OGC® DOCUMENT: 21-999R1

External identifier of this OGC® document: https://github.com/opengeospatial/CRS-Deformation-Models/tree/master/products/specification

ABSTRACT SPECIFICATION TOPIC FOR OGC DEFORMATION MODEL FUNCTIONAL MODEL

ABSTRACT SPECIFICATION TOPIC

CANDIDATE SWG DRAFT

Version: 1.0
Submission Date: 2021-08-27
Approval Date: 2099-01-01
Publication Date: 2099-01-01
Editor: Chris Crook

Note

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

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

I ABSTRACT

All objects on the surface of the earth are moving. Apparently fixed features such as buildings are moving with the earth's crust — subject to ongoing plate tectonic movement and episodic deformation events such as earthquakes. Increasingly we rely on Global Navigation Satellite Systems (GNSS) which can precisely determine the coordinates of features — coordinates we then store in Geographic Information Systems (GIS) and other spatial databases. However, the coordinates from GNSS are referenced to global reference frames and coordinate systems In these reference frames the ongoing movement of apparently fixed objects means that the coordinates of these features are continually changing over time.

This creates a challenge for the geospatial and positioning community — how to account for this movement when comparing data sets observed at different times, or how to relocate an object located in the past.

One solution to this challenge is to use a deformation model — a mathematical model of the deformation of the crust. It allows the coordinates of an apparently fixed object to be propagated to different times so that they track the object's location.

Common uses of a deformation model include:

- determining the current location of an object based a historic measurement of its location,
- transformation of the current observed location of an object to the reference epoch of a static coordinate system
- determining the correct spatial relationship of data sets observed at different times.

Currently many national geodetic agencies construct deformation models. However, these generally use customized formats and are typically only used by software developed by the agency.

This specification defines a functional model for describing a deformation model. It describes a way of parameterizing deformation in such that it can be encoded into a data set and incorporated into software for use in coordinate transformations. It also defines how to calculate the displacement of a point between two different epochs using this parameterization. This specification allows producers of deformation models, such as national geodetic agencies, to publish these models in a consistent way. Users of coordinate transformation software can be confident that the model is being applied as intended by the producers.

II KEYWORDS

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, deformation model, crustal deformation, coordinate transformation, coordinate operation, specification

III SECURITY CONSIDERATIONS

No security considerations have been made for this document.

IV SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- Land Information New Zealand
- ESRI

V SUBMITTERS

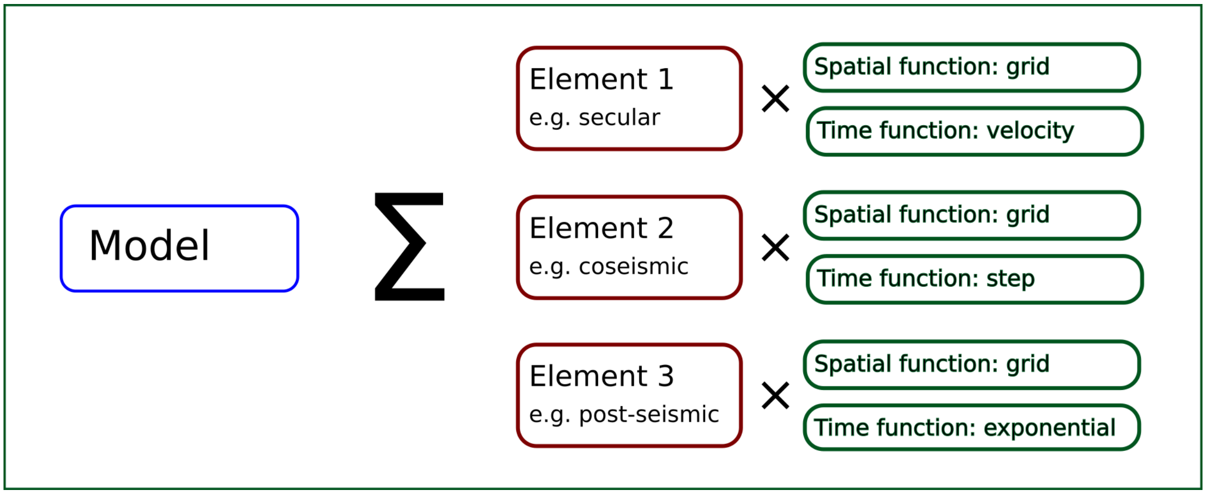
SUBMITTER	AFFILIATION
Chris Crook	Land Information New Zealand
Kevin Kelly	ESRI
…	…

VI INTRODUCTION

This specification has been developed by the OGC Coordinate Reference System (CRS) domain working group (DWG) in conjunction with the CRS DWG project team developing the “Gridded Geodetic Exchange Format (GGXF)” and with the International Association of Geodesy (IAG) Working group 1.3.1 on “Time-dependent transformations between reference frames in deforming regions”.

This specification defines a general parameterization of deformation models as illustrated in Figure 1. The total deformation is decomposed into a set of elements, each of which is defined by a spatial function and a time function. The spatial function defines the displacement that applies at a given location, and the time function defines a scalar function that applies for the element at a given time. The displacement due to the element is the spatial function displacement multiplied by the time function scale factor. The total displacement at the location and time is the sum of the displacements from each element.

Figure 1 — Functional decomposition of a deformation model



Commonly elements represent identified geophysical causes, such as coseismic deformation due to an earthquake. However, an element could equally represent some measured displacement for which no physical cause has been ascribed. Also, several elements may represent different aspects of a single geophysical cause. For example, the horizontal deformation due to an earthquake may be represented by one element, and the vertical deformation by another.

This specification does not encompass other forms of deformation model such as:

- Rigid body transformation between reference frames such as time dependent Helmert or Bursa-Wolf transformations,
- Trajectory models for individual points, such as ITRF reference station coordinate models,
- Static distortion models used to represent the deformation between two datums
- Global plate tectonic models
- Three dimensional geophysical models of deformation in terms of dislocations on fault surfaces or using finite element models.

Also, this specification does not describe how the parameters and metadata of a deformation model should be encoded into a file.

The primary use case for this specification is for deformation model producers such as national geodetic agencies to encode regional deformation models for use in spatial data operations in geographic information systems and positioning systems. The purpose of this specification is to define the parameters that must be encoded in a compliant deformation model, and the formulae that compliant software should use to transform coordinates using the model.

1

SCOPE

1 SCOPE

This abstract specification defines a functional model that is used to define deformation models — models of the ongoing deformation of the earth's crust — to support their implementation into coordinate operations and transformations.

