently robust programming language;

— Interfaces to software libraries and services; or

— Interfaces to databases using query languages such as SQL[12] or GeoSPARQL[40].

Each interface will have a default constructor with an associated datatype that describes one common creation operation (constructor) based on a minimal and logically complete representation of that interface's datatype implementation. Interchange and encoding will be based on these default constructors and datatypes.

Compliance to a conformance class by a "datatype" structure is a logical conformance, not a specific structure. If the datatypes implemented correspond in its informational content to the default constructors, they are deemed compliant. The following requirement defines the test more concisely.

REQ. 1 **Conformant data structures shall be logically equivalent to the default constructor for those interfaces supported by a requirements class.**

Rec. 1 **Equivalence should be demonstrated by a one to one bidirectional association between the values of the conformant datatype values and the possible values of the classifiers in the corresponding requirements class in this document.**

Two structures or statements are logically equivalent (≅) if each can be derived or its truth determined (⇒) from the other. For two logical statements or constructs A and B:

$$\left[A \cong B\right] \Leftrightarrow \left[\left[A \Rightarrow B\right] \wedge \left[B \Rightarrow A\right]\right] \tag{6}$$

The mechanism defined in the above recommendation essentially makes all implementations compliant to ISO 19107 also compliant as a data application to this document. For example, all arcs constructors in ISO 19107 can be used to calculate the parameters of the default arc data constructor (see 7.9.3) which is the arc centres and a set of arc endpoints.

Few applications will require the full range of capabilities described by this conceptual schema. This document, therefore, defines a set of conformance classes that will support applications whose requirements range from the minimum necessary to define data structures to full object implementation. The conformance classes are arranged as a core set of functionality with an orderly progression of extensions, so that a partial implementation of this document has a logical path to the functionality of the extensions.

Implementations that define full object functionality will allow the implementation of all operations and support logical access to the conceptual attributes and associations defined by the classifiers of the chosen conformance classes.

These implementations will normally be programming language dependent, but there are no specific language restrictions. UML is generally used for object-oriented languages with class systems, but as long as instances can be created that support the functions required, the use of a class system is an implementation decision. For example, C++ or JAVA may directly implement the UML model in this document, but JavaScript (which does not have a class-based system) is still capable of implementing object instances that can support the class contracts in the conformance classes in this document.

© Standards Australia Limited/Standards New Zealand 2020

AVA Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

Implementations that choose to depend on external "free functions" for some or all operations, or forgo them altogether, will support datatypes sufficient to record the state of each of the chosen classifier as defined by its member variables and operations.

The support of sufficient data implies that all functions defined here can be implemented based sole on these data descriptions. Common encodings may include XML, Well Known Text, marshalling languages such as JSON (JavaScript Object Notation) and any other format/language needed by an application environment.

Common names for "metaphorically identical" but technically different entities are necessary to link knowledge common to different domains or applications. The UML model in this document defines abstract interfaces and data structures, application schemas define conceptual classes, various software systems define implementation classes or data structures, and the encoding standard (e.g. ISO 19118) defines entity tags. These may reference the same information content. There is no difficulty in allowing the use of the same name to represent the same information content even though at a deeper level there are significant technical differences in the digital entities being implemented. This "allows" named types defined in the UML model to be used directly under the same or similar names in application schemas (see examples in Annex B). If distinctions need to be made, package name or namespace should be supplied. For example, this document defines a Geometry interface logically equivalent to the Geometry defined in ISO 19107, and gml: AbstractGeometry defined in ISO 19136 Geography Mark up Language (GML).

Because ISO 19107 is a logical subset of this document (see Annex D), all implementations conformant to ISO 19107 that have corrected the pertinent errors in that document are conformant to some conformance class of this document. Each Requirements Class described in this document will have two associated conformance classes,

a) based on data, operations, and behaviour or

b) based on data alone.

The purpose of "a type" conformance is to identify the minimal functionality for the active interoperation of geometric data applications. The purpose of "b type" conformance is to enable universal communication of spatial information between applications consistent with this document and its predecessor ISO 19107 and thereby defining a consistent and universal information community for geometry in spatial applications. For example, all versions of GML would be in "b type" conformance.

REQ. 2 **An “a-type” or “behavioral” conformant implementation of any requirements class in this document shall be able to satisfy all requirements in that Requirements Class.**

REQ. 3 **A “b-type” or “data” conformant implementation of any requirements class in this document shall be able to communicate spatial information (coordinate, geometry or topology as appropriate) based on the default constructors defined in that requirements class or logically informational-equivalent data structures whose transformation to the default constructor data types by a well-known and well-defined logical mechanism.**

There are many conformant options for schemas that define types for geometric and topological objects; they are differentiated based on differences in the following criteria:

— Geometry metrics (geodesy);

— Topological dimensionality;

— Interpolation schemes;

— Structural complexity;

— Functional completeness.

Documents
Document Sharing Hub
© Standards Australia Limited/Standards New Zealand 2020
Full document details: AS NZS ISO 19107-2020

5.1.2 Geometry metrics (geodesy)

In defining the geometry metrics, the application schema will be required to specify which underlying surfaces will be supported in calculating geometric measures. In general, the underlying surfaces (called "GeometricReferenceSurface") are:

— The 2D plane or map geometry uses standard Cartesian geometry and Pythagorean "flat" metrics, and has inherent inaccuracies associated with the curvature of the Earth dependent on the size of the area covered and the characteristics of the chosen projections (coordinate reference systems or CRS).

— A 2D sphere uses classical spherical surfaces (a perfectly round Earth of constant curvature). These systems use great circles as geodesics and do not adjust for the eccentricity of the Earth's surface. Depending on the CRS, slightly different spheres may be chosen for a "best fit" for a limited locale. While theoretically more accurate than planar, as the area of coverage increases accuracy of measurements do degrade but at a slower rate.

— A 2D ellipsoid uses a surface generated by the rotation of an ellipse about its shorter axis (polar) creating a circular equator and other lines of constant latitude, with orthogonal, elliptical lines of constant longitude. These systems adjust for the eccentricity of the ellipsoid, but do not factor in local gravitational anomalies. Again, varieties of "best fit" ellipsoids are used.

— A continuous 2D surface of an equipotential surface of gravity (such as the geoid or quasi geoid) or a tidal Reference Surface (such as mean sea level or lowest astronomic tide); Potential (or height) differences are referenced to an ellipsoid.

NOTE Although generally smooth and spatially continuous in two dimensions, such a surface is irregular and there is insufficient information to compute across it. It is usually represented through an analytical approximation using spherical harmonics where the spatial context is referenced to an ellipsoid, such as the Earth Gravity Model 2008 (EGM 2008) referenced to the WGS 84 ellipsoid, or by a digital elevation model referenced to a 2D plane.

