OGC® DOCUMENT: 21-999R1

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

OGC®
Making location count.

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. Intuitively fixed features such buildings are moving with the earth's crust — subject to ongoing plate tectonic movement and episodic deformation events such as earthquakes. This creates a challenge of the geospatial community. Increasily 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). However the coordinates from GNSS are reference to global coordinate systems and in this reference frame the ongoing movement of these 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 a "fixed" object's location to be propogated to different times.

Common uses for this include:

- determining the current location of an object based a historic measurement of its location,
- transformation 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

This specification defines a functional model for describing a deformation model. It describes a way of parameterising deformation such that it can be encoded into a data set and incorporated into software for using in coordinate transformations. It also defines how to calculate the displacement of a coordinate between two different times using this parameterisation.

This specification allows producers of deformation models, typically national geodetic agencies, to publish the deformation model in a consistent way such that users of coordinate transformation software can be confident that the deformation 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

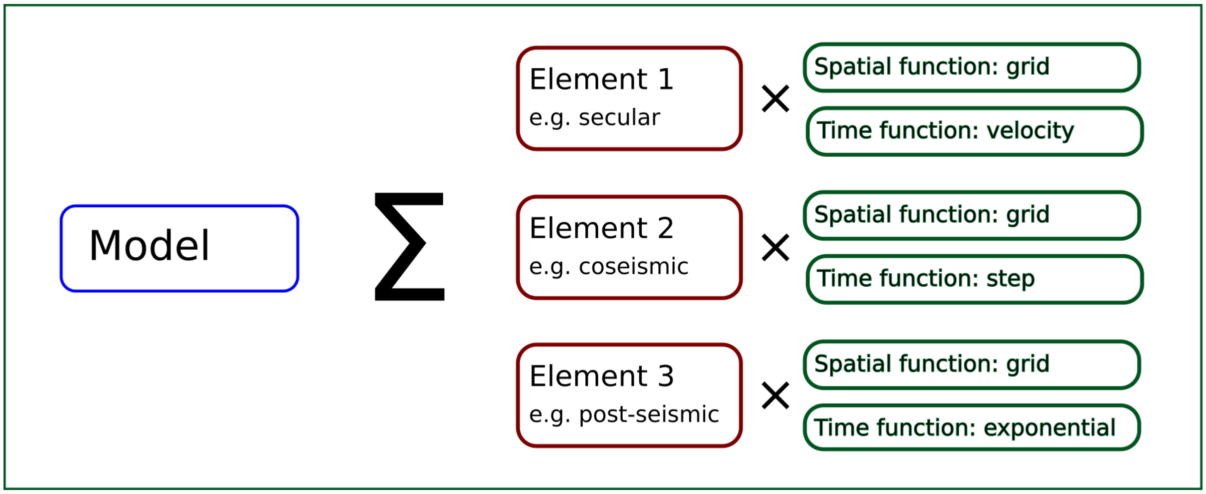
SUBMITTER	AFFILIATION
Chris Crook	Land Information New Zealand
Kevin Kelly	ESRI
…	…

VI INTRODUCTION

This specification has been developed by the OGC Coordinate Reference System (CRS) domain working group (DWG) in conjunction with the 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 parameterisation 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 will represent specific geophysical causes, such as coseismic deformation due to an earthquake. However an element could equally represent some measured for which no physical cause has been ascribed. Equally one event may be represented by more than one element. For example the horizontal deformation due to an earthquake may be represented by one element, and the vertical deforamtion 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.

This specification also does not describe how the 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 in positioning systems. The purpose of this specification is to define what parameters must be encoded in a compliant deformation model, and what formulae compliant software should use to transform coordinates using the model. This compliance will ensure that transformations carried out by users give the results intended by the model producers.

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 calculates a one, two, or three dimensional coordinate displacement. It may also calculate 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 supports. It also defines metadata attributes that should accompany the encoded model to allow users to understand its applicability and limitations.

2

CONFORMANCE

2 CONFORMANCE

<Insert conformance content here>

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 characterised 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 deformation at any time and position is calculated by summing the contribution from each element 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 a nested grid structure comprising one or more grids.

   The grids must conform to the following requirements:

   - Grids rows are spaced equally spaced
   - Grid columns equally spaced.
   - Grid rows and columns are aligned with the axes of the definition CRS
   - Grids may share a common edge but otherwise may not intersect unless one of the grids is fully contained by the other grid.
   - Grid cells of the contained grid must each be fully contained in a grid cell of the containing grid.