Within this context a deformation model is defined as follows:

> A deformation model is a model of the displacement of the earth's surface within a defined spatial and temporal extent. It predicts the location of a point fixed to the surface at any time within the temporal extent in terms of an accessible coordinate system. The location of the point is represented as a displacement from a reference position for that point.

In the most abstract sense a deformation model is a mathematical function of a two dimensional coordinate and a time that defines a one, two, or three dimensional coordinate displacement. It may also express the uncertainty of the displacement.

This specification describes a means of representing a deformation model using a set of parameters and gridded data that defines this mathematical function. It specifies how the displacement is calculated from this data and how it is applied to a coordinate to apply the coordinate operation that the deformation model defines. It also specifies metadata attributes that should accompany the encoded model to allow users to understand its applicability and limitations.

2

CONFORMANCE

2 CONFORMANCE

NOTE: Provide a short description of the content approached in subsequent sections and the main subject of the document

3

NORMATIVE REFERENCES

3 NORMATIVE REFERENCES

There are no normative references in this document.

4

TERMS AND DEFINITIONS

4 TERMS AND DEFINITIONS

No terms and definitions are listed in this document.

5

DEFORMATION MODEL DEFINITION

5 DEFORMATION MODEL DEFINITION

The deformation model is characterized as follows.

a) The model defines deformation within a specified spatial and temporal extent. Beyond these extents the model is undefined and cannot be evaluated.

b) The model partitions deformation into one or more elements. The total displacement at a specific time and position is calculated by summing the contribution from each element at that time and position, as described in Clause 6.3.

c) Each element of the model comprises a spatial function and a time function. The spatial function evaluates displacement as a function of position. The time function evaluates a scale factor as a function of time. The deformation contribution for the element at a given position and time is obtained by multiplying the displacement from the spatial function by the scale factor from the time function.

d) The spatial function may be defined over a subset of the total extent of the model. Where the spatial function is not defined it is assumed to be zero.

e) The spatial function is represented by displacement vectors at the nodes of a grid or a set of grids.

   If a set of grids is used then only one of the grids will be used to evaluate the function at a specific location. The implementation must unambiguously define which grid (if any) is to be used at a specific location.

   The spatial function displacement components at a location are calculated from the selected grid using a specified interpolation method. Supported interpolation methods are described in Clause 6.1.

f) The spatial function can define displacement and uncertainty components. This set of components must be defined at every node of every grid of the function. Different spatial functions may have different components. For example, the spatial function of one element might define horizontal displacement, and the spatial function of another may define vertical displacement.

g) Displacement components of the spatial function may include:

   - horizontal displacement — the east and north displacement in meters or degrees
   - vertical displacement — the upwards vertical displacement in meters.

   Displacement components not defined by the spatial function are assumed to be zero.

h) Uncertainty components of the spatial function may be one of:

- horizontal uncertainty specified as one of:
  - horizontal circular uncertainty e_h
  - east and north uncertainty e_e, e_n
- vertical uncertainty e_v

The set of uncertainty components used to represent uncertainty are defined in the model metadata and must be used for all spatial functions which have uncertainty components. For example, the deformation metadata could specify that the model uses horizontal circular uncertainty and vertical uncertainty. A particular spatial component could define just horizontal circular uncertainty. Another could define verical uncertainty. But it would not be valid for an element to define east and north uncertainty.

The probability level for the uncertainty is defined in the model metadata, e.g. “1 standard error”, “95% confidence”.

If the uncertainty of an element is not defined at a model node its value is defined by a default value element metadata or, if it is not defined there, in the model metadata.

i) The displacement defined by the model is required to be continuous and invertible within the spatial and temporal extent of the model except where it evaluates to *no data*. This is not necessarily enforced by the mathematical formulation of a model. It is a compliance requirement on producers of functional models. Continuity can be assumed by software implementations of the model.

j) The displacement and uncertainty components at a node of a spatial function may be assigned a *no data* value. This means that their value at that node is undefined. The model displacement and uncertainty is undefined at specific time and location if calculating it would require using a using a *no data* value.

k) The time function of an element is a scalar function of time calculated as the sum of one or more base functions. Each base function is one of:

- a velocity function
- a step function
- a reverse step function
- a ramp function
- an exponential decay function
- a logarithmic function
- an acceleration function
- a hyperbolic tangent function
- a sine or cosine function

These base time functions are defined mathematically in Clause 6.2.

l) The model may define a 14 parameter Bursa-Wolf transformation definition. This is applied after the displacement calculated from the deformation elements in a forward transformation, as described in Clause 6.6. It is applied before the displacement calculated from the deformation elements in an inverse transformation.

m) Each element definition specifies:

- The spatial interpolation method to use (for example bilinear, bicubic)
- The quantities it defines (displacement components, uncertainty components)
- A spatial definition of the extent of the spatial function (to determine if it is required at a specific position)

It may also include:

- The default uncertainty, which applies if the spatial function does not explicitly define uncertainty.
- Other metadata required by the implementation
- Other producer metadata

n) The model definition specifies:

- The source CRS definition (e.g., EPSG:xxxx)
- The target CRS definition (if the model is implemented as a point motion model this will be the same as the source CRS).
- The spatial function definition CRS
- The valid spatial extent of the model (defined in terms of the definition CRS)
- The valid time extent of the model
- The units of horizontal displacement
- The units of vertical displacement
- The parameters used to represent uncertainty, for example, horizontal 95% circular confidence, vertical 95% confidence level.
- The default uncertainty for each element of the model, if the model does not explicitly define uncertainty

It may also include:

- Other metadata required by the implementation, such as discovery metadata and licence information.
- Other producer metadata, such as model name, model version and publication date.

6

CALCULATION FORMULAE

6 CALCULATION FORMULAE

The formulae below define the expected behavior of software implementations of the deformation model. Whatever algorithm a software implementation may use it should produce numerically indistinguishable results to be a compliant implementation. This ensures the software implementation applies the model as intended by the producers.

In this specification the grid can be defined either in terms of a geographic (longitude/latitude) or projected (easting/northing) coordinate system. Displacements and uncertainties are expressed as linear distance except that the horizontal displacement may be an angular distance if the source and target coordinate systems are geographic.

6.1. SPATIAL INTERPOLATION

For spatial function and interpolation types defined in this specification, the position of the calculation point is used to identify a set of grid nodes for calculating displacement at that point and the weights to be assigned to the displacement at each of those nodes. The displacement values at the nodes are combined using a simple weighted average as described in Clause 6.1.2. For the horizontal components (east and north displacement) near the geographic poles an alternative method of combining the node horizontal displacements is described in Clause 6.1.3.

