OGC® DOCUMENT: 22-010

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

Open Geospatial Consortium

DEFORMATION MODEL FUNCTIONAL MODEL

ABSTRACT SPECIFICATION TOPIC

CANDIDATE SWG DRAFT

Version: 1.0
Submission Date: 2022-05-10
Approval Date:
Publication Date:
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, being 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 and use in Geographic Information Systems (GIS) and other spatial databases. However, the coordinates from GNSS are referenced to global reference frames and coordinate reference systems (CRSs). In these reference frames ubiquitous movement of fixed objects means that the observed coordinates of these features are continually changing.

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 locate an object observed 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 a 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 on historic measurement of its location,
- transformation of the current observed location of an object to the reference epoch of a static CRS,
- determining the spatial relationship of data sets observed at different times,
- predicting the location of an object at some future time.

Currently, many national geodetic agencies constructed or are planning for deformation models. However, these current models generally use customized formats and are typically only used in software developed by the agency.

This specification defines a framework for describing a deformation model. We call this framework a functional model. It describes a way of parameterizing deformation such that it can be encoded into a data set and incorporated into software for use in coordinate operations. It also defines how to calculate the displacement of a point between two different epochs. This specification allows producers of deformation models, such as national geodetic agencies, to publish these models in a consistent way. Users of software that performs coordinate operations, such as coordinate transformations, 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, time-dependent transformation

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

Table — List of submitters

SUBMITTER	AFFILIATION
Maciej Bazanowski	Geographic Information Services, Department of Service Nova Scotia and Internal Services
Dana Caccamise	NOAA
Michael Craymer	Canadian Geodetic Survey, Natural Resources Canada
Chris Crook	Land Information New Zealand
Muzaffer Kahveci	Konya Technical University
Kevin M. Kelly	Esri
Roger Lott	IOGP
José Antonio Tarrío Mosquera	University of Santiago de Chile
Chris Pearson	Trimble Inc
Craig Roberts	University of New South Wales
Keith Ryden	Esri
Richard Stanaway	University of New South Wales

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 (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. This model is described in detail in Clause 5.

Commonly, elements represent the effects of geophysical events, 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 event. For example, the horizontal deformation due to an earthquake may be represented by one element, and the vertical deformation by another.

Other types of deformation model not encompassed by this specification include:

- 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 difference between two datums,
- global plate tectonic models,
- three dimensional geophysical models of deformation in terms of dislocations on fault surfaces such as elastic half space or finite element models.

This specification defines a set of parameters and metadata that define the deformation model, and how they are used to apply that model in coordinate transformations. It does not prescribe how they should be encoded into a digital file.

The intention of this specification is to allow deformation model producers, such as national geodetic agencies, to publish models, and consumers to use them, with confidence that the model is being correctly interpreted and used.

2

CONFORMANCE

2 CONFORMANCE

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

1

SCOPE

1 SCOPE

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

For the purposes of this document 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 CRS. 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 provides a framework for representing a deformation model using a set of parameters and gridded data. It specifies how the displacement is calculated from these data and how it is used in a coordinate operation. It also specifies metadata that should accompany the encoded model to allow users to understand its applicability and limitations.

3

NORMATIVE REFERENCES

3 NORMATIVE REFERENCES

There are no normative references in this document.

4

TERMS, DEFINITIONS AND ABBREVIATED TERMS

4 TERMS, DEFINITIONS AND ABBREVIATED TERMS

This document uses the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

For the purposes of this document, the following additional terms and definitions apply.

4.1. Terms and definitions

4.1.1. **producer**

agency compiling a deformation model for publication

4.1.2. **carrier**

specific format used to encode a deformation model.
Note 1 to entry: The carrier defines how the attributes of the deformation model are encoded.

4.1.3. **implementation**

software that uses an encoded deformation model to calculate displacements and apply coordinate operations.
Note 1 to entry: The implementation encodes the formulae defined in this specification or numerically equivalent formulae.

4.2. Abbreviated terms

CRS	coordinate reference system

5

DEFORMATION MODEL DEFINITION

5 DEFORMATION MODEL DEFINITION

The deformation model is characterized as follows.

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

2. 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.

3. 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. This structure is illustrated in Figure 1.

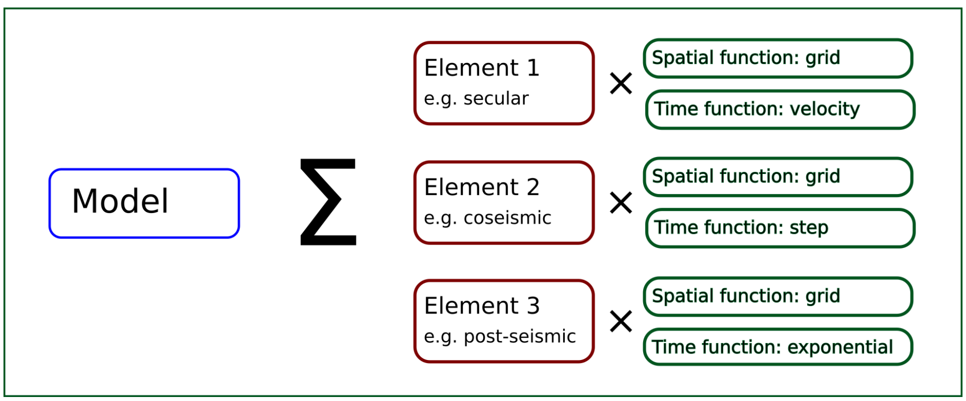

Figure 1 — Functional decomposition of a deformation model

4. The spatial function may be defined over a subset of the total extent of the model. Where the spatial function is not defined within the extent of the model it is assumed to be zero. This allows an element to represent a deformation event that does not affect the full extent of the model.