   To evaluate the spatial function at a specific position the nesting algorithm identifies a single grid is applicable at that position. The spatial function displacement components are calculated from just that grid using the formulae. This calculation 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 defined different componenents. E.g. one function might define horizontal displacement and another may define vertical displacement.

g) Displacement components of the spatial function may include:

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

Displacements 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
  - horizontal covariance matrix components c_{ee} , c_{en}, c_{nn}.
- vertical uncertainty e_v
- horizontal and vertical uncertainty
- covariance of horizontal and vertical displacement components c_{ee} , c_{en}, c_{nn}, c_{eu}, c_{nu}, c_{uu}

The components used to represent uncertainty are defined in the model metadata and must be the same for all spatial functions which have uncertainty components.

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

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

i) A spatial function may include a quality parameter at each node providing guidance on the reliability of the spatial function in the vicinity of the node. For example, a quality parameter could indicate surface faulting affecting cells adjacent to the node.

j) 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 and thus can be assumed by software implementations of the model.

k) 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 not 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.

l) The time function 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 ramp time function defined by a start and end time and a start and end scale factor. Two specialisations of the ramp time function are the step and reverse step functions.
- 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.

m) 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 or before it in an inverse transformation as described in Clause 6.6.

n) Each element definition includes metadata defining:

- The type of spatial function (grid)
- The spatial interpolation method to use
- The quantities it defines (displacements, uncertainties)
- 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 horizontal and vertical uncertainty which applies if the spatial function does not explicitly define uncertainty.
- Other metadata required by the implementation
- Other producer metadata

o) The model definition includes metadata defining:

- The version of the this specification with which the model complies
- The name of the model
- The version of the model
- The publication date
- The licence under which the model is published
- A description of the model
- Contact information for the agency publishing the model
- 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 units of horizontal displacement
- The units of vertical displacement
- The default horizontal and vertical uncertainty for each element of the model
- The parameters representing uncertainty at each spatial function node, e.g. horizontal covariance, vertical uncertainty
- the probability level of uncertainties in the model, e.g. 95% confidence level
- The spatial extent of the model
- The time extent of the model
- The algorithm used to apply the calculated displacement to an input coordinate.

It may also include:

- Links to reference information about the model
- Other metadata required by the implementation
- Other producer metadata

6

CALCULATION FORMULAE

6 CALCULATION FORMULAE

The formulae below define the expected behaviour of software implementations of the deformation model. Whatever algorithm a software implementation may use it should produce numerically indistinguisable 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 nested grids. At a given position only one of the grids is used to calculate the spatial function displacement or uncertainty. The selected grid is the smallest grid by area that contains the position. The strictly nested arrangement of grids ensures that this is uniquely defined at any position. If no grid in the spatial model contains the position 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

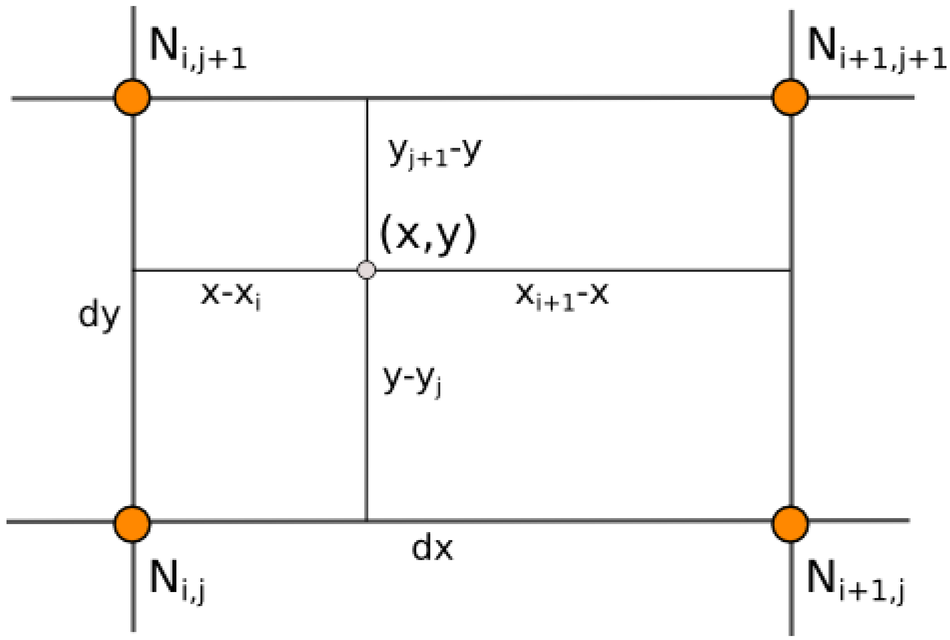
$x_i = x_o + i.x_s$
$y_j = y_o + j.y_s$

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 were 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).