6.1.1. Selection of spatial function grid

Spatial functions are defined as a grid or a set of grids. At a given position only one of the grids is used to calculate the spatial function displacement or uncertainty. The implementation must uniquely identify the grid, if any, that applies at any location within the valid extents of the model. If no grid applies at a location then the displacement and uncertainty of the element at that location are both zero.

6.1.2. Bilinear grid interpolation

Spatial functions may be represented as regular grids defined in an interpolation CRS. The coordinates of a grid node are given by

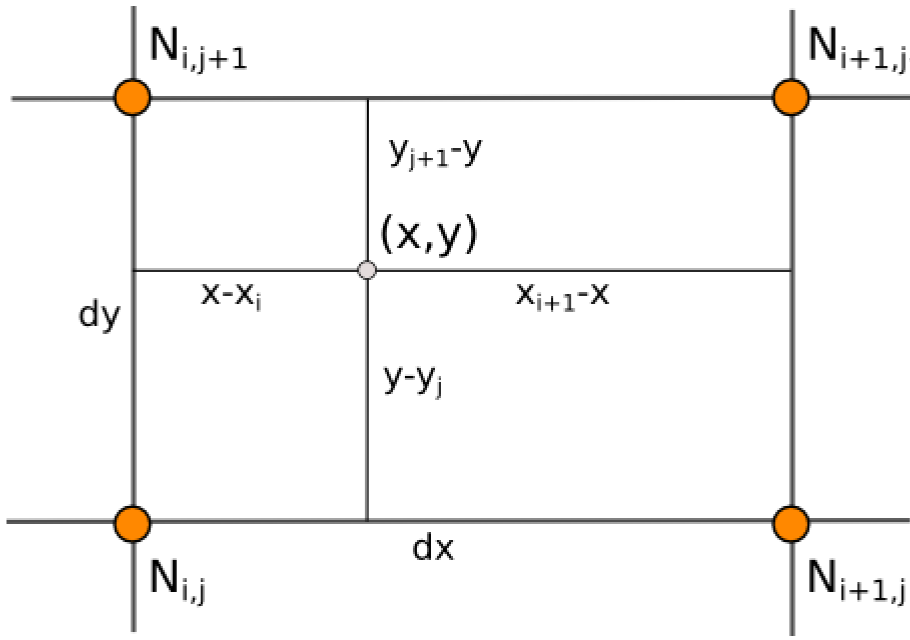
$x_i = x_o + i.x_s$
$y_j = y_o + j.y_s$

where x_o, y_o are the coordinates of a reference point of the grid, x_s and y_s are the grid spacing along the x and y axes respectively, and i and j are the column and row number of the grid node. Note that the x and y grid spacing need not be equal. For example, in a geographic interpolation coordinate system where x is longitude and y is latitude it may be preferable that

x_s is approximately equal to $y_s/\cos(y_m)$, where y_m is the latitude of the middle of the grid, as this makes the grid cells approximately square (except at polar latitudes).

Model components are calculated using bilinear interpolation with respect to x and y between the nodes at the corners of the grid cell within which the calculation point lies, as shown in Figure 2. Each element of the displacement is calculated independently.

Figure 2 — Bilinear interpolation on a grid cell



Bilinear interpolation is defined as follows:

The calculation point (x,y) is located in the grid cell between columns i and i+1, and rows j and j +1.

The weights are assigned to these nodes using the formulae:

$W_{i,j} = ((x_{i+1}-x)/x_s) * ((y_{j+1}-y)/y_s)$
$W_{i+1,j} = ((x-x_i)/x_s) * ((y_{j+1}-y)/y_s)$
$W_{i,j+1} = ((x_{i+1}-x)/x_s) * ((y-y_j)/y_s)$
$W_{i+1,j+1} = ((x-x_i)/x_s)*((y-y_j)/y_s)$

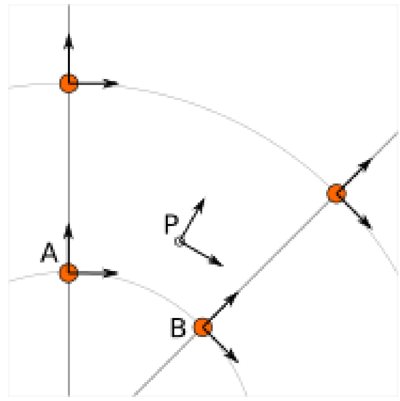
The model components at (x,y) are calculated as the weighted average of the values at the nodes as, e.g.:

$de = W_{i,j}.de_{i,j} + W_{i+1,j}.de_{i+1,j} + W_{i,j+1}.de_{i,j+1} + W_{i+1,j+1}.de_{i+1,j+1}$

6.1.3. Geocentric bilinear grid interpolation

A simple average of the east and north displacements may not be appropriate where a grid cell spans a wide longitude range, typically for grids in polar regions. In this situation the directions of the east and north vectors may be quite different at nodes being averaged, as shown in Figure 3 where their directions at nodes A and B and the evaluation point P differ significantly.

Figure 3 — East and north directions at near pole grid axes



In this situation the bilinear interpolation of vector components can be adapted by reparametrizing the vectors to a common set of axes before forming the weighted average.

The geocentric bilinear interpolation method converts the displacement components from east and north components to geocentric X, Y, and Z components. These are consistently aligned at the grid nodes and can be scaled and summed using the bilinear interpolation formulae in Clause 6.1.2 to calculate the X, Y, Z components of displacement at the calculation point, which are then converted back to east and north components at the calculation point.

At longitude λ and latitude φ the geocentric displacement components dx, dy, dz are calculated from the east and north components de, dn as:

$dx_{i,j} = -de_{i,j}.\sin(\lambda_{i,j}) — dn_{i,j}.\cos(\lambda_{i,j}).\sin(\varphi_{i,j})$
$dy_{i,j} = de_{i,j}.\cos(\lambda_{i,j}) — dn_{i,j}.\sin(\lambda_{i,j}).\sin(\varphi_{i,j})$
$dz_{i,j} = dn_{i,j}.\cos(\varphi_{i,j})$

The X, Y, and Z directions are the same at any location, so the dx, dy, and dz displacements can be interpolated independently using bilinear interpolation, e.g.:

$dx = W_{i,j}.dx_{i,j} + W_{i+1,j}.dx_{i+1,j} + W_{i,j+1}.dx_{i,j+1} + W_{i+1,j+1}.dx_{i+1,j+1}$

where the weights $W_{i,j}$ are as defined in Clause 6.1.2.

