OGC®
Making location count.

ABSTRACT SPECIFICATION TOPIC FOR OGC DEFORMATION MODEL FUNCTIONAL MODEL

ABSTRACT SPECIFICATION TOPIC

CANDIDATE SWG DRAFT

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Copyright notice

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

<Insert abstract here.>

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 PREFACE

<Insert Preface text here.>

NOTE: Give OGC specific commentary: describe the technical content, reason for document, history of the document and precursors, and plans for future work.

There are two ways to specify the Preface: "simple clause" or "full clasuse"

If the Preface does not contain subclauses, it is considered a simple preface clause. This one is entered as text after the `.Preface` label and must be placed between the AsciiDoc document attributes and the first AsciiDoc section title. It should not be give a section title of its own.

If the Preface contains subclauses, it needs to be encoded as a full preface clause. This one is recognized as a full Metanorma AsciiDoc section with te title "Preface", i.e. `== Preface`. (Simple preface content can also be encoded like full preface.)

IV   SECURITY CONSIDERATIONS

No security considerations have been made for this document.

V SUBMITTING ORGANIZATIONS

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

- Land Information New Zealand
- ESRI

VI SUBMITTERS

All questions regarding this submission should be directed to the editor or the submitters:

NAME	AFFILIATION
Chris Crook	Land Information New Zealand
Kevin Kelly	ESRI

VII INTRODUCTION

<Insert introduction here.>

VIII REFERENCE NOTES

<Place reference notes here.>

NOTE: If you need to place any further sections in the preface area use the `[.preface]` attribute.

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 position of the point is represented as a displacement from a reference position for that point.

A deformation model allows a "fixed" object's location to be propogated to different times. Here "fixed" means in the intuitive sense — fixed to the earth's crust and moving with it.

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.

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 that parameterisation.

This specification allows producers of deformation models, typically national geodetic agencies, to publish the deformation model in a consistent way and for users of coordinate transformation software to be confident that the deformation model is being used as intended by the producers.

1.1. EXCLUSIONS

This scope of this functional model definition does not 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 deformation between two datums (typically between older datums based on local terrestrial observations and modern datums based on global satellite systems.)

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

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO: ISO 19101-1:2014, *Geographic information — Reference model — Part 1: Fundamentals*. International Organization for Standardization, Geneva (2014). https://www.iso.org/standard/59164.html

ISO: ISO 19103:2015, *Geographic information — Conceptual schema language*. International Organization for Standardization, Geneva (2015). https://www.iso.org/standard/56734.html

John Herring: OGC 06-103r4, *OpenGIS Implementation Specification for Geographic information — Simple feature access — Part 1: Common architecture*. Open Geospatial Consortium (2011). https://portal.ogc.org/files/?artifact_id=25355

Clemens Portele: OGC 07-036r1, *OpenGIS Geography Markup Language (GML) Encoding Standard — with corrigendum*. Open Geospatial Consortium (2018). https://portal.ogc.org/files/?artifact_id=74183&version=2

4

TERMS AND DEFINITIONS

4 TERMS AND DEFINITIONS

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

4.1. accessible coordinate system

A coordinate system within which positions can be measured directly?!

Note 1 to entry: Need a meaningful definition here. Need to track this one down to its source.

Note 2 to entry: This section provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

Example   The position used to calculate the spatial model is not defined in an currently accessible coordinate system == Conventions

Source: OGC 07-036r1

4.2. Identifiers

The normative provisions in this standard are denoted by the URI

http://www.opengis.net/spec/{standard}/{m.n}

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

4.3. Other conventions

<Place any other convention needed with its corresponding title>

5

FUNCTIONAL MODEL DEFINITION

5 FUNCTIONAL MODEL DEFINITION

The functional model for a deformation model is characterised as follows.