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

   If a set of grids is given, 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.

6. The spatial function displacement components at a location are calculated from the selected grid using a specified interpolation method. The only interpolation method explicitly supported is bilinear interpolation, as described in Clause 6.1.

   A producer may specify other formulae supported by implementations, but this specification provides no assurance of the displacements an implementation will calculate from such a model.

7. 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.

8. 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.

9. Uncertainty components of the spatial function may include:

   - 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 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 vertical 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 grid node, then the uncertainty is taken from a default value in the element metadata or, if it is not defined there, in the model metadata.

10. The displacement defined by the model is required to be continuous 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 deformation models. Continuity can be assumed by software implementations of the model.

11. 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. A calculated displacement or uncertainty is considered undefined at a specific time and location if the calculation involves a no data value.

12. 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 ramp function
    - an exponential decay function
    - a logarithmic function
    - an acceleration function
    - a hyperbolic tangent function
    - a cyclic function

    These base time functions are defined in Clause 6.2.

13. Each element definition specifies:

    - The spatial interpolation method to be used (currently only bilinear is supported)
    - 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 that applies if the spatial function does not explicitly define uncertainty.
    - Other metadata required by the implementation
    - Other producer metadata

14. The model definition specifies:

- The source CRS
- The target CRS (if the model is implemented as a point motion model this will be the same as the source CRS).
- The interpolation CRS used to define the spatial function(s)
- The valid spatial extent of the model (defined in terms of the interpolation 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, used if the element does not explicitly define uncertainty

It may also include:

- Other metadata required by the implementation, such as discovery metadata and license 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. A compliant software may use different formulae provided that for every location and time within the extents of a “realistic” deformation model the displacement calculated by the software is “practically identical” to that calculated from these formulae. The terms “realistic” and “practically identical” are imprecise. Within the context of deformation model usage, displacement differing by less than 0.1mm can be considered “practically identical”.

Grids can be defined either in terms of a geographic (latitude/longitude) or projected (easting/northing) CRS. Horizontal displacements and uncertainties are expressed as linear distances or, if the source and target CRSs are geographic, an angular measure. Vertical displacements are expressed in linear units.

6.1. Spatial interpolation

This specification defines spatial functions by bilinear interpolation on a grid or set of grids. In the future other interpolation methods or spatial representations may be supported.

6.1.1. Selection of spatial function grid

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

6.1.2. Bilinear grid interpolation

Each grid of a spatial function is a regular grid in the interpolation CRS. The coordinates of a grid node are defined as

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

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 CRS where y is latitude and x is longitude, 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.

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 component of the displacement is calculated independently.

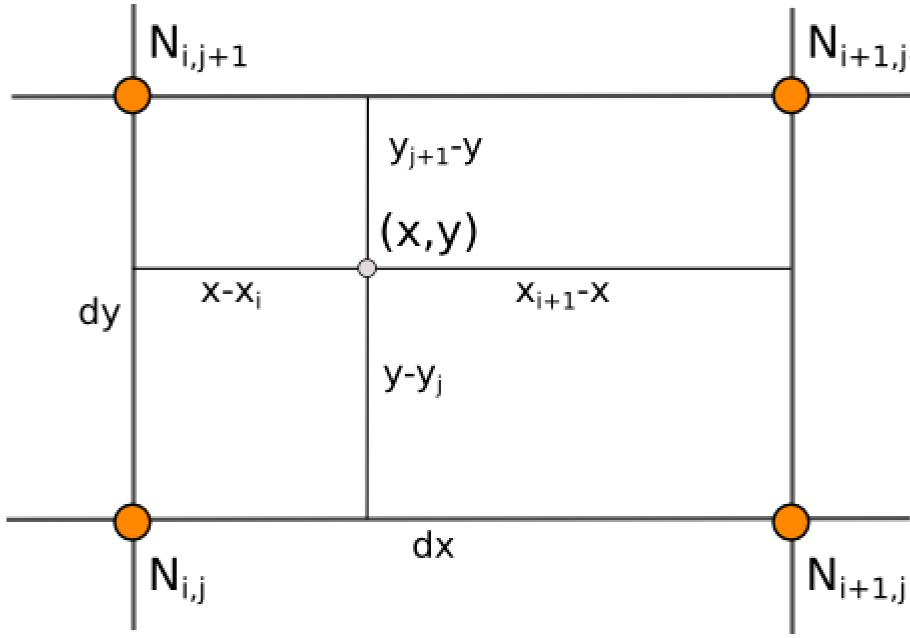

Figure 2 — Bilinear interpolation on a grid cell

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.

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.:

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

Uncertainties are interpolated from the grid node values using the same formulae.

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 function 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.

The time function is defined as the sum of one or more base functions.

Each base function is one of the types listed in the following Table 1.

Table 1 — Base time function types

TIME FUNCTION TYPE	DESCRIPTION
velocity	Defines a scale factor that is linear with time. When multiplied by the spatial function, it defines a constant velocity field, typically used to represent secular tectonic velocity.
acceleration	Defines a rate of change of the velocity function. This is sometimes used to model glacial isostatic adjustment.
step	Defines an instantaneous displacement, typically used to model coseismic earthquake deformation.
ramp	Defines displacement accumulating at a linear rate over a fixed period. This can be used to represent post-seismic deformation or slow slip events. Combining several ramp functions creates a piecewise linear time function that can emulate any time behavior to an arbitrary level of accuracy.
exponential	Defines a displacement accumulating at an exponentially decaying rate after an event. This is commonly used to approximate post-seismic deformation. Often used in conjunction with a logarithmic model.
logarithmic	Defines a displacement accumulating proportionally to the logarithm of elapsed time since an event. This is commonly used to approximate post-seismic deformation. It is often used in conjunction with an exponential model.
hyperbolic tangent	Approximates time behavior observed in slow slip events.
cyclic	Represents cyclic behavior, such as deformation due to seasonal hydrological loading.