The displacement at the calculation point is then calculated as:

$de = -dx.\sin(\lambda) + dy * \cos(\lambda)$
$dn = -dx.\cos(\lambda).\sin(\varphi) — dy.\sin(\lambda).\sin(\varphi) + dz.\cos(\varphi)$

Note that this is only used to determine the horizontal displacement. The vertical displacement and uncertainties are computed using the bilinear formula above.

6.2. TIME FUNCTIONS

The time function f(*t*) for an element evaluates a scalar value by which the spatial function displacement is multiplied to determine the displacement at time *t*. For example, in a velocity model the spatial model represents the displacement that happens in one year and the time function evaluates the scale factor f(t) applied to the displacement at time t as the number of years since a reference epoch t_0, i.e. f(t)=(t - t_0).

The deformation model metadata defines a temporal extent for the model from T_{min} to T_{max}. At times before T_{min} and after T_{max} every time function is considered undefined and the model cannot be evaluated.

Time functions are parameterized as a function of decimal years. For example, velocities are expressed meters/year.

The conversion of a date/time *yyyy-mm-dd*T*HH*:*MM*:*SS* to decimal years is implemented as follows. The year number *yyyy* of the UTC epoch forms the integer part of the decimal year. The fractional part of the decimal year is determined by dividing the number of seconds between the beginning of the year *yyyy*-01-01T00:00:00Z and the date/time by the total number of seconds in the year (i.e. the number of seconds between *yyyy*-01-01T00:00:00Z and *yyy1*-01-01T00:00:00Z, where *yyy1* is *yyyy*+1).

Note that there is a small ambiguity in this formulation due to the occasional introduction of leap seconds. This impacts calculations because 1) it is not known at the beginning of the year whether a leap second will be added, and 2) standard software libraries used to implement the time functions may not include leap seconds, and if they do there will often be a delay before updates including leap seconds are distributed to users. Since leap seconds impart no practical difference to the deformation model calculations the decimal year is considered compliant whether or not it accounts for leap seconds — there are two nominally correct answers.

The time function is defined as the sum of one or more of the following base functions. In this table some functions include optional parameters which are indicated by italics. The default value for optional parameters is shown in parentheses where a value is required.

Table 1 — Time functions

TIME FUNCTION TYPE	PARAMETERS	FORMULA ($T_{I,MIN} \le T < T_{I,MAX}$)
velocity	Function reference epoch t_0 *Scale factor* s (1.0) *Start epoch* t_s *End epoch* t_e	f(t) = s.(t — t_0) f(t) = f(t_s) for t < t_s if t_s is defined f(t) = f(t_e) for t < t_e if t_e is defined
step	Reference epoch t_0 *Scale factor* s (1.0)	f(t) = 0 when t < t_0, f(t) = s when t ≥ t_0
reverse step	Reference epoch t_0 *Scale factor* s (1.0)	f(t) = -s when t < t_0, f(t) = 0 when t ≥ t_0

TIME FUNCTION TYPE	PARAMETERS	FORMULA ($T_{I,MIN} \le T < T_{I,MAX}$)
ramp	Start epoch t_s End epoch t_e *Start scale factor* s_s (0.0) *End scale factor* s_e (1.0)	$f(t) = s_s$ for $t < t_s$ $f(t) = (s_s.(t_e - t) + s_e.(t-t_s))/(t_e-t_s)$ for $t_s \le t < t_e$ $f(t) = s_e$ for $t \ge t_e$
exponential	Start epoch t_s Decay constant θ *Scale factor* s (1.0) *End epoch* t_e	$f(t) = 0$ for $t < t_s$ $f(t) = s.(1 - \exp(-(t-t_s)/\theta))$ for $t \ge t_s$ $f(t) = f(t_e)$ for $t > t_e$ if t_e is defined
logarithmic	Start epoch t_s Time constant τ *Scale factor* s (1.0) *End epoch* t_e	$f(t) = 0$ for $t < t_s$ $f(t) = s.\ln(1 + (t-t_s)/\tau)$ for $t_s \le t$ $f(t) = f(t_e)$ for $t > t_e$ if t_e is defined
acceleration	Function reference epoch t_0 *Scale factor* s (1.0) *Start epoch* t_s *End epoch* t_e	$f(t)= s.(t - t_0)^2$ $f(t) = f(t_s)$ for $t < t_s$ if t_s is defined $f(t) = f(t_e)$ for $t > t_e$ if t_e is defined
hyperbolic tangent	Start epoch t_s End epoch t_1 Time constant τ *Scale factor* s (1.0)	$f(t)= 0$ for $t < t_s$ $f(t) = s/2 + A.\tanh((t - t_r)/\tau)$ for $t \ge t_s$ and $t < t_e$ $f(t) = s$ for $t \ge t_e$ where: $t_r = (t_s + t_e)/2$ $A = s/2.\tanh((t_e-t_r)/\tau)$ $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$
cyclic	Frequency *f* (cycles per year) Function reference epoch t_0 Cosine scale factor α Sine scale factor β	$f(t) = \alpha\cos(f(t - t_0)/2\pi) + \beta\sin(f(t - t_0)/2\pi)$

The step and reverse step functions are specializations of the ramp function provided to improve simplicity and readability for the two most common uses of it.

Future versions of this specification may add new base functions as required.

6.3. COMBINATION OF ELEMENTS

To calculate the total displacement at a time and location, the displacement due to each element is calculated independently and summed. The total displacement is then applied to the coordinate. Displacement uncertainty is calculated similarly using the formula below. The same input position coordinate is used for each element. Deformation components are not applied sequentially, that is the input coordinate is not updated by the first element before being used to calculate the deformation of the second element.

At a given time and location the values obtained from each element are combined to determine the overall displacement. For example, if there are n components for which the spatial model calculates de as de_1, de_2, … to de_n, and the time function evaluates to f_1, f_2, … to f_n then the total displacement de is

$$de = f_1.de_1 + f_2.de_2 + \ldots + f_n.de_n$$

The uncertainty eh or ev is the root sum of squares (RSS) of the uncertainty values determined for each element. For example,

$$eh = \sqrt{(f_1^2.eh_1^2 + f_2^2.eh_2^2 + \ldots + f_n^2.eh_n^2)}$$

6.4. APPLYING THE TOTAL DISPLACEMENT TO A COORDINATE

The method used to add the calculated displacement to the reference coordinate is defined in the deformation model metadata. Two methods are defined — *addition* described in Clause 6.4.1 and *geocentric addition* described in Clause 6.4.2. The *addition* method simply adds the displacements to the coordinates. The *geocentric* method accounts for the difference between a linear east offset and a longitude offset in polar regions. It is only applicable if the displacements are expressed as linear distance (e.g. meters) and the source and target coordinate system are geographic.