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

b) The deformation model defines a trajectory for each point on the physical surface by adding the calculated displacement as function of time to the position used to evaluate the spatial model. This trajectory is terms of an explicitly defined accessible coordinate system.

c) The position used to calculate the spatial model is not defined in an currently accessible coordinate system — it may be in terms on an accessible coordinate system at a specific epoch. Its value is only accessible by an inverse calculation using the deformation model. See the inverse calculation method below.

d) The total deformation is partitioned into one or more components. The total deformation at any time and position is calculated by summing the contribution from each component. See discussion and formulae below.

e) Each component comprises a spatial model and a time function. The spatial model defines displacement as a function of position. The time function evaluates a scalar multiplier as a function of time. The deformation contribution for the component at a given position and time is obtained by multipying the displacement components by the time function.

f) The spatial model may be defined over a subset of the total extent of the model. Where it is not defined it is assumed to be zero (see the discussion below).

g) The spatial model may define any or all of the following quantities:

- horizontal displacement – the east and north displacement in metres or degrees

- vertical displacement – the upwards vertical displacement in metres.

Displacements not defined by the model are assumed to be zero. Uncertainties not defined by the model take a default value defined in the spatial model or deformation model metadata.

h) The spatial model may define the uncertainty of the displacement in metres at each node.

Horizontal uncertainty can be represented by:

- horizontal circular uncertainty e_h

- east and north uncertainty e_e, e_n

- covariance matrix components [c_{ee} , c_{en}, c_{nn}].

Vertical uncertainty is represented by a single value e_v.

Alternatively the uncertainty can be represented by a full covariance matrix at the node [c_{ee} , c_{en}, c_{nn}, c_{eu}, c_{nu}, c_{uu}]

The parameters used to represent uncertainty will be the same for all spatial models in the deformation model and will be defined in the deformation model header.

The header will also define probability level for the uncertainty, examples would be "1 standard error", "95% confidence"

i) Each node may have a quality parameter used to identify potential issues in the quality of the deformation model in the vicinity of the node. For example it could indicate that there is surface faulting affecting cells adjacent to the node. See the discussion below.

j) A producer may include additional parameters at each node that will be ignored by compliant software. The set of parameters must be the same for each node of a spatial model.

k) The displacements and uncertainties of a spatial model may evaluate to a *no data* value at some locations, meaning that the deformation cannot be evaluated at that location. At such locations the total deformation is undefined and cannot be calculated. See the discussion below.

l) The displacement defined by the deformation 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 enforced by the mathematical formulation of the deformation model. It is a compliance requirement on producers of deformation models and can be assumed by implementors of software using the deformation model. See the discussion below.

m) The spatial model is represented by a nested grid structure comprising one or more grids. To evaluate the spatial model at a specific position the nesting algorithm identifies which grid is applicable at that position and the disclocation is interpolated from this grid alone. The grids are constrained to be:

- Grids with rows equally spaced and columns equally spaced.

- Grid rows and columns are aligned with the axes of the definition CRS

- Grids are strictly nested. Grids may share a common edge but otherise may not intersect unless one is a child grid fully contained by the other grid. The model may have more than one root grid (not contained in any other grid).

See grid interpolation formulae below.

See also the discussion of spatial model types below.

n)   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 piecewise linear time function defined by a start and end time and a start and end scale factor. Two specialisations of the piecewise linear 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

See the discussion and formulae for time functions below.

o)   The deformation model specification may also include a 14 parameter Bursa-Wolf transformation definition. This is applied after the deformation components in a forward transformation or before them in an inverse transformation as described in the formula.

The 14 parameter transformation can be used to represent the bulk of the transformation between the source and target coordinate systems or epochs. This allows the more complex deformation components to efficiently represent perturbations from this movement which may be smaller and localised. Note that jurisdictions may choose to implement this using an intermediate coordinate system rather including it than deformation model. A deformation model should not be used to define a transformation which is fully described by a 14 parameter transformation.

p)   Each component includes metadata defining:

- The type of spatial model (grid)

- The spatial interpolation method to use

- The quantities it defines (displacements, uncertainties)

- A spatial definition of the extent of the spatial model (to determine if it is required at a specific position)

- (optional) default horizontal and vertical uncertainty which applies if the spatial model does not explicitly define uncertainty.

- (optional) text description of the source of the deformation represented in the model.