The following table lists the reference formulae for each type of time function.

Table 2 — Time function reference formulae

TIME FUNCTION TYPE	FUNCTION PARAMETERS	FORMULA ($T_{I,MIN} \le T < T_{I,MAX}$)
velocity	Function reference epoch t_0	$f_r(t) = (t - t_0)$

TIME FUNCTION TYPE	FUNCTION PARAMETERS	FORMULA ($T_{I,MIN} \leq T < T_{I,MAX}$)
acceleration	Function reference epoch t_0	$f_r(t) = (t - t_0)^2$
step	Event epoch t_v	$f_r(t) = 0$ when $t < t_v$, $f_r(t) = 1$ when $t \geq t_v$
ramp	Start epoch t_s End epoch t_e	$f_r(t) = 0$ for $t < t_s$ $f_r(t) = (t-t_s)/(t_e-t_s)$ for $t_s \leq t < t_e$ $f_r(t) = 1.0$ for $t \geq t_e$
exponential	Event epoch t_v Time constant τ	$f_r(t) = 0$ for $t < t_v$ $f_r(t) = (1 - \exp(-(t-t_v)/\tau)$ for $t \geq t_v$
logarithmic	Event epoch t_v Time constant τ	$f_r(t) = 0$ for $t < t_v$ $f_r(t) = \ln(1 + (t-t_v)/\tau)$ for $t \geq t_v$
hyperbolic tangent	Event epoch t_v Time constant τ	$f_r(t) = (1 + \tanh((t - t_v)/\tau))/2$ where $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$
cyclic	Frequency *f* (cycles per year) Function reference epoch t_0	$f_r(t) = \sin(f(t - t_0)/2\pi)$

Each base time function is computed using the reference formula $f_r(t)$ as in Table 2. The result can be modified by three epoch parameters, start epoch t_s, end epoch t_e, and function reference epoch t_0, and a scale factor s. (Note that the ramp function explicitly uses the start and end epoch; the velocity, acceleration, and cyclic functions explicitly use the function reference epoch.)

The base time function $f(t)$ is evaluated from the reference function $f_r(t)$ using these epochs as follows:

- If the start epoch is defined, then the function value at the start epoch applies for all times before the start epoch. If the end epoch is defined then the function value at the end epoch applies for all times after the end epoch. That is:

  $f_1(t) = f_r(t_s)$ if t_s is defined and $t < t_s$
  $f_1(t) = f_r(t_e)$ if t_e is defined and $t > t_e$
  $f_1(t) = f_r(t)$ otherwise

- If the function reference epoch t_0 is defined, then a constant is added to f_1 so that it evaluates to zero at the function reference epoch. That is:

  $f_2(t) = f_1(t) - f_1(t_0)$ if t_0 is defined
  $f_2(t) = f_1(t)$ otherwise

- If the scale factor s is defined, then the function is multiplied by it:

  $f(t) = s.f_2(t)$ if s is defined
  $f(t) = f_2(t)$ otherwise.

In these formulae all epochs *t* are defined as a decimal year in the Proleptic Gregorian calendar.

A date/time *yyyy-mm-dd*T*HH*:*MM*:*SS* is converted to a decimal year 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). An implementation is considered compliant whether or not it accounts for leap seconds.

Future versions of this specification may add new base time functions.

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. 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 algorithm for applying a displacement to a coordinate depends on the units of the displacement and the source and target CRSs.

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

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

If the source and target CRSs 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 latitude and longitude coordinates. The conversion from meters to degrees requires the ellipsoid parameters of the geographic CRS.

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

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

Note that these formulae does not account for the height of a point above the ellipsoidal surface. The deformation model is assumed to represent deformation on the ellipsoidal surface, so the actual east and north offset of a point above or below this surface will be slightly different to that defined in the model.

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

6.5. Iterative calculation of inverse transformation

If the interpolation CRS is derived from the target CRS in a transformation, then the coordinate of a transformation point is not known until after the displacement has been applied to the source CRS coordinate to obtain the target CRS coordinate. In this case an iterative calculation is required starting with an approximation for the interpolation CRS coordinate, and iteratively refining this approximation by calculating the target CRS coordinate.

The iterative calculation uses the following steps:

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

Note that at the edge of the model it may not be possible to calculate the inverse transformation. The model is undefined outside its spatial extent in the interpolation CRS. If the transformation of a point near the edge of the model moves it to a location outside that extent, then the first step of the iterative calculation will fail. This step uses the transformed coordinate as a first estimate for the untransformed coordinate. However, that will be a point outside the model, so calculating the transformation at that point is not possible. A more sophisticated algorithm could address this, for example using the nearest point within the spatial extent of the model at the first iteration.

6.6. Calculation of deformation between two epochs

The displacements de, dn and du used to transform a coordinate between two epochs are calculated by replacing the time function values f_i used in Clause 6.3 with the difference between the time function for the two epochs. That is, to calculate the displacement from epoch t_0 to epoch t_1 the time function for the i^{th} element is calculated as:

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

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

A

ANNEX A (INFORMATIVE) NOTES FOR DEFORMATION MODEL PRODUCERS

A

ANNEX A (INFORMATIVE) NOTES FOR DEFORMATION MODEL PRODUCERS

A.1. Decomposition into elements

This specification assumes that the deformation can be partitioned into one or more elements each of which is characterized by a time function. There are many other ways deformation can be modeled. For example, in geophysical research it is often modeled using uniform dislocation on rectangular fault surfaces in a uniform elastic half space. While such models may be used to research and develop deformation models, common practice in publishing them for coordinate operations is to convert the model to a set of grids defining the surface displacement.