If the interpolation coordinate system is directly related to the source or target coordinate systems then applying the displacement to a point may change its coordinate in the interpolation coordinate system, which in turn may change the calculated value of displacement. In this case the calculation and application of displacement to an input coordinate may require iteration, as described in Clause 6.5.

6.4.1. Addition method

The method of applying a displacement to a coordinate depends on the units of the displacement and the type of the source and target coordinate systems.

For geographic coordinate systems at locations near the poles the alternative Clause 6.4.2 may be more suitable.

If the source and target coordinate systems are projected coordinate systems then the units must be meters and the east and north displacements are simply added to the easting, northing coordinates.

If the source and target coordinate systems are geographic coordinate systems and the east and north displacement units are degrees, then the displacements are added to the longitude and latitude coordinates.

If the source and target coordinate systems are geographic and the east and north displacement units are meters then the displacement components must be converted to degrees before being added to the longitude and latitude coordinates. The conversion from meters to degrees requires the ellipsoid parameters of the geographic coordinate system.

If a is the ellipsoid semi-major axis (e.g. 6378137.0), f is the flattening (e.g. 1.0/298.25722210), λ is the longitude, and φ is the latitude then corrections to longitude dλ and latitude dφ (in radians) are given by:

$b = a.(1-f)$
$d\lambda = de.\sqrt{}(b^2\sin^2(\varphi)+a^2\cos^2(\varphi))/a^2\cos(\varphi)$
$d\varphi = dn.(b^2\sin^2(\varphi)+a^2\cos^2(\varphi))^{3/2}/a^2b^2$

The vertical displacement is always in meters and is simply added to the height coordinate.

6.4.2. Geocentric addition method

The geocentric addition method is a more complex method than the addition method which is suitable for geographic coordinate systems at locations close to the poles.

This method applies where the coordinates are in a geographic coordinate system and displacements are in meters.

The horizontal displacements are applied using the following procedure:

- the horizontal displacement components are converted to geocentric displacement components using the formulae in
- the geographic longitude and latitude are converted to geocentric cartesian coordinates X,Y,Z. For this calculation the ellipsoidal height is set to zero.
- the geocentric displacement components are added to the cartesian coordinates
- the cartesian coordinates are converted back to geographic coordinates. The resultant longitude and latitude are used in the final coordinate (the ellipsoidal height is discarded).

Note that the height coordinate is not updated by this method — a vertical displacement is added directly to the height coordinate.

TBC *Add formulae for conversion lat/lon to XYZ and vice versa*

Standard formulae are used to convert geographic coordinates to and from geocentric cartesian coordinates. The input ellipsoidal height is set to zero before converting to Cartesian coordinates, and the resultant ellipsoidal height after converting back to geographic coordinates is discarded.

6.5. ITERATIVE APPLICATION OF DISPLACEMENT

Calculating the inverse of the model requires an iterative solution if the interpolation coordinate system is dependent on the output coordinate system. The coordinate in the interpolation coordinate reference system is required to evaluate the displacement, but that coordinate is not known until the displacement has been calculated and applied to the input coordinate to derive the output coordinate.

This will apply in a forward transformation if the interpolation coordinate system is dependent on the target coordinate system, and in a reverse transformation if it is dependent on the source coordinate system.

The iterative calculation uses the following steps:

- use the input coordinate as an initial estimate for the output coordinate
- at each iteration:
  - use the current estimate of the output coordinate to determine the displacement that applies
  - apply this displacement to the input coordinate to obtain a new estimate for the output coordinate
  - calculate the difference between the current and new estimates of the output coordinate
  - if this difference is greater than the precision required for the inverse operation then iterate again, otherwise finish

6.6. CALCULATION OF THE 14 PARAMETER TRANSFORMATION

If the model includes a *14 parameter transformation* then this is applied to the coordinates after the model is calculated and applied in a forward transformation. In an inverse transformation it is applied before the model components are applied to the coordinate.

TBC *The 14 parameter transformation formulae need to be included here*

6.7. CALCULATION OF DEFORMATION BETWEEN TWO EPOCHS

The displacement de, dn and du to transform a coordinate between two epochs is calculated in a similar way to above except that the time functions are evaluated by taking the difference between the values at each epoch.

To calculate the displacement from epoch t_0 to epoch t_1 the time factors in Clause 6.3 are calculated for the i^{th} element as:

$f_{i,t1\text{-}t0} = f_i(t_1) - f_i(t_0)$

Note that for calculating displacement components this gives the same result as taking the difference between the displacement components calculated individually for each epoch. However for the calculating the uncertainty of the displacement between two epochs this method must be used to give the correct result.

A

ANNEX A (INFORMATIVE) NOTES FOR DEFORMATION MODEL PRODUCERS

A

ANNEX A (INFORMATIVE) NOTES FOR DEFORMATION MODEL PRODUCERS

A.1. DECOMPOSITION INTO ELEMENTS

This deformation model specification assumes that the deformation can be partitioned into one or more elements each of which is characterized by a time function. The same time function applies at every point in the element. The spatial model defines a displacement at each point which is multiplied by the time function so the effect is different at different locations, but it is the same time function.

Deformation is caused by a number of geophysical phenomena, each of which may be represented be one or more elements in a deformation model.

Typically, there is a secular velocity element representing the long term tectonic deformation — this is a single element with a velocity time function.

Overlaid on this there may be deformation due to earthquakes. The coseismic deformation will be represented by an element with a step time function. Postseismic deformation from the earthquake may be represented by several elements depending on the complexity of the deformation and the quality of information about it. Several versions of a deformation model may be published after an earthquake as the ongoing postseismic deformation evolves and as the measurements and modelling of deformation are refined.

Alternatively, the deformation model may be based more directly on observations. For example, in Japan the dense network of CORS stations provides a continuous record of deformation. Each year a new version of the deformation model is generated based on these measurements. Each of these could be represented by an element with a step time function.

A.2. REQUIREMENT FOR THE MODEL TO BE CONTINUOUS AND INVERTIBLE

Deformation models described by this specification are used for coordinate operations — either to define time dependent transformations between different coordinate systems or to track the motion of objects between different epochs within a single coordinate system.

Generally, the user expectation of these functions is that they are invertible. That is, data can be transformed between any two epochs, and transforming data from one epoch to another and then back again leaves it unchanged.

To support this expectation the deformation model should be spatially continuous and invertible for any valid epoch. This is specified above as a compliance requirement of a deformation model.

Within a single grid the use of bilinear interpolation ensures that the deformation is spatially continuous. It also ensures the deformation is invertible provided the deformation is not greater than the grid spacing, which is generally very unlikely.

The only opportunity for discontinuity in a deformation model is at the edges of grids.