- (optional) definition of areas where quality is impacted, for example where there is surface faulting. The areas each include a description, multipolygon defining the extent of the affected area, and a start and end epoch for the event causing the unmodelled deformation. See the discussion below.

q) The deformation model includes metadata defining:

- The version of the deformation model specification with which the model complies

- The name of the model

- The version of the deformation model

- The publication date

- The licence under which the model is published

- Optional description of the model

- Contact information for the agency publishing the model

- Optional links to reference information about the model

- The source CRS definition (eg EPSG:xxxx)

- The target CRS definition (if the deformation model is implemented as a point motion model this will be the same as the source CRS).

- The spatial model definition CRS (the used to define the position at which the model is calculated)

- The units of horizontal displacements

- The units of vertical displacements

- The format for representing uncertainty — which parameters are used at each node, and what level of confidence. For example horizontal covariance, vertical uncertainty, and 95% confidence level)

- The total spatial extent of the model

- The time extent of the model

- The algorithm used to apply add the deformation to the reference position coordinates.

6

CALCULATION FORMULAE

CALCULATION FORMULAE

The formulae below define the expected behaviour of software using the deformtion model. While an implementation may use a different method it should produce numerically identical results to be a compliant implementation. This provides users confidence that their software is using the deformation model as intended by the producers of the model.

The formulae below are adopted from the JSON GeoTIFF specification. This does not describe interpolating uncertainty other than represented by horizontal and vertical uncertainties e_h, e_v. In this specification the grid can be defined either in terms of a geographic (longitude/latitude) or projection (easting/northing) coordinate system. Displacements and uncertainties are all in metres except that the horizontal displacement may be in degrees if the coordinate system is geographic.

6.1. SPATIAL INTERPOLATION

For all the proposed spatial model and interpolation types the location of the calculation point is used to identify the set of grid nodes that influence the calculated deformation at that point, and also a set of scalar multipliers or weights defining ths significance of that grid node on the calculated value. That is these the value of any dimension (eg east displacement) at a specific evaluation point is evaluated by forming a weighted average of the values at a set of nodes P_1, P_2, .. with weights w_1, w_2. For bilinear interpolation on a grid this is a set of four nodes on the corners of the grid cell within which the evaluation point lies.

The values at the nodes are combined using these weights as described below. For the horizontal components (east and north displacement) two combination methods are defined — a simple weighted average and a geocentric combination which is applicable near the geographic poles where the east and north directions at the nodes may be markedly different.

6.1.1. Bilinear grid interpolation

Gridded spatial representations are defined as regular grids in terms of an interpolation coordinate system. The coordinates of a grid node is 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, and i and j are the column and row number of the grid node (where the west-most column and southernmost row are numbered 0). 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 latitude and longitude from the nodes at the corners of the grid cell within which the calculation point lies. Each element of the displacement is calculated independently (though of course the interpolation weighting will be the same for each, as they all refer to the same calculation point).

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.

Figure 1
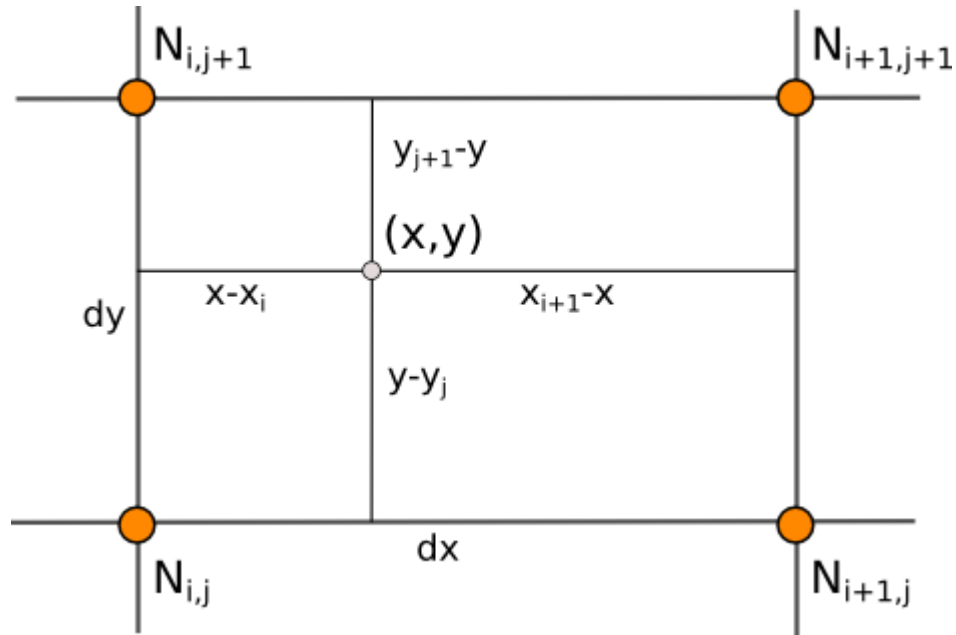
The displacement elements (de, dn, du) at the calculation point are weighted means of the corresponding elements at the four nodes.