Deformation is caused by a number of geophysical phenomena, each of which may be represented by 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 following the earthquake may be represented by several elements depending on the complexity of the deformation and the quality of information available. Typically, these will use exponential and logarithmic time functions. Several versions of a deformation model may be published after an earthquake as the ongoing postseismic deformation evolves and as measurements and modeling of deformation are refined.

Alternatively, the deformation model may be based directly on observations of displacement rather than modelling from geophysical phenomena. For example, in Japan the dense network of continuous GNSS (cGNSS) 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 in this specification are intended for use in coordinate operations — either to define time dependent transformations between different CRSs or to track the motion of objects between different epochs within a single CRS.

Generally, the user expectation of such operations is that they are invertible. If a coordinate is transformed from the source CRS to the target CRS, and then the resulting coordinate is transformed back to the source CRS, the final coordinate will be the same as the original coordinate. Similarly, if the model is used as a point motion model, transforming a coordinate from one epoch to another, and then transforming the resulting coordinate back to the original epoch, will result in the original coordinate.

To support this expectation, this specification requires that the spatial function of each element of the deformation model is continuous. This does not guarantee that the resulting coordinate operation is invertible. It is possible to define deformation models that are continuous but are not invertible. However, these are not realistic models. Ensuring that a spatial function is continuous is much easier than ensuring that the resulting coordinate operation is invertible. Continuity is used as a practical substitute for invertibility.

Within a single grid the spatial function must be continuous — that is ensured by the use of bilinear interpolation.

If a spatial function uses more than one grid, the requirement for continuity places two constraints on deformation model producers.

1. Where an element does not cover 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 of the element should be set to zero, even though the actual displacements will be slightly different from zero.

2. Where an element is represented by multiple overlapping or nested grids there will be transitions across which the selected grid changes from one grid to another. At these transitions the values interpolated from each grid must be the same. For example, the displacement at the edge of a child grid will generally have to exactly match that of its parent grid. However, if the edge abuts the edge of another child grid, then its 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 does not purport to model deformation for scientific use.

The challenge for model producers is to decide the extent to which the deformation can be usefully modeled 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 not to 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 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 to specify a value with a large uncertainty. This may be preferable to ensure a continuous, interpolable 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 gray 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.

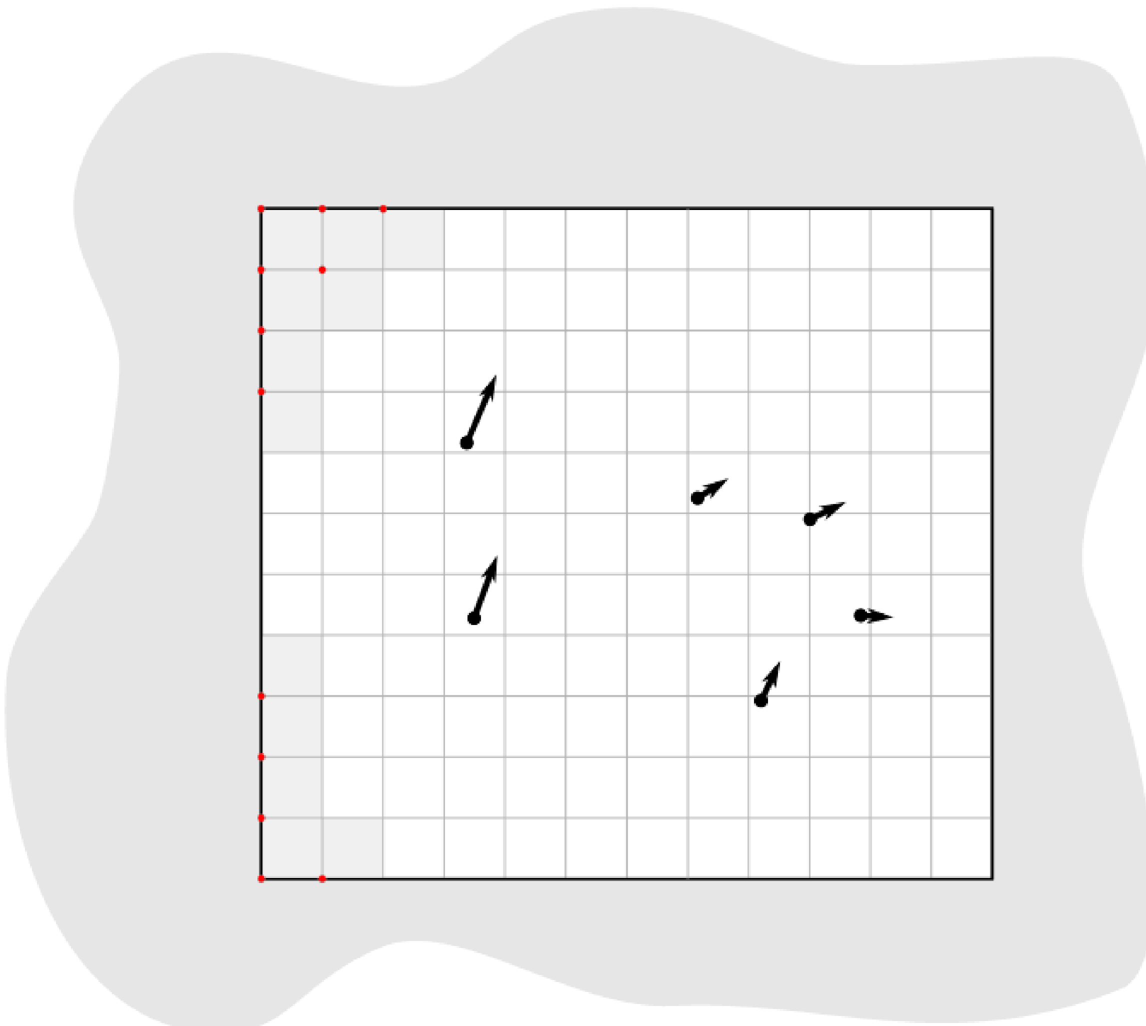

Figure A.1 — Influence of "no data" values

The gray 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. Alternative interpolation methods

This specification only supports the bilinear interpolation. A producer may choose to use, and an implementation may support, other methods, such as bicubic or biquadratic interpolation. However they are not defined in this specification. The calculation of displacements by a specific implementation may differ from that expected by the producer.