This places two constraints on deformation model producers.

- where an element does not over the full extent of the deformation model then the deformation at the edge of the element must be zero. For example, an element might represent coseismic deformation from an earthquake. The deformation caused by the earthquake may not be significant over the full extent of the model so the element would only cover a subset of that area. To ensure continuity the displacements at the edge element should be set to zero, even though the actual displacements will be slightly different from zero.

- where an element is represented by multiple overlapping or nested grids there will be transitions between one preferred grid and another. At these transitions the values interpolated from each grid must be exactly the same. For example, the displacement at the edge of a child grid will generally have exactly match that of its parent grid. However if the edge abuts the edge of another child grid, then it displacement only needs to match that of the abutting grid.

In practice the second of these constraints means that parent and child grids must be aligned — any node of a parent grid that lies within the child grid must also be a node of the child grid. Where grids are abutting then along the common edge at least one of the grids should include all the nodes of the other.

These constraints, and indeed the use of a gridded model, mean that the model cannot represent the true deformation exactly. For example, where there is surface faulting the actual deformation may not be continuous across a fault line. The deformation also may not be invertible (at least as a function of horizontal position) in areas of thrust faulting where points originally on opposite sides of the fault may be moved to the same horizontal position (though at different heights).

For model producers, this can create a tension between the scientific desire to represent deformation as accurately as possible, and the practical constraints of a deformation model suitable for coordinate operations. This specification is framed within the context of coordinate operations — it is not purporting to model deformation for scientific use.

The challenge for model producers is to decide the extent to which the deformation can be usefully modelled to best support the geospatial community.

A.3. NO DATA AND ZERO VALUES

There may be areas within the extent of the deformation model grids where the deformation is either not known or is not within the model producer's jurisdiction. In these areas the model producer may prefer to not specify an unreliable or unauthoritative displacement.

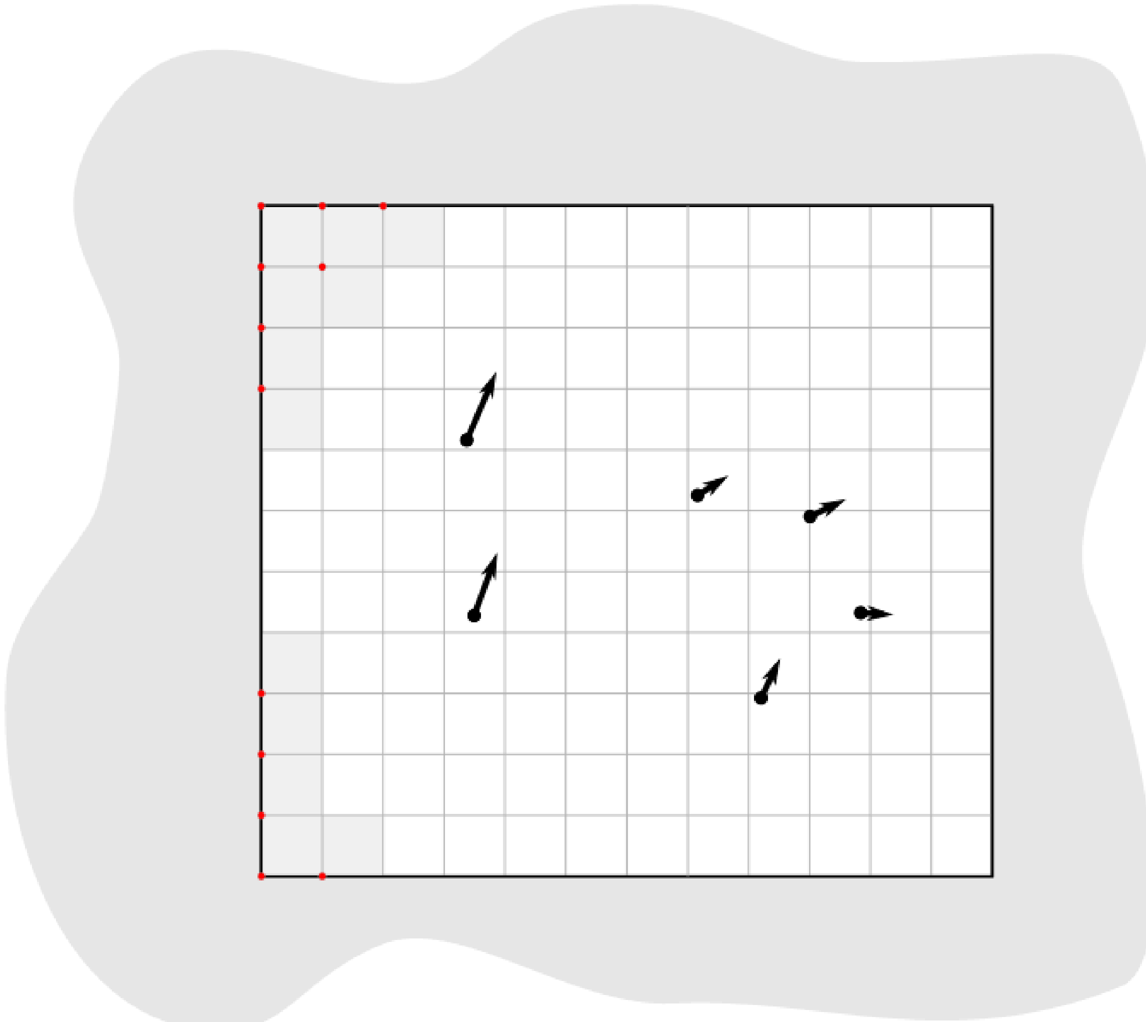
Grids must be defined by rectangular regions, which are unlikely to match the extent of the known deformation. To avoid grids being used in areas where the deformation is undefined, it is desirable that special "no-data" values can be used for displacements at nodes where this applies. The implementation of "no-data" will be dictated by the format used to carry the deformation model.

The "no data" value for a displacement is different from a zero displacement. A value of zero is used where there is no significant displacement. A "no-data" value is used where the displacement is unknown.

Instead of using "no data" values, a producer may choose specify a value with a large uncertainty. This may be preferable to ensure a continuous and invertible model, as discussed in Annex A.2. However not all software will use uncertainty information.

If any element of the model evaluates to "no-data" at a given location and time then the displacement (or uncertainty) at that location is undefined. Figure A.1 illustrates the use of "no-data" values in the model. In this figure the grey area shows the region where displacement is undefined and the square marks the total extent of the model. Outside this area the model cannot be evaluated. Within the gridded spatial function there are a number of nodes at which the displacement is not defined — these are identified with a "no-data" value. Where these nodes are required to interpolate a displacement, which is any grid cell they are on the boundary of, the displacement cannot be calculated..