Displacement vector elements 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 displacement components de, dn, dh at (x,y) are calculated as the weighted average of the values at the nodes as, e.g.:

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

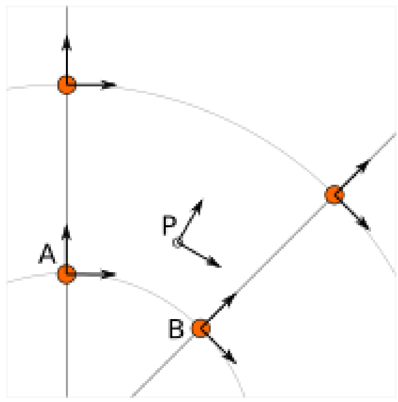
The uncertainties eh, ev at (x,y) are calculated from the values at the nodes as, for example

$eh = \sqrt{(W_{i,j}*eh_{i,j}^2 + W_{i+1,j}*eh_{i+1,j}^2 + W_{i,j+1}*eh_{i,j+1}^2 + W_{i+1,j+1}*eh_{i+1,j+1}^2)}$

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 reparameterising the vectors to a common set of axes before forming the weighted average.

The geocentric blinear 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 parameterised as a function of decimal years. For example, velocities are expressed metres/year.

The conversion of a UTC epoch *yyyy-mm-dd*T*HH*:*MM*:*SS*Z 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 epoch by the total number of seconds in the year (ie 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.

Table 1 — Time functions

TIME FUNCTION TYPE	PARAMETERS	FORMULA ($T_{I,MIN} \le T < T_{I,MAX}$)
velocity	Function reference epoch t_0	$f(t) = t - t_0$
	Start epoch t_s	$f(t) = f(t_s)$ for $t < t_s$ if t_s is defined
	End epoch t_e	$f(t) = f(t_e)$ for $t < t_e$ if t_e is defined
step	Step epoch t_s	$f(t) = 0$ when $t < t_s$,
		$f(t) = 1$ when $t \ge t_s$
reverse step	Step epoch t_s	$f(t) = -1$ when $t < t_s$,

TIME FUNCTION TYPE	PARAMETERS	FORMULA ($T_{I,MIN} \le T < T_{I,MAX}$)
		$f(t) = 0$ when $t \ge t_s$
ramp	Start epoch t_s Start scale factor f_s End epoch t_e End scale factor f_e	$f(t) = f_s$ for $t < t_s$ $f(t) = (f_s.(t_e - t) + f_e.(t-t_s))/(t_e-t_s)$ for $t_s \le t < t_e$ $f(t) = f_e$ for $t \ge t_e$
exponential	Start epoch t_s Start amplitude α_0 Final amplitude α_∞ Decay constant θ *End epoch* t_e	$f(t) = 0$ for $t < t_s$ $f(t) = \alpha_0 + (\alpha_\infty-\alpha_0).(1 - \exp(-(t-t_s)/\theta))$ for $t \ge t_s$ $f(t) = f(t_e)$ for $t > t_e$ if t_e is defined
logarithmic	Start epoch t_s Scale factor α Time constant τ *End epoch* t_e	$f(t) = 0$ for $t < t_s$ $f(t) = \alpha.\ln(1 + (t-t_s)/\tau)$ 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 Acceleration a *Start epoch* t_s *End epoch* t_e	$f(t)= a.(t - t_0)^2$ $f(t) = f(t_s)$ 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 α	$f(t)= 0$ for $t < t_s$ $f(t) = \alpha/2 + A.\tanh((t - t_r)/\tau)$ for $t \ge t_s$ and $t < t_e$ $f(t) = \alpha$ for $t \ge t_e$ where: $t_r = (t_s + t_e)/2$ $A = \alpha/2.\tanh(t_e-t_r)$ $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$
cyclic	Frequency *f* (cycles per year) Function reference epoch t_0 Cosine amplitude α Sine amplitude 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 specialisations 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 deformation 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. See 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. 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 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 the method described here does not apply close to the poles. See the section below "calculation horizontal deformation near the poles" for details.

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

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

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

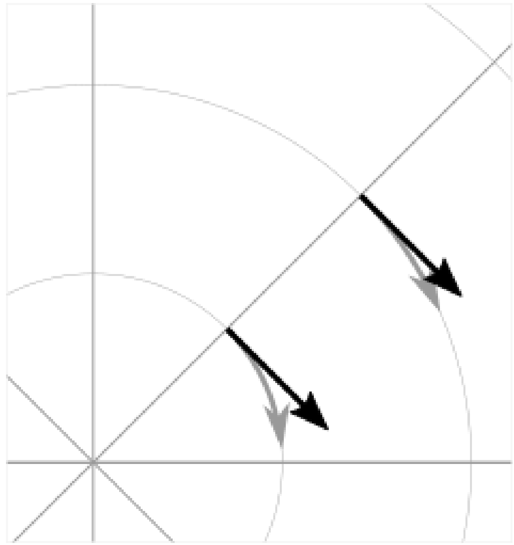
The vertical displacement is always in metres and is simply added to the height coordinate.