Three dimensional geometry models will also be supported. There are essentially two approaches:

— Geocentric Cartesian 3D, using a 'flat' 3D Euclidean space: These include geocentric (origin at the centre of the Earth) and projective (origin on the surface of the Earth).

— Using one of the surfaces from the four classes above, and adding a measure of the distance in 3D either above or below the "GeometricReferenceSurface": The third dimension is measured parallel to a near normal or normal line (orthogonal to the surface, in the general direction of gravitational up).

5.1.3 Topological dimensionality

In defining the topological dimensionality of object types to be implemented, the application schema will be required to specify which of the interpolation types for curves or surfaces they wish to implement.

There are four levels for simple spatial geometry:

— 0-dimensional objects (points and locations in a well-defined coordinate system);

— 0-dimensional and 1-dimensional objects (adding curves of various types);

— 0-, 1-, and 2-dimensional objects (adding surfaces or areas of various types);

— 0-, 1-, 2- and 3-dimensional objects (adding solids).

5.1.4 Interpolation schemes

Curve implementations, for those application schemas including topological 1-dimensional objects, will always include "linear" or "geodesic" interpolation. Application schema including 1 dimensional objects should always include a mechanism to approximate any curve as a line or geodesic string to allow for transfer of data into simpler schemas when needed.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

Surface implementations, for those application schemas including 2 dimensional objects, will always include a "planar" or "reference surface" interpolation technique. The most common such interpolations are boundary representations, where boundary curves lie on a single surface. These include triangular irregular networks (TIN) and polyhedral surfaces. Obviously, in 2D coordinate reference systems these polygons (with any curves as boundary) are the only surface types. "Non-flat" surfaces require 3D coordinate reference systems.

Rec. 2 **Application schema should always include a mechanism to approximate any surface as collections of planar surface segments to allow for transfer of data into simpler schemas when needed.**

Additional curve and surface interpolation mechanisms are optional, but if implemented, they will follow the definition included in this document.

There are no restrictions on extensions of this document. The most common extension themes may include, but are not limited to the following:

— Mechanisms to improve geometric measures such a Gaussian and Riemannian metric techniques and algorithms for the improvement of accuracy or efficiency in measures for length, area and volume for geometric objects;

— Interpolation techniques that extend the point, curve, surface and solid techniques established in this document, e.g. triangulated networks, tetrahedral networks, specific spline interpolation methods. These may be targeted to specific applications;

— Indexing or tiling algorithms that improve performance or efficiencies for geometries in important geographic coordinate reference systems or datums;

— Encoding techniques for geographic geometry, such as WKT, JSON or XML;

— Algorithmic improvements to the maintenance and editing of geometry and topology structures both cellular and simplicial topology;

— Dealing with spatial temporal (4D) geometries and topologies.

5.1.5 Structural complexity

Four levels of structural complexity are identified:

— Geometric primitives;

— Geometric complexes;

— Topological primitives and complexes;

— Topological primitives and complexes with geometric realization.

NOTE If single definitions of each component of geometry are required, then geometric complexes are introduced into the schema. Primitives within the same geometric complex share only boundaries. If the schema requires explicit topological information, then the geometric complex is expanded to include the structure of a topological complex. The most obvious implementation would have every geometry object also implement the topological object interface. The types of object included in a complex are controlled by the dimension of that complex. What is commonly called "chain node" or "link node" topology is a 1-dimensional topological complex. What is commonly called "full topology" in a cartographic 2D environment is a 2-dimensional topological complex realized by geometric objects in a 2D spatial coordinate system. A full topology in a 3D environment would be a 3-dimension topological complex realized by geometric objects in a 3D spatial coordinate system.

The third criterion is the level of functional complexity or completeness that determines the member elements (attributes, association roles and operations) of those types that are implemented.

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

5.1.6 Functional complexity

There are three levels of functional completeness as follows:

— Data types only;

— Simple data access operations;

— Complete operations.

Subclause 10.8, Requirements Class Derived Topological Relations, of this document defines Boolean operators that may be used to investigate topological and metric relations between geometric objects.

The normative content of this document is divided into the "requirements class" packages listed in the UML model and normative Clauses. Each of these sets of requirements will correspond to a conformance class in the Abstract Test suite. Each requirements class Dependency will be realized as a conformance class extension.

5.2 Conformance classes

To conform to this document, an implementation will satisfy the requirements of the abstract test suite (ATS) in Annex A for a specified conformance class. In general, these conformance classes can be given by "description" parameters. These parameters include the following identifiers and their list of possible values:

Geometric Reference Surface (coordinate surface representing 0-height):

— planes;

— spheres and planes;

— ellipsoids (biaxial rotated on the polar axis), spheres and planes; or

— geosurfaces and geoids, ellipsoids, spheres and planes.

Mechanism for height:

— none (2D);

— distance from the GeometricReferenceSurface (3D or compound 2+1D).

Dimension of non-empty geometry supported (empty is "–1" dimensional):

— 0-dimension, only points or direct positions;

— 1-dimension, curves and points;

— 2-dimension, surfaces, curves and points; or

— 3-dimension, solids, surfaces, curves and points.

Support of topological structures up to dimension[2)]

— no support;

— 1-dimensional network graphs, e.g. links and nodes;

— 2-dimensional graphs, e.g. networks spanned by surfaces; or

2) Higher dimension topological complexes are acceptable, but probably restricted to cellular or simplicial topological complexes, where each element is an isomorphic copy of a disk of appropriate dimensions $\left(D_n = \{\vec{p} \in \mathbb{R}^n \mid \|p\| \leq 1\}\right)$. This will allow academic theorem from algebraic topology (specifically CW or simplicial complexes) to be applied appropriately.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

— 3-dimensional, e.g. as above with surfaces spanned by solids;

— Higher dimension based on application domains, see ISO 19111, usually parametric extension of a base spatial representation of at most 3D.

Support for interpolations:

— linear including small circle along a parallel of latitude or great circle along a meridian;

— planar;

— bilinear and trilinear;

— polynomials, such as bicubic;

— splines which use the vector algebra in the domain $\left(\mathbb{R}^{n}\right)$ of the coordinate reference system used in the direct positions;

— rhumb line curves of constant navigational bearing, crossing all meridians at the same angle;

— and others.

Euclidean geometry in local spaces (using the tangent plane) which is then mapped to the surface supported using the mapping defined by geodesics (see exponential map in the definitions below (includes geodesics, geodesic circles and other conic sections including spirals).