The reasons the specification is restricted to bilinear interpolation are: * there are few implementation choices, and hence less ambiguity, in using bilinear interpolation, and * it is easier to ensure continuity in a nested grid structure if bilinear interpolation is used.

There are few implementation choices because calculating the displacement at a point only uses the values from the four nodes of the grid cell in which the point lies.

The more sophisticated bilinear and bicubic interpolation use values from the nodes of cells adjacent to the grid cell in which the calculation point lies. Where the point lies in a cell on the edge of the grid some of these adjacent cells do not exist. There is more than one way that implementation can be written handle this situation, and there is no “right way” to do it. Different choices may give different results.

A.5. Deformation models near poles

The formulae defined in this specification are not suitable for transforming geographic (latitude and longitude) coordinates in near polar regions. Both the means of interpolating within a grid cell, and the formulae for applying the displacement to a coordinate, may yield unintended results as described below. The simplest way to represent deformation in such cases is to use a suitable projected (easting and northing) CRS. Future versions of this specification may offer better support for geographic CRSs in polar regions by offering alternative methods for interpolating displacements and adding displacements to coordinates.

Near the poles, grid cells may span a large longitude range. Consequently, the east and north components of a vector at adjacent grid nodes may lie in quite different directions, as illustrated in Figure A.2. Interpolating east and north components independently as described in Clause 6.1.2 fails to account for this. For example, if the grid cell spans 1° of longitude and the displacement is 1m then 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). This can be mitigated by using a finer longitude grid spacing.

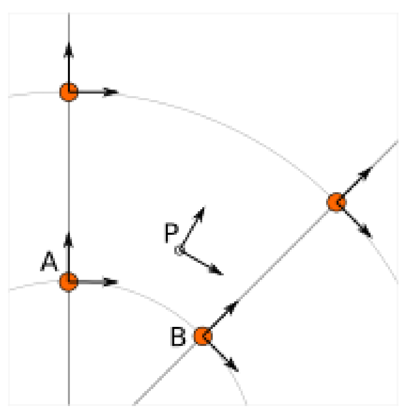

Figure A.2 — Different directions of east and north components at grid nodes and a calculation point

Where the source and target CRSs are geographic, then adding east north offsets to the longitude may not be appropriate. This is illustrated in Figure A.3 where the gray 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, applying an east displacement by changing the longitude coordinate, as defined by the formulae in Clause 6.4, may give a significantly different result than displacing the coordinate in the direction of the east vector. At the pole itself, changing the longitude coordinate makes no difference to the location at all.

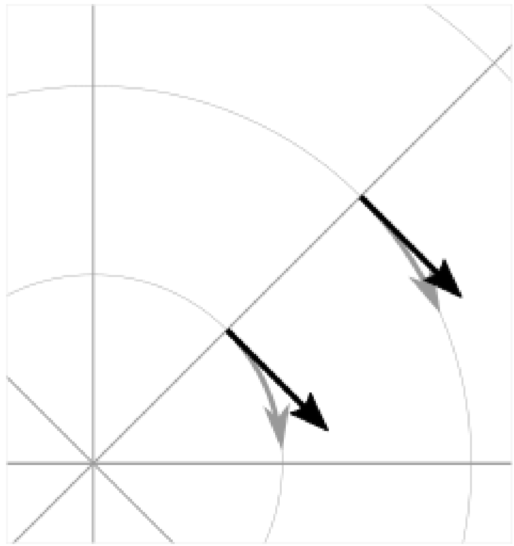

Figure A.3 — Comparison of vector and angular displacement near a pole

This is only a significant issue very close to a geographic pole. It diminishes rapidly away from the pole. For displacement d at a point at distance R from the pole, 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 changing the longitude would incur an error of only 5.10^{-7}m.

A.6. Time function epochs

The time functions used for deformation model elements can be modified by specifying a function reference epoch t_0, a start epoch t_s, and an end epoch t_e. The effect of these is illustrated in Figure A.4 showing a base hyperbolic tangent function without these parameters (graph A) and the same function with each parameter applied in turn. The base function has an event epoch t_v = 2013.8 and a time constant τ = 0.5 years.