6.4.2. Geocentric addition method

The geocentric method can be applied if the spatial function uses a geographic coordinate system and displacements are given in metres. In this case the horizontal displacements are first converted to geocentric (XYZ) displacements, then added to the geocentric coordinates, and finally converted back to geographic coordinates. The vertical coordinate is always calculated by simple addition of the vertical displacement to the vertical coordinate.

This method may be applicable for points near the pole, where simple addition of displacements to the geographic coordinates may not give the desired result. This is illustrated in Figure 4 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 applying a displacement in the direction of the east vector gives a different result to hanging the east (longitude) coordinate.

Figure 4 — 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 $d^3/2R^2$. or example, a 1 m east displacement 10 km from the pole would have an error of 10^{-8}m, but increases rapidly as one approaches closer to the pole.

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

Standard formulae are used to convert geographic coordinates to and from geocentric 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.

The horizontal components of displacement are converted to Cartesian (X,Y,Z) components using the same formulae as described for the geocentric bilinear method in Sec. 6.1.3.

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:

- using the input coordinate as an initial estimate for the output coordinate
- at each iteration:
  - use the current estimate of the output coordinate to determine that 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 less than the precision required for the inverse operation then 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*

A

ANNEX A (INFORMATIVE) EXPLANATIONS AND EXPANDED DISCUSSION POINTS

A

ANNEX A (INFORMATIVE) EXPLANATIONS AND EXPANDED DISCUSSION POINTS

Note: This section is likely not appropriate for the final abstract specification

A.1. DECOMPOSITION INTO COMPONENTS

This specification assumes that the deformation can be decomposed into a set of spatial functions each multiplied by a scalar time function. This is suitable for many geophysical phenomena such as secular motion (velocity models) and coseismic ground deformation.

It may be less suitable to deformation with a complex time evolution such as slow slip events propogating along a fault system, or post-seismic deformation. However currently deformation models for coordinate operations are all represented in this way. Decomposing in this way can represent any deformation to an arbitrary level of detail, but it may not be the most efficient way to do so.

A.2. SPATIAL MODEL TYPES

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 include this capability in the initial specificiation.

In New Zealand triangulated models were considered for modelling the deformation due to the 2011/12 Christchurch earthquakes but did not offer much advantage in the size of the model, and also are much less efficient to evaluate since it is necessary to search the triangulation to determine which triangle applies at a location. (See https://www.linz.govt.nz/system/files_force/media/file-attachments/winefield-crook-beavan-application-localised-deformation-model-after-earthquake.pdf?download=1).

The project team is not aware of any triangulated deformation models either in use or proposed for coordinate operations. At least one datum transformation is defined this way (Finland EPSG:2393 (KKJ / Finland Uniform Coordinate System) projected CRS to EPSG:3067 (ETRS89 /

TM35FIN(E,N))). This is a similar type of transformation, and indicates that there may be an appetite for triangulated deformation models in the future.

Other potential models may also be considered in the future.

The project team briefly considered including global plate motion models in the specification.

Plate motion models are defined by a set of polygons defining the extents of tectonic plates. Each plate has a defined rigid body rotation around the centre of the earth is defined. Typically these are used in GNSS analysis to predict the movement of survey marks. They are discontinuous at the boundaries between plates.

While these are a form of deformation model they are considered out of scope for this work. The project team concluded that these models have a very different from the regional deformation models under consideration and there is little value in trying to including them in a common specification.

It may be worth considering a standard for plate motion models as a separate exercise in the future.

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 as a mechanism for describing a global deformation model.

A.3. NO DATA AND ZERO VALUES

Most deformation models only cover a limited part of the globe. They are limited by jurisdictions area of authority or by lack of data — for example areas of sea where there is no measure of deformation. Also in areas affected by recent large deformation we may not have good data before the event. There are several approaches to both identifying and handling these areas where there of no or poor information.