This means that there will be one core conformance class containing general abstract classes but implementations for point only. Each extension adds new implementations classes, clustered into packages as needed to create a logical progression of valuable extensions.

Each implementation will have a set of "codelists" that specify support for geometry classes, coordinate systems and underlying geometric surfaces (see 6.2.1).

5.3 Requirements classes

Each Conformance Class corresponds to a single Requirements Class listed by clause number in the Annex A. Each such pair or conformance plus requirements corresponds to a package with stereotype «requirementsClass» in the UML model. Dependencies between these packages, see Figure 1, correspond to dependencies between the classes. For example, the core requirements class is "Coordinates" in the UML. All other requirements classes are dependent on this core. The second layer is rooted at "Geometry" containing primitive such as Point, Curve, Surface, Solid or Topology. Directly below are the extensions for each of the variations of the primitive types: curve, surface, solids and topology. "Point" does not need a separate requirements class since there is only one Point interface that is in the core "Geometry" package. For Curves, Surfaces and Solids there are more conformance extensions based on interpolation type.

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

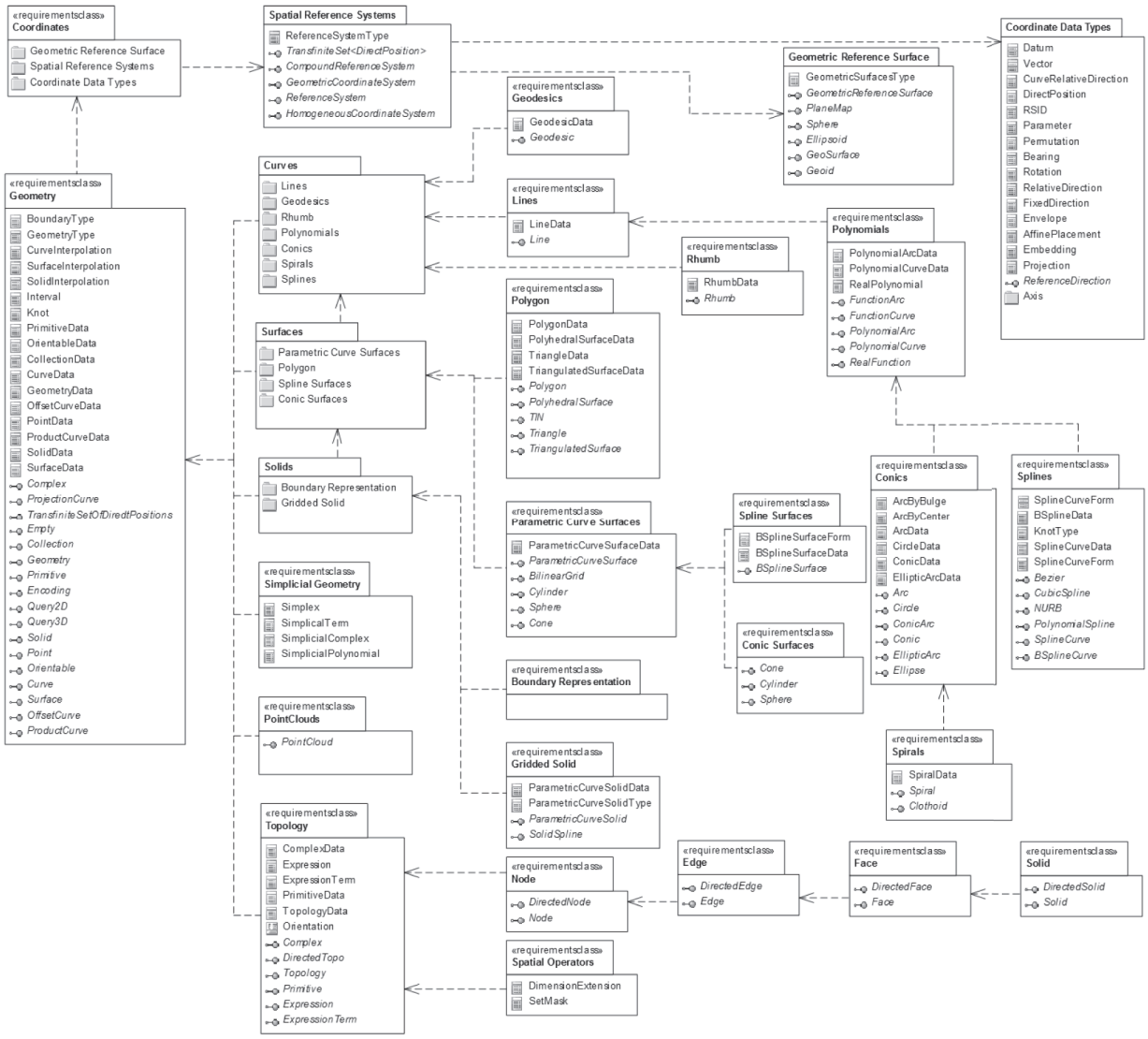

Figure 1 — Geometry packages and requirements classes

6 Coordinates and core geometry

6.1 Semantics

Coordinate geometry requires a coordinate space to exist. This space is usually not planar, so the space is generally not Euclidean. The first requirements class describes the geometric space and the following requirements classes describe the geometry that goes into them (Figure 1). ISO 19111 describes how an ordered list of coordinate values are used to represent a point on a geometric object called a datum. Euclidean spaces use Cartesian coordinates, $\mathbb{E}^n$, and use a Pythagorean metric. Other non-Euclidean ("curved") spaces may use $\mathbb{R}^n$ as coordinates but generally do not have Pythagorean metrics.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub
Full document details: AS NZS ISO 19107-2020

6.2 Requirements Class Coordinate

6.2.1 Codelists to specify capabilities

Any implementation will need to specify how deeply it has implemented the object types and other requirements. To do this a set of configurations codelists will be available as explained in 5.2[3)].

REQ. 4 **Applications shall implement the full functionality that is specified by their claims for support of coordinate systems, coordinate reference systems, geometry types and any other parameters of conformance listed in a local codelists. The codelists that shall be available, if applicable, include GeometryType, BoundaryType, CurveInterpolation, SurfaceInterpolation, SolidInterpolation, GeometricSurfaceType, ReferenceSystemType and Datum**

NOTE Each codelist will be specified in its own separate requirement. Any such requirement is part of the testing described by REQ. 4.

6.2.2 Coordinate systems for Geometry — Semantics

6.2.2.1 Coordinates, points and locations

The n dimensional coordinate space $\mathbb{R}^n$ consists of all n long arrays of numbers (or any type that can be interpreted as numbers). Each array of numbers represents a "point" in the "$\mathbb{R}^n$" space. In some processes or situations, this may be restricted to a subset of such points, called the 'extent of validity', and usually based on a set of constraints on values of the various offsets within the array. Each point is associated with a location, but a single location may be the target of multiple coordinate space points.