Figure A.1 — Influence of “no data” values



The grey area, where displacement is undefined, might include coastal regions where the deformation of the seabed is not measured, or it may cross a jurisdictional boundary. Since the grid is rectangular, it may include regions where the deformation is not known, which are represented in the model by a “no data” value.

A.4. DEFORMATION MODELS NEAR POLES

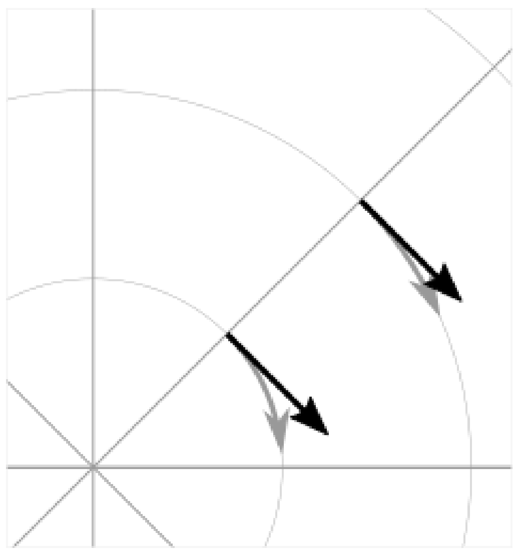
Special consideration may be required for deformation models using geographic coordinates near the poles. This specification provides some support for this situation. However, if possible it is simpler to use a projection coordinate system in polar regions.

Where the interpolation coordinate system is geographic then geocentric bilinear interpolation may be required for interpolating horizontal displacements across a grid cell. This is only required if the grid cells span a large longitude range. For example, if the grid cell spans 1° of longitude and the displacement is 1m then conventional bilinear interpolation may give rise to an error of up to about 2cm (the length of the displacement vector multiplied by the cosine of the longitude range of the grid cell). If this is an issue it may still be preferable to define a denser grid and use conventional bilinear interpolation.

Where the source and target coordinate systems are geographic then geocentric displacement addition is needed to correctly apply a calculated displacement to a coordinate near the pole. Note that this only applies very close to the pole.

This is illustrated in Figure A.2 where the grey vector shows the result of adding an east displacement to the longitude coordinate, and the black vector shows the result applying the same east displacement in the direction of the east vector component. Close to the pole, offsetting the longitude coordinate can give a significantly different result to applying a displacement in the direction of the east vector. At the pole itself, offsetting the longitude coordinate makes no difference to the location at all, and the geocentric displacement addition method must be used.

Figure A.2 — Comparison of vector and angular displacement near a pole



Moving away from the pole this discrepancy becomes less significant. For a point at distance R from the pole with a displacement d, the difference is approximately d*(1-cos(d/R)), or approximately $d^3/2R^2$. For example, applying 1 m east displacement 1 km from the pole by offsetting the longitude would incur an error of only 5.10^{-7}m.

If the deformation model doesn't include the pole itself then this method is not recommended as it is significantly more computationally intensive.

B

ANNEX B (INFORMATIVE) SPECIFICATION RATIONALE

ANNEX B (INFORMATIVE) SPECIFICATION RATIONALE

Note: This section contains notes from the development of this specification that will most likely be removed from the final document. These are retained here for the present until we need to remove them or find a better home for them!

B.1. SPATIAL FUNCTION TYPES

Audience: specification maintainers

Purpose: Present and explain choice

In practice nearly all current deformation models use grid representations. Consideration has been given to the inclusion of models based on triangulated networks, though the project team has decided not to include this capability in the initial specification.

In New Zealand triangulated models were considered for the deformation due to the 2011/2012 Christchurch earthquakes but they did not significantly reduce the size of the model ([Winefieldetal2010]). They are also less efficient to evaluate since it is necessary to search the triangulation to determine which triangle applies at a location.

The project team is not aware of any triangulated deformation models either in use or proposed for deformation coordinate operations. One datum transformation uses a triangulated model (Finland EPSG:2393 (KKJ / Finland Uniform Coordinate System) projected CRS to EPSG:3067 (ETRS89 / TM35FIN(E,N))). This is similar to transformations implementing deformation, and indicates that triangulated deformation models may play a role in the future.

The project team briefly considered including global plate motion models in the specification. These are defined by a set of polygons defining the extent of each tectonic plate. A rigid body rotational velocity around the center of the earth is defined for each plate. Plate motion models exhibit discontinuities at the boundaries between plates. These models are often used in GNSS analysis to predict the movement of survey marks. Since this specification focuses on regional deformation, plate motion models were considered out of scope for this work.

Another potential future development is the use of Discrete Global Grid Systems to define a global grid with a heterogeneous level of detail. As these acquire more support in software then this may be worth considering.

B.2. QUALITY PARAMETERS AT SPATIAL FUNCTION NODES

Audience: specification maintainers

Purpose: Retain thoughts on undecided issue

Comment: This is really a placeholder at the moment. We haven't reached a consensus on how this would be implemented. The primary use case is identifying areas near surface faulting where the actual deformation is not well represented by the model.

See also:

- https://github.com/opengeospatial/CRS-Deformation-Models/issues/25
- https://github.com/opengeospatial/CRS-Deformation-Models/issues/29

The project team identified an interest in having a quality parameter defined at nodes. The main driver is to identify where there is surface faulting causing distortion or discontinuities that are not well represented by the deformation model.

The quality parameter could be implemented in software to warn users when the coordinate conversion quality is compromised by such distortion.

Two different options for encoding a quality parameter have been considered. One is to define a quality measure parameter at each grid node. Another is to define this in the model metadata with a list of polygons defining areas of poor quality, each with an associated time at which the distortion occurred.

There are unresolved issues in using the quality parameter, including:

- if the parameter is defined for grid nodes how should it be interpolated to provide a quality measure at an interpolated point and how should the time of conversion be used? If the time function evaluates to zero it should clearly be ignored. However it is not clear how the parameter will apply when the time function is very close to zero.
- Uncertainty parameters give a quantitative measure of quality of the model. How should the quality parameter affect calculating the uncertainty of the deformation?

B.3. TIME FUNCTIONS

Audience: specification maintainers

Purpose: Present and explain choice

The proposed set of base time functions includes those commonly used in geophysical models. However in practice complicated time functions using multiple base functions may not represent the deformation much better than simple functions, as it is unlikely that the same time function will apply at all points in an area affected by, for example, post-seismic deformation. The actual time evolution may have different attributes and parameterization at different points. Any element represents a best fit simplification of the actual deformation over its spatial and temporal extent.