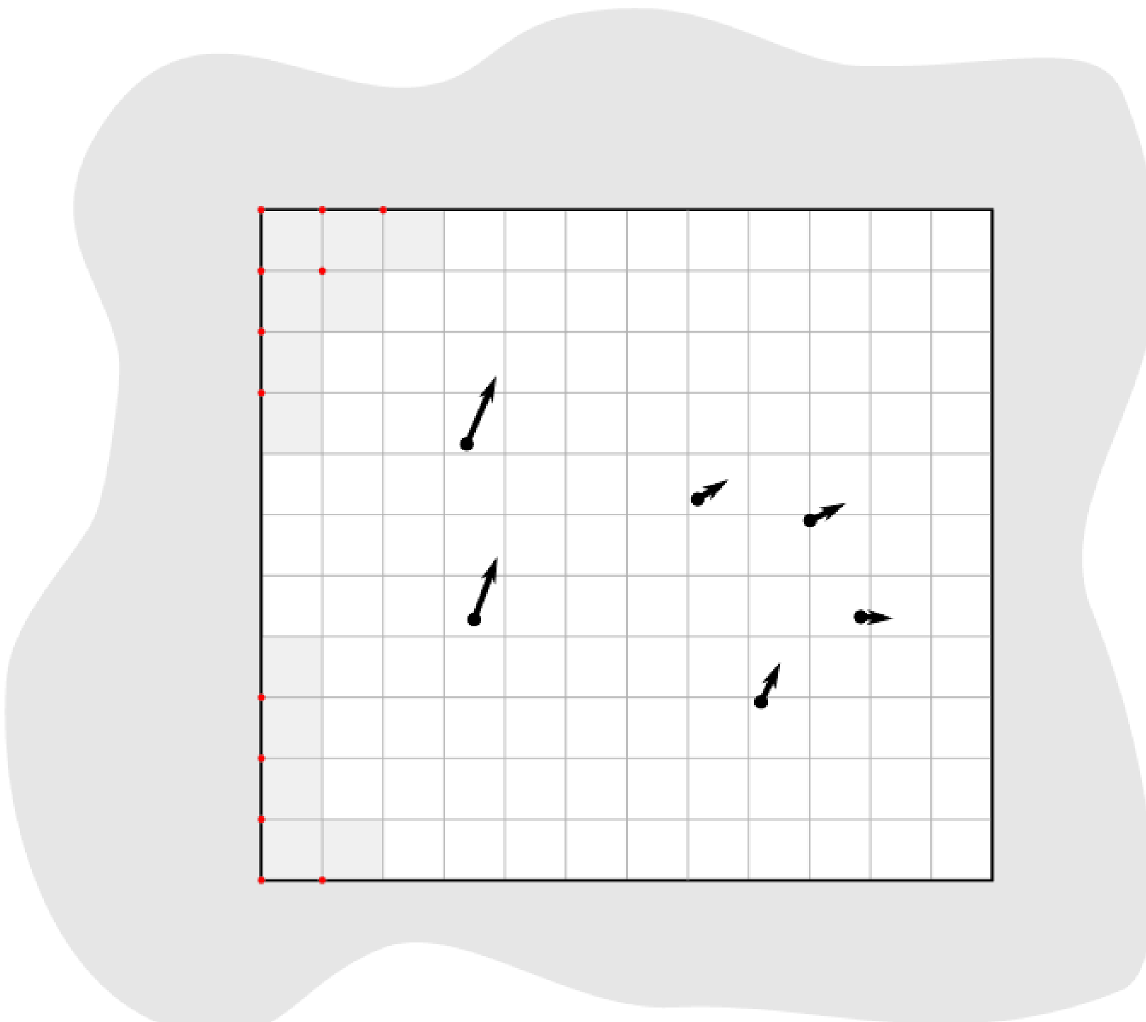
The areas where displacement is not defined could be defined by a complex geographical extent. However grids must be defined for rectangular regions in their coordinate system, which are unlikely to match the extent of the model. From an implementor's point of view it is preferable that complex extents are identified by special grid values that identify points beyond the extent rather than a complex bounding shape.

This specification supports the notion of a "no data" value. Note that an alternative approach is to specify an unreal value with a large uncertainty. This is considered below in the discussion on continuity of the model. These "no-data" values could be identified in a number of ways, for example by a flag value on the grid nodes, by special values of displacement (eg 99999), or NaN (not a number) floating point values.

The "no data" value 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 deformation is unknown, and might be significant.

If evaluating the deformation model at a given location and time requires using a “no data” value then the displacement (or uncertainty) calculated at that location is undefined. This would typically results in an error message to users to this effect. The diagram below shows how this might look in a deformation model. In this diagram the square marks the total extent of the deformation model. Outside this area the deformation model cannot be evaluated.
In the deformation model is a gridded spatial model. Within there are a number of nodes at which the displacement is not defined (that is it has a “no-data” value). Where these nodes are required to calculate the displacement, which is any grid cell they are on the boundary of, the deformation model cannot be calculated. The grey area in this diagram shows the region in which the displacement is not defined by the model and cannot be calculated.