In a small region near a point in $\mathbb{R}^n$, if there is an isomorphism from this region to a geometry then the dimension of that geometry is "n" in that region. In barycentric coordinates, the mapping is restricted so that "the sum or the ordinates is 1" it restricts the coordinates to a hyperplane in $\mathbb{R}^n$ and therefore the dimension is "n–1". In homogeneous coordinates, variations only in the coordinate "w" will create a line whose coordinate sets map to the same point. Locations are given by direct positions; see 6.2.9, but the dimension of the coordinate space depends on how those coordinates are used.

EXAMPLE A 1-dimensional coordinate space may be mapped to a circle by angular displacement (in radians) in a chosen direction from a chosen origin on the circle. In this case, "x" and "x + 2π" are distinct points in the coordinate space, but represent the same angular location on the circle. Any interval of x shorter than 2π is one-to-one and invertible, so the geometric dimension is preserved. In this example,

$$\text{location (DirectPosition(x))} = \text{location (DirectPosition (x + 2}\pi\text{))}$$

A coordinate space and a function that maps coordinate tuples to locations define a coordinate reference system (see ISO 19111) that is associated with an implementation of the datatype DirectPosition that uses representations from this coordinate system.

3) This implies conformance testing suites should test these functionality according to the claims of the applications documentation claim.

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

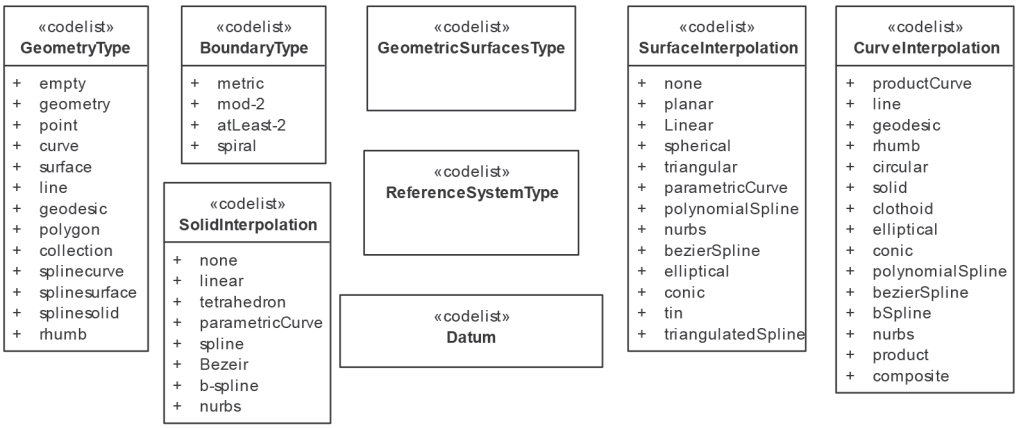

Figure 2 — Codelists used in REQ. 4

All geometric objects in this document are represented as mathematical or geometric constructions within a coordinate system that represents some set, usually infinite, of coordinated points, each point describable by a DirectPosition. **This document deals with both Euclidean ("flat" spaces) and non-Euclidean geometries ("curved" spaces).** A simple example of such geometry constructions would be a (coordinate) line defined by two coordinate points, and a linear interpolation in $\mathbb{R}^n$, here represented as vector $\overrightarrow{P_1}$ and $\overrightarrow{P_2}$ that represents all points in. $\left\{\vec{X} = u\overrightarrow{P_1} + (1-u)\overrightarrow{P_2} \middle| 0 \le u \le 1\right\}; \vec{X}(0) = \overrightarrow{P_1}; \vec{X}(1) = \overrightarrow{P_2}$.

NOTE This "line" in the coordinate system does not necessarily represent a line in the real, physical world, i.e. a "line of sight" projected back down onto the surface of the Earth, which is a segment of a geodesic curve path of shortest length between any two of its points which are the "great circles" on a sphere.

EXAMPLE A Mercator projection "line" is a line of constant bearing (rhumb line or loxodrome), and is either the arc of a circle (for east-west lines) or a segment of polar spiral. The only rhumb lines in a Mercator map that are geodesics on a sphere or ellipsoid are the equator or a line of constant longitude ("meridians") running north-south.

REQ. 5 **An implementation of the Package Coordinate shall have all instances and properties specified for this package, its contents, and its dependencies, contained in the UML model for this package in this document.**

REQ 5, put simply, requires that this package needs to be implemented completely including all of its dependent packages, in particular those that may be associated to ISO 19111.

6.2.2.2 Coordinate systems

Many of these spaces will be coordinate reference systems as defined by ISO 19111, but not all. Nonetheless, all of these systems will satisfy some basic properties.

REQ. 6 **Each coordinate offset in the tuples of a direct position shall contain instances of a class that can be represented as and therefore coerced into a single real number.**

REQ. 7 **Spatial coordinates shall be consistent with a coordinate reference system defined by the mechanisms in ISO 19111.**

REQ. 8 **Temporal coordinates shall be consistent with a reference system consistent with ISO 19108.**

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

REQ. 9 Parametric coordinate shall be consistent with the extension mechanism defined in ISO 19111.

This does not mean that the coordinate offset will be a real number, but that it can be converted to one. For example, 4° 30' (four degrees, thirty minutes) is equivalent to 4,5° and could therefore be used as a coordinate offset. In the discussions in this document, we will assume that all coordinates have been represented by real numbers, but this does not restrict representations such as DMS (degree, minutes and seconds) as they can always be cast as a real number.

A coordinate reference system that speaks of longitude will often restrict them to the equivalent of the interval [–180°, 180°]. This causes some problems for geometries that cross the 180° meridian. The easiest work around for this sort of problem is to eliminate the discontinuity by allowing –179° and 181° both to be valid and to represent the same longitude as far as position on the earth is concerned.

REQ. 10 For a coordinate offset that can be extended using the period of a function, the additions or subtractions of that period shall not affect the validity and position of the DirectPosition.

The coordinate offsets are not assumed to represent the same or even compatible units of measure. For example, in a 3D latitude longitude system, the first two offsets are angles such as in decimal degrees, and the third (usually height or, more properly, "ellipsoidal height") will be a linear unit of measure such as metres. With this and the curvature of the ellipsoid, the standard Euclidean distance formulae using the Pythagorean Theorem will not represent a distance, but will define the same set of continuous functions or topology. In other words, the underlying space is a "manifold" and is locally a continuous image of the Cartesian coordinate space of the same dimension. The continuous function will not normally be an isometry (a "preserver of distance", see the definition in Clause 3).