There is some redundancy in the selected set of base functions. In particular a velocity function including a start and end epoch is functionally identical to a ramp function. However these two options are provided to support quite different use cases.

Typically the velocity function will not be specified with both a start time and an end time. It represents secular deformation and the displacements from the spatial function quantify deformation accumulating in one year regardless of the start and end time. A velocity function with just a start time would be appropriate where a deformation event causes a velocity change of indefinite duration.

The ramp function is used for specific time bounded events. It more meaningfully represents specific deformation events as the spatial function displacements reflect the total deformation rather than the rate of deformation. Also it supports the step function specialization which is not possible with a velocity function (as the velocity is infinite). Multiple ramp functions can be combined to approximate any time evolution.

In the near future it is likely that we may generate far more complex and accurate models using technology such as CORS and InSAR. The deformation model representing this would most likely have multiple elements, each with its own spatial function and time function, rather than a complex time function applying to a single spatial function. For example each year there could be an updated gridded spatial function. The deformation at any epoch could be interpolated or extrapolated from the nearest to models (or as in Japan modelled with a step function for each year). This is in effect a three dimensional grid with dimensions latitude, longitude, and time. It can be easily encoded into this model by constructing time functions for each grid that define the interpolation between one grid and the next.

B.4. SEQUENTIAL OR PARALLEL EVALUATION OF ELEMENTS

Audience: specification maintainers

Purpose: Present and explain choice

The calculation formulae above use the same input coordinate to calculate the deformation for each element.

An alternative approach that could be used is to compute deformation elements sequentially. In this approach, the first element is calculated and applied to the starting coordinate, and then the updated coordinate is used to calculate the second element, and so on. This may result in a different final coordinate to the proposed method, as the second and subsequent elements are evaluated at different locations.

From a theoretical point of view neither method is necessarily more correct. The main reason for specifying one approach is to ensure that there is an "authoritative" correct value, particularly where the model is integral to the definition of a datum (as in New Zealand for example).

If the elements are an ordered sequence of discrete events then the sequential approach might seem more intuitive. However, this is not necessarily the case. For example, consider a model where the first element is a velocity function and the second is a step at 2003-01-01. If displacement is calculated at 2004-01-01, the velocity function is applied as at 2004, and then that coordinate is used for the step function. If the deformation is calculated at 2014-01-01, then the velocity function is applied as at 2014, and a different coordinate is used to interpolate the step function. This means that the contribution from the step function could be different even though nothing else has changed other than the evaluation epoch.

In practice the choice of independent or sequential evaluation of elements should yield an insignificant difference to the coordinates. Independent element evaluation has the following advantages:

- using the same input coordinates is slightly more efficient as the calculated displacement need only be applied to the coordinate once. This could make a significant difference if the horizontal displacement is applied using the "geocentric" method. Coordinate differences are insignificant if the displacement is obtained by simple addition.
- using the same input coordinates for each element provides an opportunity for parallelising calculation of elements.
- using the same input coordinates for each element allows optimising transformations between two versions of the deformation model since elements common to both versions can be ignored.

B.5. SIGNIFICANCE OF ITERATION FOR THE INVERSE DEFORMATION MODEL EVALUATION

Audience: specification maintainers

Purpose: Present and explain choice

The error incurred by not iterating the inverse transformation is evaluated for the New Zealand NZGD2000 deformation model.

The most complex deformation in New Zealand is in the Kaikoura region resulting from the 2016 Kaikoura earthquake. Coordinates here have been updated with “reverse patching” and the inhomogeneity of the deformation field primarily affects pre-earthquake transformations. Testing across the fault zone finds that the maximum error caused by not iterating an inverse transformation of epoch 2000.0 coordinates is about 0.015 meters. However, this error is in an area where the deformation model is inaccurate — it has been smoothed across the fault zone and exhibits errors of several decimeters. For transforming epoch 2019.0 coordinates the maximum error is well below 1 millimeter. In the North Island, an area largely unaffected by episodic events, the maximum error is about 0.2 millimeters.

Based on these results it is recommended to iterate the inverse transformation. Although this may increase computation time, in most cases two iterations will suffice While the iteration will not improve transformation accuracy, it ensures that the inverse transformation returns identical coordinates as its corresponding forward transformation.

Note that this is not about creating a more accurate transformation — the differences are much less than the uncertainty in the deformation model. The reason for iterating is to satisfy a user expectation that applying a transformation followed by the inverse transformation will result in coordinates that are materially unchanged.

B.6. TRANSFORMING DATA BEYOND THE DEFORMATION MODEL

Audience: implementors, specification maintainers

Purpose: Retain thoughts on undecided issue

Software implementations of the model may need to transform data that extends beyond the model boundary. If, as is likely, the deformation is not zero at the edge of the model then there is discontinuity across the boundary. There are a number of possible approaches to handling this including:

- Require that valid models have zero deformation at the boundary. Deformation model producers may have to create an artificial buffer around their area of interest and calculate an unreal deformation field that reduces to zero at the outer edge of the buffer. The model could also include uncertainties which are larger in the buffer to indicate that data is not reliable there.
- Specify (or recommend) algorithms for transforming data beyond the edge of the model that smooth out the discontinuity. Model metadata could include parameters to support the implementation, for example a width of the smoothed

region. The algorithms could also specify how uncertainty is calculated to reflect this.

- Specify that transformation of data beyond the extent of the model is not permitted, and will result in an exception (or equivalently a “no-data” value).
- Do not specify a behaviour — implementations can choose if and how to transform data outside the extent of the model. Transformations beyond the extent of the model are considered out of scope of this specification.

From a model producer’s perspective the third of these options, fail if transformation beyond the model extent is attempted, is the correct approach. Also, model producers may not be concerned about transformations beyond their jurisdiction, so that any of the last three options could be acceptable. The first option — building a model with information that is known to be incorrect — is not desirable. While this might be mitigated to an extent by increasing the uncertainty of the model in these regions, most current software does not consume or report uncertainty information, so the user may be misled into thinking that the transformation is accurate.

From a user’s point of view, having a transformation fail beyond the extent of the model could be undesirable. For example, they may have features or observations that include points both inside and outside the extent of the model which are observed at different times and which they want to compare accurately within the extent of the model. Trimming the features to the extent before doing this would be inconvenient. However, users need to be aware of potential inaccuracy in the comparison beyond the model extent. This could be further complicated if the features span more than one deformation model. Global deformation models may mitigate this problem, but no suitable models exist at the time of writing.

Also for users it is important that different software implementations give the same result.

BIBLIOGRAPHY

BIBLIOGRAPHY