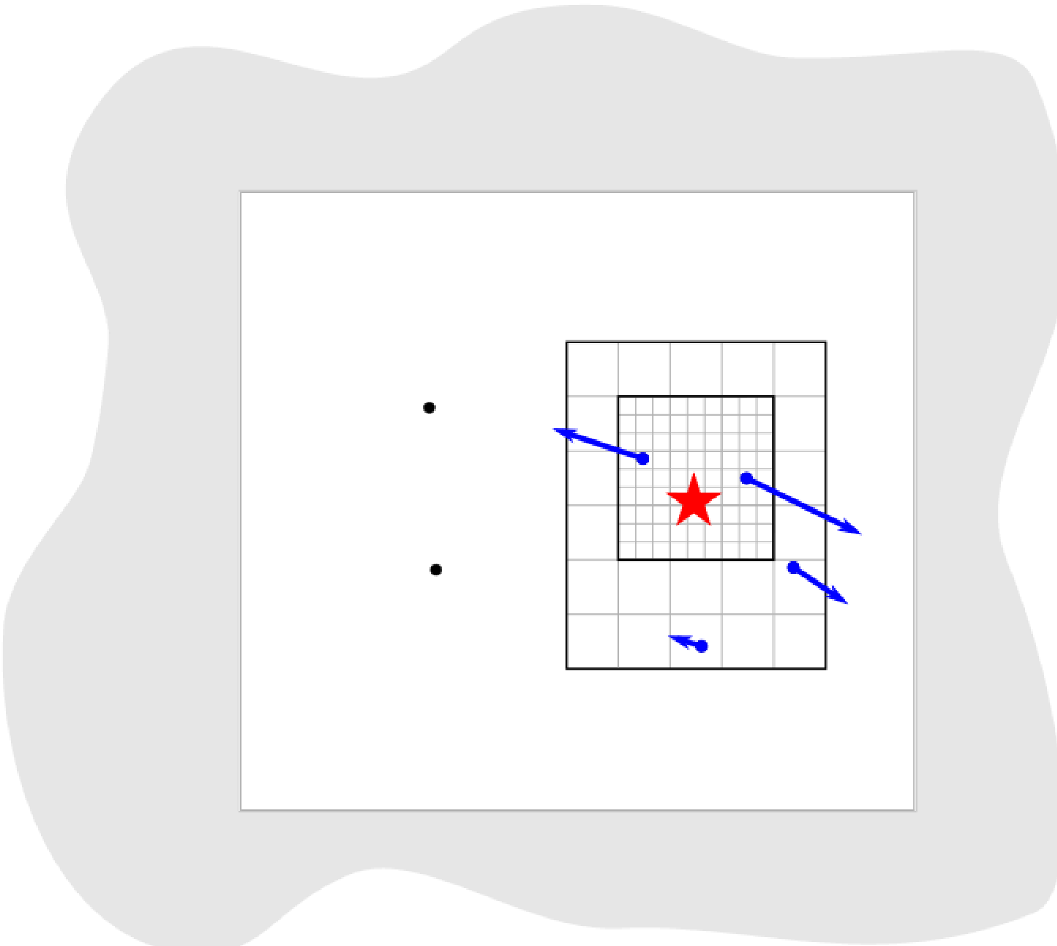
Figure A.1



This may occur where the area in which deformation is defined is an irregular shape. It might include coastal regions where the deformation of the seabed is not measured, or it may be that it crosses a jurisdictional boundary. As the gridded model is by definition a rectangular area it will include these regions in which the deformation is not known, which are correctly represented by a “no data” value.

Where a component only covers a subset of the total area of a deformation model it is assumed to have zero displacement beyond its extent. An example of this is a deformation model component representing deformation due to an earthquake. In the vicinity of the epicentre there may be extensive deformation. However there may also be large regions that lie within the extent of the deformation model but at which the deformation is zero or insignificant. The deformation component created to represent this only needs to include the area where there is significant deformation. This is shown in the figure below. In this diagram the outer square represents the total extent of the deformation model. Beyond this extent the deformation is undefined. The nested grid inside the total extents is used to represent the deformation due to an earthquake. In the region outside the nested grid component the deformation due to this component is zero.

Figure A.2



A.4. QUALITY PARAMETERS AT SPATIAL MODEL NODES

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 which where the deformation includes significant distortion or discontinuities that are not well represented by the deformation model.

The intention is that software could warn users when the coordinate conversion quality is compromised by such distortion.

This could be represented by a quality parameter the corner nodes of affected grid cells. Software could then assess the impact on an interpolated coordinate conversion by compiling the quality information from each of the nodes used in the interpolation.

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

- how should it be represented
- how should the measure be interpolated to provide a quality measure at an interpolated point
- how should the quality measure relate to the time (or times) of a conversion. If time function evaluates to zero it should clearly be ignored, but how large can the time function be before the quality parameter is considered significant.
- how does the quality parameter relate to uncertainty

An alternative methods of defining affected areas could be to include this in the model metadata. For example the component header could include one or more areas of concern, each with a spatial definition as a multipolygon and an event time.

A.5. REQUIREMENT FOR MODEL TO BE CONTINUOUS AND INVERTIBLE

The deformation model is required to be continuous and invertible within the spatial and temporal extent of the model except where it is not defined (ie “no data” value). This is a practical requirement on deformation models within the context of coordinate transformations.