Any function that depends on this sort of measure, such as distance, curve length or surface area, can have two interpretations: one using the local Euclidean metrics, which are essentially unit less, and another that has been converted to an approximation of the real-world measure. This is an application option, but should be consistent with the conceptual model of the application. For example, the Earth is not a sphere, but spherical trigonometry makes spheres easier to handle than ellipsoids, and if the sphere uses a local mean radius of the Earth, the differences in length measures may exhibit only small inaccuracy, which may be sufficient for many applications.

REQ. 11 Metric (measurement) operations in geometry-related interfaces shall use the best approximation of real-world values practicable. A conformant application shall supply error budget information to allow the user to determine the fitness for use of the values returned.

Almost all modern coordinate systems used in geography can be mapped back to an Earth centred; Earth fixed (ECEF) "geocentric" Cartesian coordinate system which is essentially a Euclidean space where the Pythagorean metric is valid. Mapping back to this coordinate system will always allow a final option for calculations using 3D calculus techniques. There are methods for ellipsoids (see [48] and [47]) which can calculate directly in elliptical or spherical coordinates (latitude, longitude). Less extreme, less accurate but equally valid is to map geometry to a local engineering system (map distances) where the same "Euclidean geometry" based calculations (either corrected or not) are feasible and have controllable error budgets especially in small areas.

REQ. 12 All locations in a list or array shall use the same coordinate system and shall reference reality in a manner representable by continuous functions from the coordinate tuples (direct positions) to reality in such a manner that "nearby" coordinates in the direct positions map to "nearby" positions in reality.

Documents
Document Sharing Hub
© Standards Australia Limited/Standards New Zealand 2020
Full document details: AS NZS ISO 19107-2020

NOTE The formal definition using topological terms of a continuous mapping:

Let X, Y be topological spaces and $f: X \to Y$ is a mapping from X to Y.

$$[f \text{ is continuous}] \Leftrightarrow \left[\forall B \subseteq Y, [B \text{ is open}] \Rightarrow \left[f^{-1}(B) \text{ is open}\right]\right]$$

This document does not assume that these functions maintain coordinate dimension. For example, direct position may be given in "homogeneous coordinates" which equates, for every w≠0, the coordinate (*wx, wy, wz, w*) in four dimensional space $\mathbb{R}^4$ with the point (*x, y, z*) in a three dimensional space $\mathbb{R}^3$. The topology of both systems is three dimensional, since the value of "*w*" does not affect the spatial position.

Some offsets in the coordinate array of a direct position may have no spatial interpretation, such as a temporal parameter, or one that corresponds to some other measurement or quantity associated with the position being described, such as a measurement in a linear reference system, or a physical property of space, such as temperature or pressure. Such offsets will not usually affect distance measures, unless those measures are interpreted by the application in some manner.

All direct positions will have a projection into a purely spatial coordinate as would be associated with a coordinate reference system (SC_CRS) from ISO 19111. This projection is not assumed one to one, even if the coordinate system is purely spatial. It is assumed locally one to one, and bicontinuous. This means that the coordinate system will locally be reversible, but that it may wrap around singularities in the definition of the SC_CRS, such as occurs at 180° longitude line in a latitude longitude system or at the poles in such a system.

6.2.2.3 Metrics and distance measures

From the above description, the "flat" concept of distance given by the Pythagorean Theorem is not directly applicable to DirectPositions except in limited circumstances.

The general solution is the use of Riemannian or Differential geometry, but this level of generality is not normally needed in geographic information. The key to the simplification is in ISO 19111, in that the datum surfaces that approximate the Earth are usually embedded in a 3D geocentric Euclidean space. This means that the geometries on this datum surface are also in $\mathbb{E}^3$. Most calculations on ellipsoids can use Vincenty's formulae[48][47][46], or in the ECEF $\mathbb{E}^3$ that contains the datum surface. The distance formulae for great circles on a sphere for two points is (angles in radians) is:

$$\begin{aligned} &\text{points} = \{(\phi_1, \lambda_1), (\phi_2, \lambda_2)\},\ r = \text{(local) radius of the Earth} \\ &d(p_1, p_2) = r(\arccos(\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \sin(|\lambda_1 - \lambda_2|) \end{aligned} \tag{7}$$

A recursive form of Vincenty's formula[47] is straight forward and it creates accurate ellipsoidal results. Several open source versions of the algorithms may be found on the web.

6.2.2.4 Underlying geometric space

Geometry systems inherently make assumptions about the "space" in which the geometric objects are defined. Classical Euclidean geometry assumes that the objects lie in an infinite flat plane represented, for example, by a Cartesian coordinate space with a standard Pythagorean metric. Geographic information if restricted to relatively small areas can be analysed using classical Euclidean methods or algebraic calculations in a Cartesian coordinate space that depend on the 'flat' Pythagorean metric. For larger areas, the effects of the Earth's curvature become significant even for primitive surveying systems.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

EXAMPLE Much of the United States is surveyed using the Land Ordinance of 1785, which defines Public Land Survey System (PLSS) commonly called "the Jeffersonian Grid". In this system, major boundaries are defined by "equally spaced" horizontal (east west) lines of constant latitude and vertical (north south) lines of constant longitude. Since these lines are generally the boundary lines between owned parcels, roads often follow them. Since the lines are spaced by distance (6 or 24 miles, approximately 10 or 38 kilometres), they misalign due to the curvature of the Earth, the north south roads often "jog" (meaning a brief abrupt double change in direction, "left then quick right" or vice versa) when they cross the major east west lines also being used as rights of way for other roads. The further away from the origin of the "grid" the greater the width of the "jog". Similarly, on any UTM projection, the metres easting and northing for a grid square would not be sufficient to calculate the area of that square with the distortion of the area dependent on the square's position.

In other words, Euclidean based intuition is inaccurate on a curved, and therefore non-Euclidean, surface. To make such calculations more accurate, a more accurate model of the Earth's surface is needed. The most common models involve 2D surfaces embedded in a 3D space. The 3D space is a Euclidean space, with a Pythagorean metric represented by Cartesian coordinates. The geometric/metric structure of the surfaces is inherited from the embedding 3D Cartesian space. On the surface, the distance between points is defined by curves of shortest distance lying on the surface that joins the points in question, similarly to the way lines in flat Euclidean spaces define distances in the Pythagorean metric. Thus, the distance between points on a sphere is defined by the length of the great circle on the sphere joining the two points. For ellipsoids, the geodesics are more complex, but they are defined as curves of shortest distances between points, and their length defines the metric on the underlying surface.