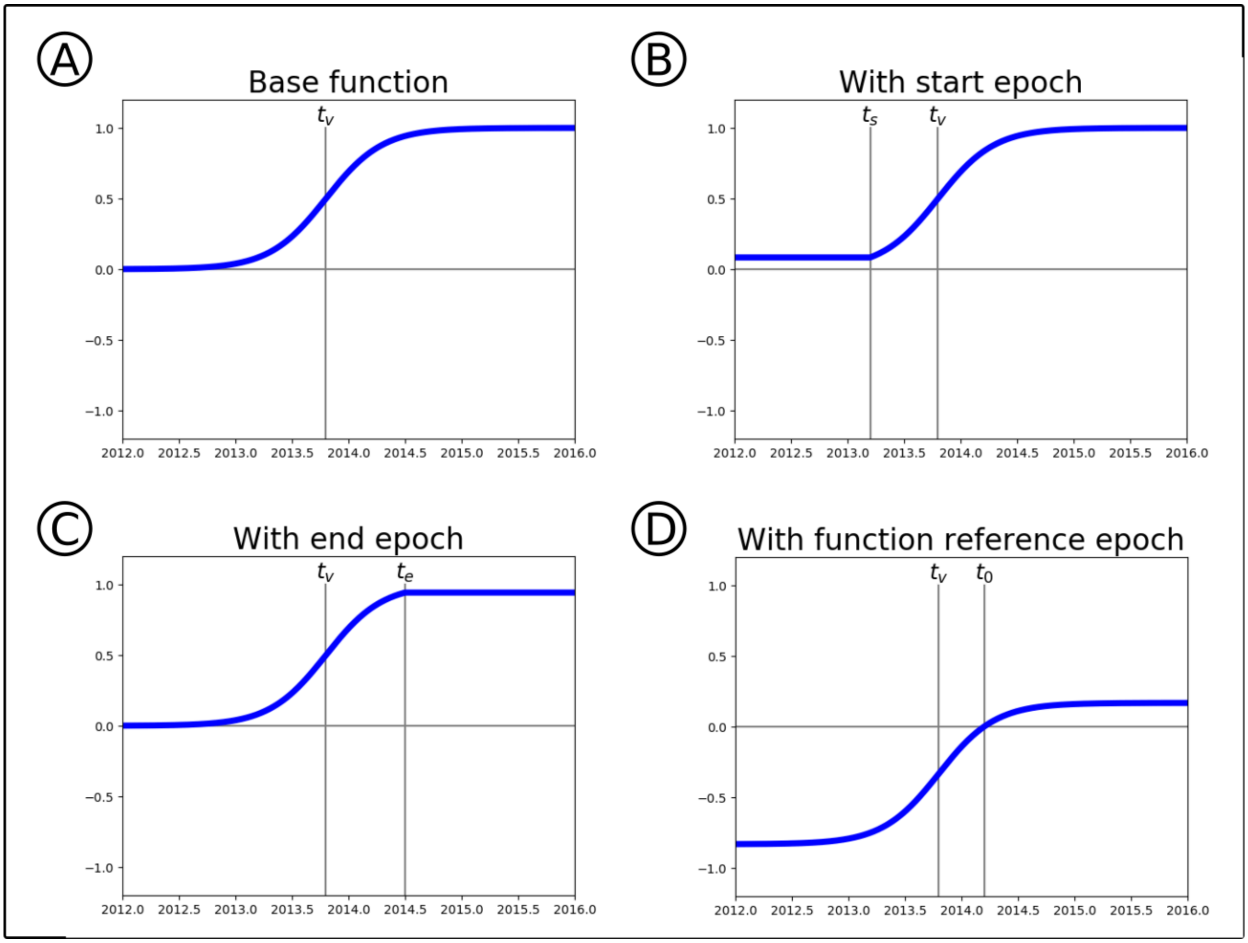

Figure A.4 — The effects of adding a start epoch (t_s), end epoch (t_e), and function reference epoch (t_0) to a base hyperbolic tangent function

The base function shows an event building in magnitude from 0.0 to 1.0. This might represent, for example, a slow slip event. However, it is never exactly 0 or 1, even though it appears to be in the graph. It approaches 0 exponentially as the epoch moves further into the past, and approaches 1 asymptotically as the epoch moves further into the future. The rate of change becomes infinitesimally small more than a few years from the event date.

The start and end epochs can be used to remove any change before the start epoch or after the end epoch. Setting a start epoch (graph B) applies the base function value at that epoch to any time before the start epoch. Similarly, setting an end epoch (graph C) applies the base function value at that epoch to any time after the end epoch. A consequence of this is that the start and end values are no longer 0 and 1. Generally a deformation model producer would use a function reference epoch as well as a start epoch, as described below, so that the time function is zero before the start date.

Using start and end epochs has practical value for deformation model producers managing a CRS. In particular, using the start epoch means that displacements calculated before that epoch do not change. This may be more realistic. Although the mathematical model does not define it, there is a time at which the slow slip event starts, and before which it is causing no deformation.

The function reference epoch t_0 is an epoch at which the function is forced to be zero, and therefore when the calculated displacement for the event will be zero. A constant offset is added to the function to force it to be zero at this epoch. In graph D the value -0.8 is added to the time functions so that it is zero at the function reference epoch 2014.2.

In a more realistic example, the start epoch and function reference epoch would both be used. In graph B, for example, the producer could set the function reference epoch to the same value as the start epoch — 2013.2. This would ensure that the time function is exactly zero for any epoch before this.

One usage of the function reference epoch is to describe deformation that occurred before the reference epoch of a datum. For example, an earthquake occurring before the datum epoch may be modeled by a step function with a function reference epoch set to the datum epoch. This is illustrated in Figure A.5 which shows the modified step time function where the event epoch is 2014.5 and the datum epoch is 2015.2. As the datum is referenced to 2015.2 the earthquake has no effect on coordinates observed at or after that date. However, to calculate the location of an object before the earthquake, the displacements caused by the earthquake must be subtracted from the object coordinates. By setting the function reference epoch to 2015.2, the step function becomes -1.0 for dates before the earthquake, and 0.0 for dates after it — a “reverse step” function.