This means that the deformation model cannot exactly represent the true deformation. For example where deformation is due to surface faulting the actual deformation may not be continuous across a fault line.

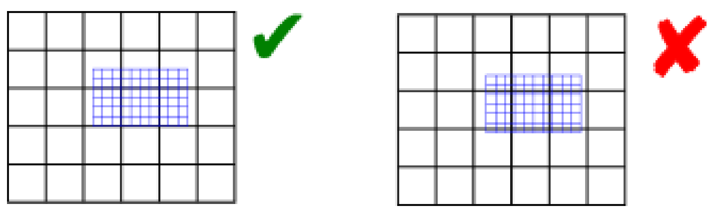
The actual deformation also may not be invertible (at least as a function of horizontal position only) in an area of thrust faulting where points originally on opposite sides of the fault may be moved to the same horizontal position (though at different heights).

However the purpose of this deformation model specification is not to exactly represent deformation, but to represent it to the extent that is useful within the context of coordinate transformations.

At least for the initial release of a functional model specification it is proposed to require a compliant model is continuous and invertible within the extent of the model. This simplifies implementations and avoids the need to specify the behaviour where the model is not continuous and invertible.

The requirement for continuity does have implications for how models are defined. For example it means that in nested grids child grids must be aligned with the parent grid as shown below to ensure continuity at the edge of the child grid. (Note that an alternative approach would be to define child grids as independent deformation components with the same time function which would model a perturbation from the simple parent grid — in this case there would be no requirement for the models to be aligned).

Figure A.3



A question for implementers is how to transform data that extends beyond the deformation model. 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 in the deformation model functional model.

- Require that valid models should have zero deformation at the boundary. 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 this data is not reliable.
- 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 extents of the deformation model is not permitted, and will result in an exception (or equivalently a no data value).

- Not specify a behaviour — implementors can choose if and how to transform data outside the extents of the model. Transformations beyond the extent of the model would be considered out of scope of this functional model specification.

From a producer's perspective the third of these — fail if data beyond the model is transformed — is most correct. Also producers may not be concerned about transformations beyond their jurisdication, so that any of the last three options could be acceptable. In any case it is beyond their control. 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, in practice most current software does not consume or report uncertainty information, so the user may be misled to 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 they 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. Until we have a global model there may be no good solution for this.

Also from a user's point of view it is desirable that different implementations give the same result — implementation specific behaviour is not desirable.

Currently this specification takes the producer's perspective — a transformation beyond the extents of the model should fail. However this is open to debate!

A.6. TIME FUNCTIONS

The proposed set of base time functions includes those commonly used in geophysical models, for example reference station coordinates in the International Terrestrial Reference Frame. However in practice there may be little benefit in complex time models, as it is unlikely that the same time function will apply at all points in the area affected by, for example, post-seismic deformation. That is to say that the actual time evolution at each point within the spatial model may have different attributes and parameterisation. The deformation model component is necessarily an simplification attempting to best fit 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.

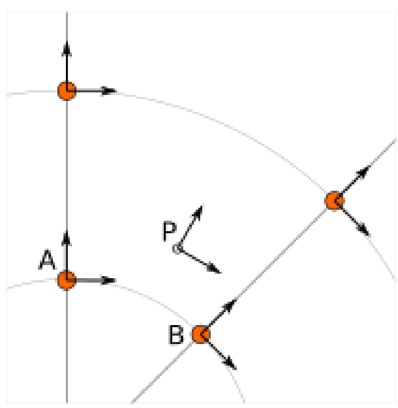
The ramp function is used for specific time bounded events. It more meaningfully represents specific deformation events as the spatial model displacements reflect the total deformation rather than the rate of deformation. Also it supports the step function specialisation which is not possible with a 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 components, each with its own spatial model and time function, rather than a complex time function applying to a single spatial model. For example each year there could be an updated gridded spatial model. 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 functional model by constructing time functions for each grid that define the interpolation between one grid and the next.

A.7. GEOCENTRIC INTERPOLATION NEAR POLES

The geocentric weighted average method proposed above is proposed for use in near polar regions where east and north topocentric vectors at adjacent grid nodes are in significantly different directions.

Figure A.4



To estimate the error that could be incurred using simple bilinear interpolation and not accounting for this difference we can consider a case where the deformation is 1m northwards at point A, and zero at point B in the diagram above. Let the longitude grid spacing be λ_S radians. If the calculation point P is λ radians past A, then the magnitude of the interpolated vector will be $(\lambda_S-\lambda)/\lambda_S$. The error of orientation will be λ radians (the difference between north at A and north at the calculation point). So the vector error will be $\sin(\lambda).(\lambda_S-\lambda)/\lambda_S$. Approximating $\sin(\lambda)$

as λ, this has a maximum absolute value in the range $(0,\lambda_S)$ of $\lambda_S/2$. So for example with a grid longitude spacing of 1° this could result in a 2cm error in the 1m of deformation vector.