The surfaces in most general use are:

— Maps – a relatively local map projection onto a 2D plane where Euclidean geometry properly scaled is a close approximation to reality,

— Spheres – a 2-dimensional sphere with a fixed radius, embeddable in $\mathbb{R}^3$, allowing the use of classical spherical geometry and trigonometry,

— Ellipsoids – a bi radial surface of revolution using an ellipse of revolution whose minor axis is approximately aligned with the Earth's polar axis, or

— Calculated GeometricReferenceSurface or Geoids – an adjustment of an ellipsoid to an equipotential surface of gravity to adjust for the variations in Earth's local density and thus it gravity potential.

In ISO 19111 a sphere can be considered to be a special case of an ellipsoid and no distinction need be made between them for CRS definitions. In this document a distinction is made because geometric calculations on a sphere use simpler formulas based on the symmetries of a sphere, e.g. those based on "great circles" geodesics, whereas many geometric calculations on an ellipsoid are more complex and may require the use of more complex approximations. An approximation is not necessarily less accurate than a closed form formula, which will use polynomial approximation for trigonometric functions, but different implementations may treat these differently.

Issues of local usage can adjust a CRS to work based on a local best fit of the Reference Surface. For example, in the United States local mapping agencies often use State Plane Coordinate system specifically to preserve the accuracy of Euclidean geometry. To do this, there are 124 distinct state plane systems used to cover the country. Such a system could use the "Reference Surface — Plane" parameter for the coordinate conformance class.

6.2.3 GeometricReferenceSurface

6.2.3.1 Interface GeometricReferenceSurface

A GeometricReferenceSurface is any surface usually embedded in a Euclidean space used to describe a geometric space for determining various measures (e. g. length or area) (Figure 3).

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

The interface GeometricReferenceSurface has three attributes:

— id: URI — each GeometricReferenceSurface has a unique identity. To support Web access, the identity is a URI. In many cases this will correspond to a surface used as a datum;

— metadata: URI[1..*] — each geometric surface should have a complete set of metadata;

— datum: Datum — the actual surface (derived from metadata).

REQ. 13 A GeometricReferenceSurface implementation shall have all properties for that interface specified for it in the UML model in this document.

REQ. 14 If the GeometricReferenceSurface is not embedded in a Euclidean space, then it shall have either a Riemannian metric or some other mechanism to determine distance and area metrics within a documented level of accuracy.

REQ. 15 The use of approximations for distance and area measures shall be part of the metadata of any application

Some ellipsoid approximations based on approximating spheres are given in Formula (7). These three items, the CRS, the GeometricReferenceSurface and the association between coordinates in the CRS and position relative to that GeometricReferenceSurface are sufficient to define how geometric calculations such as length, distance and area may be done using the geometry objects defined in this document.

REQ. 16 If the CRS is supplied but the GeometricReferenceSurface is unspecified, the default surface shall be the surface of the CRS object's related Datum as defined in ISO 19111.

REQ. 17 When using a 2D CRS, all geometric measures, such as distance, bearing, length, area or volume, shall be consistent with the underlying GeometricReferenceSurface or Datum

NOTE For surfaces that are more complex (ellipsoids and geoids) numeric approximations for distances and geodesic are often used. Implementations will be required to access these approximations to determine their error budget. For ellipsoids, Vincenty's formulae[47][38] may suffice.

REQ. 18 For each 3D CRS, the horizontal 2D measures shall be consistent with the geometric reference system, and the vertical adjustment shall be consistent with an isometric embedding of the GeometricReferenceSurface in 3D Euclidean space.

REQ. 19 Each implementation system shall specify each CRS supported in a codelist.

In a 2D system, volume measures may be calculated from footprints (horizontal extent) and separate attribute information such as average height.

6.2.3.2 CodeList: GeometricSurfaceType

The codelist GeometricSurfaceType is a list of all GeometricReferenceSurface types by identify supported by the local application.

REQ. 20 The local instance of the codelist GeometricSurfaceTypes shall list all GeometricReferenceSurface or Datum types supported by the local application.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

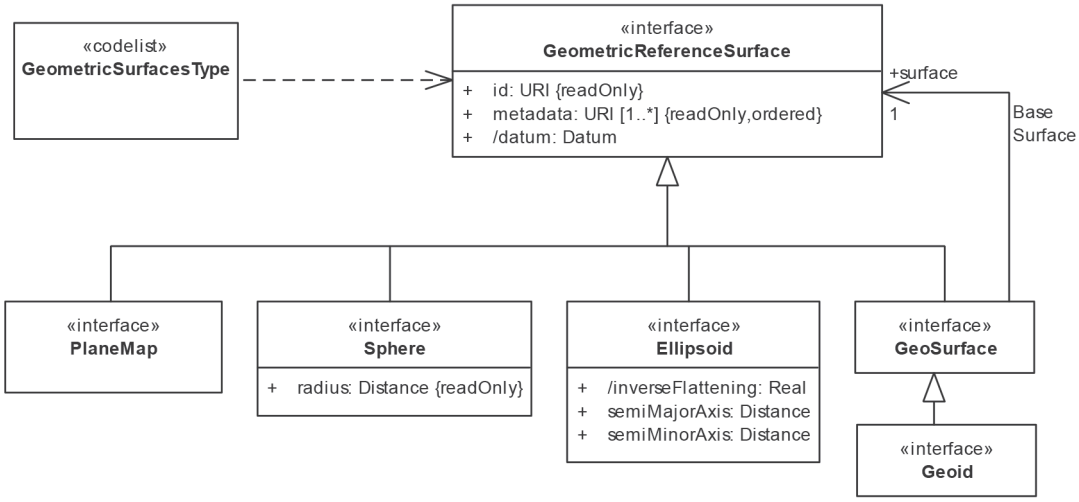

Figure 3 — GeometricReferenceSurface

6.2.3.3 Interface PlaneMap

Some geographic processing is still done using a "map metaphor" which essentially assumes that the flat image of an underlying graphic (paper or screen) differs from reality only by the scale of the map. While the versions of this assumption are varied, two cases seem to be more common than others:

1. “flat earth” assumptions, and

2. Cartographic map projections.

In the "flat earth" assumption, the plane is mapped to the Earth by assuming a single point (the "origin") of attachment (possibly given in a geographic coordinate system) and all other points defined by measured offsets and directions (vectors) from this origin. This essentially equivalent to the “exponential map” (see the definition in 3.36). This is especially useful in small areas where the curvature of the Earth makes only small variations in the measures, in general areas of less than 10 kilometres (or 6 miles). In ISO 19111, these systems are called Engineering CRSs; and may be architectural or engineering drawings using large scales.