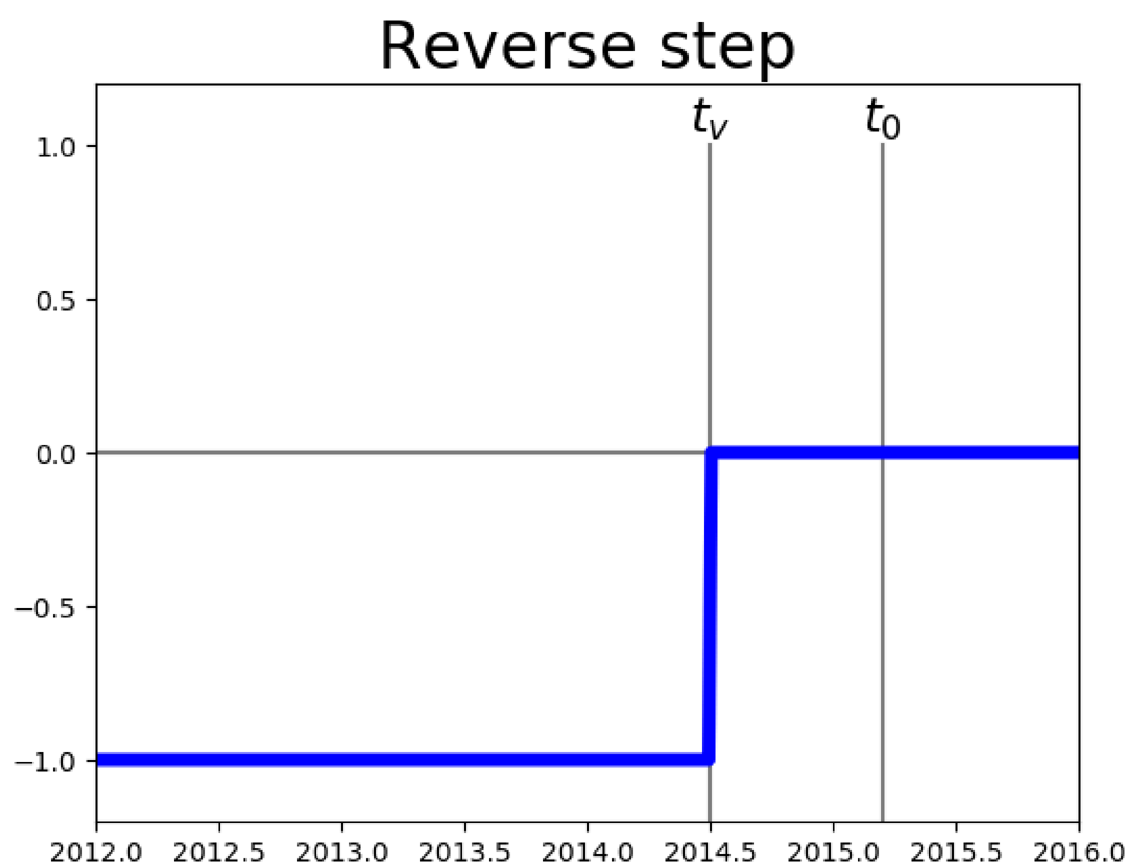

Figure A.5 — A "reverse step" function modeling an earthquake occurring before the datum reference epoch (defined by the function reference epoch t_0)

B

ANNEX B (INFORMATIVE) FUTURE DEVELOPMENT

B

ANNEX B (INFORMATIVE) FUTURE DEVELOPMENT

In the terminology of agile development this specification is a “minimal viable product”. It is intended to be suitable for the majority of current and proposed deformation models. However, during the course of its development a number of potential future enhancements were identified. Some of these are discussed below.

B.1. Discontinuities in the displacement field

One difficulty with representing deformation using gridded models is that they cannot represent discontinuities in the displacement field such as occurs in areas of surface faulting. There may be very large relative coseismic movement across a fault trace which may traverse one or more grids in the model. There may be breaks between segments of a fault trace between which the ground is distorted in unpredictable and often unmeasurable ways.

These displacements cannot be represented in a gridded model in which displacements are interpolated smoothly across each grid cell. This is illustrated in Figure B.1 showing a modeled horizontal displacement field. The left-hand diagram shows the actual displacement in two grid cells. A right lateral strike slip fault runs through the upper grid cell, shown by the gray line. The right-hand diagram shows the displacement interpolated from the values at the grid nodes (red vectors). In the cell containing the fault the interpolated displacement is very different from the actual displacement, whereas in the adjacent cell they are virtually indistinguishable.