A.8. SEQUENTIAL OR PARALLEL EVALUATION OF COMPONENTS

These formulae use the same input coordinate to calculate the deformation for each component.

An alternative approach that could be used is to apply components sequentially. That is the first component is calculated and applied to the coordinate, and then the modified coordinate is used to calculate the second component, and so on. This may result in a different final coordinate to the proposed method, as the second and subsequent components are evaluated at a different location.

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

If the components 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 in which the first component is a velocity function and the second is a step at 2003-01-01. If the deformation 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 that different coordinate is used to interpolate the step function model. 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 components is very unlikely to make a significant difference to the coordinates — at worst it is very similar to that described below for the inverse method in relation to iterating the inverse calculation or not. The choice of independent evaluation has some small advantages in calculation in that:

- using the same input coordinates is slightly more efficient as the calculated displacement only needs to be applied to the coordinate once. This could be a significant difference if the horizontal displacement is applied using the "geocentric" method as described below. It is insignificant if the displacement is applied by simple addition.
- using the same input coordinates for all components provides an opportunity for parallelising calculation of components.
- using the same input coordinates for each component allows optimising transformations between two versions of the deformation model as common components can be ignored.

A.9. SIGNIFICANCE OF ITERATION FOR THE INVERSE DEFORMATION MODEL EVALUATION

The error of not iterating the inverse transformation can be tested for the New Zealand NZGD2000 deformation model. The least smooth area of deformation in New Zealand is that affected by the 2016 Kaikoura earthquake. As this has been updated by “reverse patching” the inhomogeneity of the deformation field primarily affects pre-earthquake transformations. Testing across the fault zone finds that the maximum error from not iterating an inverse transformation of epoch 2000.0 coordinates is about 0.015 metres. However this is in an area where the deformation model is very inaccurate — it is smoothed across the fault zone and will have errors of many decimetres. For transforming epoch 2019.0 coordinates the maximum error is about 0.000014 metres. In the North Island in an area largely unaffected by episodic events the maximum error is about 0.00015 metres.

Based on this result it is recommended that the inverse transformation is iterated. It is likely that this will double computation time (it would be unusual to require more than two iterations).

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.

A.10. CALCULATION OF DEFORMATION BETWEEN TWO EPOCHS

The displacement de, dn and du to transform a coordinate between two epochs is simply the difference between the displacement values calculated at each epoch.

The uncertainties of these displacements require a more sophisticated calculation as uncertainty of displacement components calculated at different times from the same model are clearly correlated.

While there is no mathematically correct way to quantify the uncertainty without a more complete error model than defined in this deformation model representation, the following approach is suggested.

The time function error factor of the difference between t_0 and t_1 is calculated for each model component separately as $f_{e,t1-t0} = \sqrt{abs(f(t_1)-f(t_0))}$.

The eh and ev values from the spatial function for each displacement element are multiplied by these time function error factors and then combined as the root sum of squares to give the total uncertainty of the displacement between the two epochs.

A.11. CONVERSION OF COORDINATES BETWEEN VERSIONS OF THE DEFORMATION MODEL

A common source of confusion is coordinate transformations between different versions of a datum.

For example in New Zealand the deformation model was recently updated from version 20171201 to 20180701. Technically this is equivalent to a new version of the datum.

Users with a GIS datasetin terms of the 20171201 version of the datum might want to update the dataset to version 20180701. The user expectation is that this will generate correct version 20180701 coordinates of the features in the database.

The critical thing in this transformation is that the coordinate epoch for the transformation is before the event(s) implemented in the update. This is somewhat counter-intuitive.

Generally the update should not change the coordinates. The reason for the update is typically a deformation event such as an earthquake. The earthquake coseismic deformation is added to the deformation model as a step function that applies for transforming coordinates for epochs after the event. This means that the NZGD2000 coordinate system tracks the movement of features fixed to the ground and therefore the NZGD2000 coordinates of these features are not changed by the earthquake. In this case the deformation model is unchanged before the earthquake. Transforming at an epoch before the earthquake will leave the coordinates unchanged which is what is required..

Close to faulting the distortion due to the earthquake can be too intense to be included in the coordinates. In that case the deformation model will be smoothed across the fault zone. However the deformation is still measured and is used to update the coordinates. It is also added to the deformation model using a reverse step function that applies a negative deformation that applies when transforming coordinates for epochs before the earthquake. In this case transforming coordinates at an epoch before the earthquake will result in subtracting the reverse patch from the coordinates. This adds the deformation to the coordinates, which again is the correct update to coordinates to transform them to the new version of the datum.

BIBLIOGRAPHY

BIBLIOGRAPHY