If the map is created by using local surveying procedures, then one way to think of this may be to consider the map to be representation of the tangent plane at the origin, and the mapping to points on the map to be defined by the differential geometry "exponential map" from this tangent plane to any more accurate geometric reference system. The exponential map looks at the tangent plane in polar coordinates, being a bearing and a distance from the origin. This "polar" representation is then used to generate the geodesic on the Earth with this initial bearing, and this length and using its endPoint as the position on the Earth that is represented by the position on the map defined by this angle and this distance.

If the map plane is viewed as the x y, or $\mathbb{E}^2$, the normal to the plane is a vector representing "up". The geodesics in the plane are the straight lines, but these can only be used with any accuracy in small areas since when they are brought back to the curved Earth, their geometry changes. This essentially means that these entities use "map measures" and the CRS system in use is a projection.

Different projections may preserve different aspects of the real "curved" geometry. Some may preserve shape, angle, area or distance better than others. But, projection of a spherical or ellipsoidal model of the Earth to a plane cannot be done without some distortion. Conventional topographic maps usually preserve shape in a small area (they are often conformal maps, preserving angles, see the definition in 3.10). Mapping designed for statistical purposes may preserve area. For accurate measures of different types, different map projections may be required.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub
Full document details: AS NZS ISO 19107-2020

6.2.3.4 Interface Sphere

The spherical approximation to the Earth's surface is one of the oldest in continuous use, and "central angle" latitude and longitude (valid for spheres but not ellipsoids) is the most common mental model of terrestrial locations (Figure 3). Further, spherical trigonometry (19th century) gives closed form solutions to most calculations needed for model-based measures.

Each sphere's shape is completely defined by its single attribute:

— radius: Distance — radius (mean radius of Earth is about 6,371,009 m)

Assuming the sphere in $\mathbb{E}^3$ is at the origin $(x, y, z) = (0,0,0)$, then $x^2 + y^2 + z^2 = r^2$. The normal vectors to the surface are the extensions of the radial lines from the centre of the sphere. The geodesic on the sphere are the great circles defined by planes through the origin and creating circular curves centred there with their normal equal to the normal of the sphere.

NOTE This means in these spherical CRS systems, 2D geodesics are great circles. The use of 3D adds offsets from the sphere and distances calculable by integrals in the embedding Euclidean space.

Math The planes that pass through the sphere's centre cut the sphere in great circles. Since these circles are contained within these planes, their curvature vectors (2nd derivatives of the curves with respect to arc length) are vectors following the radial lines of these circles, and thus perpendicular to the sphere. Therefore, the great circles have zero tangential curvature vectors and are thus geodesics by the definition in Clause 4.

6.2.3.5 Interface Ellipsoid

Mathematically an ellipsoid can have 3 separate radii, but a geographic ellipsoid is defined by a surface of rotation for an ellipse, around its shorter axis (which becomes the polar axis). So, each geographic ellipsoid's shape is defined by two radii (Figure 3) for it axes:

— semiMajorAxis: Distance — largest radius, anywhere along the equator (a);

— semiMinorAxis: Distance — smallest radius, at the two poles (b).

A third alternate value, "inverse flattening", is often given, eliminating the need for the semi-minor axis.

— inverseFlattening: Real — derived value equal to $\left(\frac{a}{a-b}\right)$ which is close to 300,0.

REQ. 21 **A compliant implementation that supports Ellipsoids as a GeometricReferenceSurfaces for their coordinate reference shall implement all metrics consistent with this ellipsoid's embedding in $\mathbb{E}^3$.**

REQ. 22 **Distance on ellipsoids shall be determined by geodesics on the ellipsoid that are curves with a zero tangential curvature vector, or by some reasonable approximation (see REQ. 30).**

6.2.3.6 Interface GeoSurface

For the purposes of this document, a geosurface is any surface defined by offset mechanism from another standard GeometricReferenceSurface (its "Base Surface"), the most common example being an isosurface of the gravity potential (Figure 3).

REQ. 23 **Compliant implementations using a GeoSurface shall take into account the offset mechanism in calculating metric values.**

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

6.2.3.7 Interface Geoid

For the purposes of this document, a geoid is a geosurface defined by an isosurface of the gravity potential usually using an ellipsoid as it Base Surface (Figure 3).

6.2.4 Interface ReferenceSystem

6.2.4.1 Semantics

A reference system is usually an implementation of the datatype RS_ReferenceSystem from ISO 19111. The major difference is that the local version uses a URI as an identity, since most useful geographic information has to be in a commonly understood coordinate system meaning its and details needs to be easily available to all (Figure 4).

6.2.4.2 Attribute id

REQ. 24 Each ReferenceSystem shall have a URI for identification.

URI used for this purpose should be HTTP-URIs.

```
id: URI
```

6.2.4.3 Attribute rsid

In many applications, such as linear reference systems (ISO 19148) and moving features (ISO 19141), extra columns are added to the coordinates for "other" non-ISO 19111 purposes. The most common practice is to concatenate reference systems, represented by listing an RSID for each component system.

```
rsid: RSID [1..*]
```

REQ. 25 Each ReferenceSystem object shall have a sequence of RSIDs to identify projected reference systems from which it is composed.

Rec. 3 The RSID should follow a general pattern of [spatial axes] ([temporal axes]) ([homogeneous Weight]) ([parameter axes])

6.2.4.4 ttribute dimension

The attribute dimension gives the number of offsets in the coordinate array that is the same as the number of axes in the system. The ordinates and the axes are in the same order, each ordinate offset corresponding to the same offset in the axis array.

```
dimension: Integer
```

© Standards Australia Limited/Standards New Zealand 2020

Documents Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

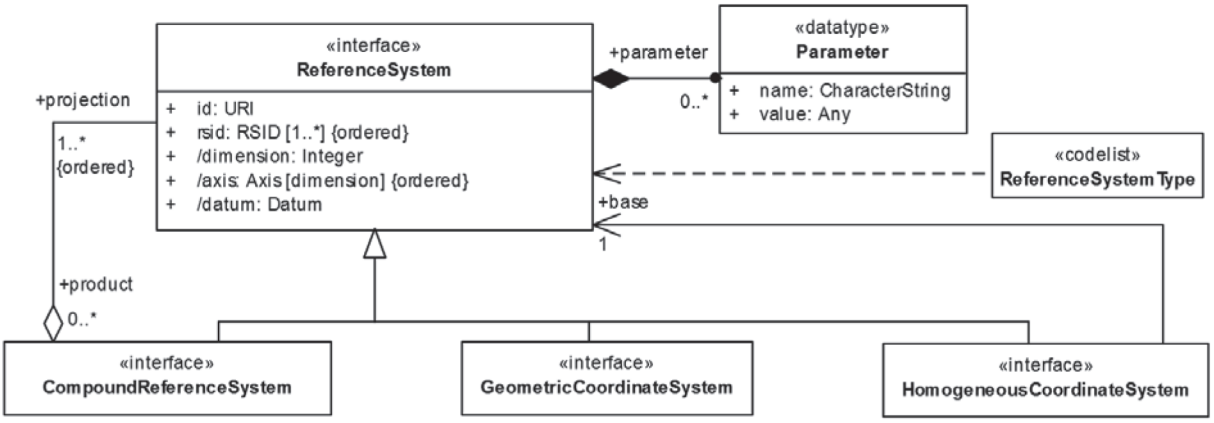

Figure 4 — Reference systems

6.2.4.5 Attribute axis

The array axis lists the various axes from the identified reference systems (in the rsid array). Axis groups from each RSID are grouped in the order given in the corresponding reference system.

```
axis: Axis [1..*]
```