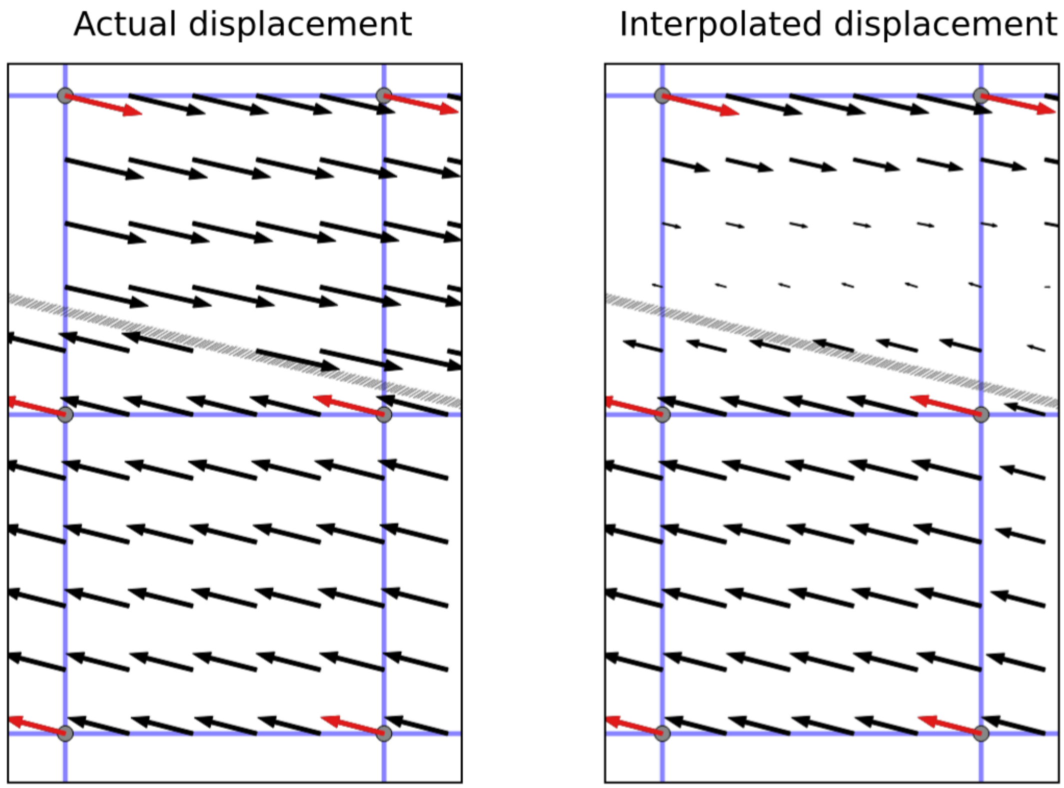

Figure B.1 — Ground displacement (left diagram) and interpolated displacement (right diagram) in a grid spanning a fault trace (hashed line).

Even if the deformation model were perfect, it might not be useful for transforming geographic features near a fault trace. Following an earthquake there are many different ways that spatial features may be changed — often not reflecting the exact deformation of the earth's surface. For example, the line of a cadastral boundary may be defined by legal considerations which override the physical movement of the ground. A road crossing a fault trace may be physically realigned to remain usable. These considerations cannot be accounted for by the deformation model. In many cases data must be recaptured in such areas to restore spatial integrity.

For the users of deformation models, it could be valuable to know when a transformation is likely to be affected by large errors which cannot be represented in the model. Conceptually, software would generate a warning to users. For example, this would happen when transforming a feature spanning a fault trace and over a time range that includes the time of the earthquake.

Four mechanisms have been suggested that could either alert users to such a situation when they do conversions, or allow them to investigate when it could affect their work. These are:

1. add metadata that defines the affected location and time. This could be a multipolygon definition in a standard spatial format, such as well-known text, associated with a date or epoch attribute. It could be either included in the deformation model metadata, or as an online resource referenced in the

metadata. This specification does not describe any standard for presenting such metadata. It would be up to the user to identify and interpret it.

2. use a large uncertainty to indicate where there is likely to be a large difference between the interpolated and actual displacements. While this can be used to indicate the approximate area affected by discontinuity, it is spatially imprecise. A large uncertainty value at a node affects all four grid cells surrounding the node. This differs from the interpolation error, shown in Figure B.1, which affects only the grid cells containing the fault trace — adjacent cells are not affected. Nonetheless it is an option available to producers, provided it is understood that it may overestimate the error in the vicinity of a fault trace.

3. analyze the displacement vectors at the grid nodes. Grid cells likely to be affected by large interpolation errors could be identified by analyzing the displacement vectors themselves. A simple measure of this uncertainty in a grid cell is the maximum vector difference between any two of the displacement vectors at the four nodes of the cell. For a grid cell spanning the fault trace, the displacement vectors for nodes on one side of the fault are in the opposite direction to those of nodes on the other side of the fault so there is a large difference between these vectors. Conversely, for a grid cell to one side of the fault trace, the four displacement vectors have approximately the same direction and magnitude, so the maximum difference between any two of these vectors is small. This method requires special purpose software to analyze the grids and determine the affected grid cells. The software would then have to alert users when these cells were used in a transformation. A benefit of this approach is that it provides an indication of the magnitude of the potential error.

4. use an additional grid node attribute — a “flag” value that specifically identifies affected cells. This is a boolean value that is true or false. There are several options for how this could identify the affected cell. One option is to use a convention defining which node identifies a grid cell. For example, the node on row i and column j could identify the cell between grid nodes on rows i and i+1 and columns j and j+1. Another option is to set the flag as true on all nodes of an affected grid cell. A grid cell is identified as affected only if the flag value is true on all four of its nodes. Such an attribute would require special processing by software.

One difficulty with all these mechanisms is that coordinate operations are “black box” operations to most users — they often happen within software applications as data from different CRSs or different epochs are combined for analysis and presentation. Often this will be done on remote servers presenting information to users in web browsers. Users may not be aware that a coordinate operation is happening. The mechanisms above would need extensive implementation in the spatial software stack for a warning generated in a coordinate transformation to be presented to users in a meaningful way. In light of this practical difficulty, this document does not explicitly specify how discontinuity should be handled.

Although handling discontinuities is not defined in this specification, the first two of these options, using metadata and using uncertainty, are available to model producers already.

The third method, analyzing the displacement vectors, can be used with any deformation model. An approach like this could be developed in software for analyzing deformation models independently of coordinate operation software. This is an area that may warrant further research once more deformation models, and particularly models incorporating coseismic displacements, become available for analysis.

B.2. Vector interpolation in polar regions

As noted in Annex A.5, this specification does not recommend the use of geographic CRSs (i.e. latitude/longitude) for deformation models close to the geographic poles. In such cases, the horizontal displacement vector cannot be interpolated or applied correctly. Early versions of this specification did include an alternative interpolation method, and an alternative method of adding horizontal displacements to a geographic coordinate. In each case the method involved transforming the displacements to and from a geocentric Cartesian CRS.

These were removed from the specification for two reasons:

1. currently there is no requirement for this capability, and
2. it was felt that further research should be undertaken before recommending specific methods.