6.2.4.6 Attribute datum

The attribute datum is the datum used for the spatial components of the geometry in this system.

```
datum: Datum
```

6.2.4.7 Role parameter

A ReferenceSystem can have direct connections to any number of parameters (name, value pairs) that may participate in the use of the system. Most parameters may be derived from the reference system IDs (RSID).

```
parameter: Parameter [0..*]
```

6.2.5 Codelist ReferenceSystemTypes

The codelist ReferenceSystemTypes is a list of all primitive systems supported by the local application

REQ. 26 **The local codelist shall list all primitive coordinate reference systems supported by the local application**

6.2.6 Interface CompoundReferenceSystem

6.2.6.1 Semantics

A compound reference system allows the implementer of geometry entities to concatenate coordinate tuples into single coordinate direct position values. Within a single set, all spatial coordinates should be from the same coordinate reference system.

REQ. 27 **A Reference System with more than 1 projection shall support the interface for a CompoundReferenceSystem which has an attribute projection consisting of an array of RSID for each projection.**

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020

EXAMPLE If a reference system consists of a 2D CRS to establish horizontal positions, a 1D vertical reference system for height and a 1D temporal reference system for time, then the rsid will consist of an RSID identifier for the horizontal CRS, a second RSID identifier for the vertical system, and a third RSID for a temporal system.

A compound coordinate system is one which is composed of a sequence (represented by an ordered association role) of other coordinate systems. The dimension of a compound system is the sum of the dimensions of its projections. The spatial dimension of the space is purely dependent on the projections that are to geographic positions in the real world.

This is a logical generalization of the CompoundCRS defined in ISO 19111, which is normally restricted to adding spatial or temporal dimensions. This interface does not restrict any additional columns, and may involve dependencies between coordinate offsets.

6.2.6.2 Role projection

The ordered association role "projection" of CompoundReferenceSystem lists the components of this instance of the interface, in the order that the coordinates are used in the compound reference system.

```
CompoundReferenceSystem::projection:ReferenceSystem[1..*]
```

6.2.6.3 Attribute rsid

The RSID identifiers of the ReferenceSystem instances in the projection role are repeated as an array of RSID system identifiers in the inherited attribute "rsid".

```
rsid: RSID [1..*]
```

NOTE The only reordering of axis or ordinates that is usable is in the essentially local-only GeometricCoordinateSystem that can rearrange the ordinates using its local attribute permutation.

6.2.7 Interface HomogeneousCoordinateSystem

Homogeneous coordinates are useful in defining rational b-splines and are used in projective geometry.

REQ. 28 **A HomogeneousCoordinateSystem shall be a GeometricCoordinateSystem that shall have an extra axis for a non-zero weight "w". All other affected offsets shall be multiplied by this "w".**

The usual mapping between the 2D base coordinate system to the corresponding homogeneous system is (x, y) → (x, y, 1) → (wx, wy, w), w ≠ 0 or in 3D (x, y, z) → (x, y, z, 1) → (wx, wy, wz, w), w ≠ 0, and so the usual projection back to the base system requires division by w as in (X, Y, W) → (X/W, Y/W, 1) → (X/W, Y/W) and so *W* must be non-zero. Since all the "classes" in this document are "«interfaces»", the internal representation of the coordinates is an implementation decision.

6.2.8 Interface GeometricCoordinateSystem

6.2.8.1 Semantics

The final step in creating a coordinate system for geometric calculation suitable for geographic measures is to combine the concepts in ISO 19111 in the CRS definitions, with the calculation mechanisms described above into a single interface (Figure 5).

REQ. 29 **Every geometry used to represent a locus of positions in a geographic space shall be associated with a coordinate reference system (CRS) as defined in ISO 19111 or one of its extensions, and to a GeometricReferenceSurface that shall be used in all "metric" calculations implemented by its object representations.**

Documents Document Sharing Hub

© Standards Australia Limited/Standards New Zealand 2020

Full document details: AS NZS ISO 19107-2020

REQ. 30 **If a geometry instance is associated with a GeometricReferenceSurface different from the datum of the associated CRS, then the metadata associated with that geometry shall make explicit reference to this inconsistency and shall describe an an error budget for errors introduced by this use of non-standard geometric reference system.**

Figure 5 describes the object model for these basic classes dealing with the relationship of geometry to its coordinate space. This is an information model, and any implementation of this model that contains the needed information regardless of the form will be compliant.

6.2.8.2 Attribute name

The attribute "name" is a human-readable name in the local language that is a mnemonic for the coordinate system.

```
GeometricCoordinateSystem::name:CharacterString
```

Rec. 4 **Information communities should standardize the Geometric Coordinate System names for common usage and provide or specify metadata locations for all such systems.**

6.2.8.3 Attribute spatialDimension

The attribute "spatial dimension" indicates the dimension of the spatial coordinate systems in use. Usual values for the spatialDimension are two and three. For spatial systems, the first two coordinates are usually "horizontal offsets". Examples include (latitude, longitude) in degrees and (easting, northing) in metres. The third dimension, at each point, is usually orthogonal, at least nominally, to the gravity isosurfaces; i.e. the usual 3rd coordinate is some form of height, often but not always in metres.

```
spatialDimension:Integer
```

REQ. 31 **Each spatial dimension shall be represented in or transformable into measures of length or distance.**

NOTE The spatial dimension of a coordinate space is the number of decisions made to locate a point in space. The spatial dimensions in a coordinate system are those decisions. So, in a (latitude, longitude) system, the spatial dimension is two, and the spatial dimensions or axes are latitude and longitude. See definitions 3.85 and 3.86.

© Standards Australia Limited/Standards New Zealand 2020

Documents
Document Sharing Hub

Full document details: AS NZS ISO 19107-2020