The weights are calculated as follows:

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

6.1.2. Calculation of weighted average of node values

Each interpolation method identifies a set of nodes P_i and weights W_i. At each node the spatial model may define any of the displacements de_i, dn_i, du_i, and uncertainties eh_i and ev_i. For the horizontal displacement components de and dn two formulae are supported. Generally a simple

weighted average will suffice, but near the poles a more complex "geocentric weighted average" formulae describe below may be preferable. This is also used for the vertical component du.

So for example the east displacement at the point (x,y) is calculated as

$$de = \sum_i W_i de_i$$

The uncertainties eh, ev are interpolated using a weighted average of the variances eh_i^2, ev_i^2, for example

$$eh = \sqrt{\sum_i W_i eh_i^2}$$

6.1.3. Geocentric weighted average

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 the set of nodes being averaged, as shown in the figure where the directions at nodes A and B and the evaluation point P are all quite different. The impact of this is discussed below under errors of normal bilinear interpolation near poles.

Figure 2
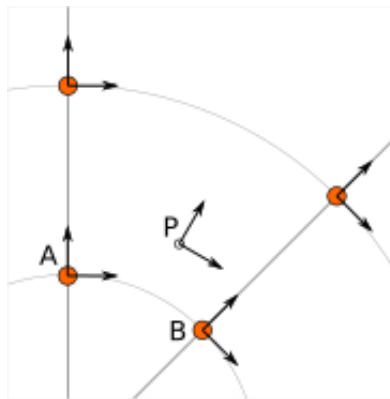
In such areas it may be more appropriate to base the grid on a local projected coordinate system. Also there is no issue even in polar regions if the grid has very narrow grid cells spanning a small longitude range.

If grid cells spanning a large longitude range are used then a weighted average of the grid node displacements can be adapted by transforming the east and north vectors to a common direction before averaging them.

The geocentric interpolation method converts the displacement components from east and north components to geocentric X, Y, and Z components. These are in the same direction and can be scaled and summed using the simple weighted avarage formulae above to calculate the

X, Y, Z components of displacement at the calculation point, which are then converted back to components east and north at the calculation point.

Note that this is only used to determine the horizontal displacement. The vertical displacement and uncertainties are computed using the simple weighted average formulae above.

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

$dx_i = -de_i.sin(\lambda_i) — dn_i.cos(\lambda_i).sin(\varphi_i)$
$dy_i = de_i.cos(\lambda_i) — dn_i.sin(\lambda_i).sin(\varphi_i)$
$dz_i = dn_i.cos(\varphi_i)$

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 as described above, eg:

$$dx = \sum_i W_i dx_i$$

The displacement at the calculation point is then calculated as:

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

6.2. TIME FUNCTIONS

The time function f(t) for a component evaluates a scalar value by which the spatial model displacement is multiplied to determine the actual displacement at time t. For example in a velocity model the spatial mode represents the displacement that happens in one year, and the time function evaluates the scale factor f(t) applied to component displacements at time t as the number of years since a reference epoch ($t — t_0$).

The deformation model metadata defines a temporal extent of the entire deformation model from T_{min} to T_{max}. This overrides any information from specific time functions. At times before T_{min} and for times at or after T_{max} every time function is considered undefined and the deformation model cannot be evaluated.

Following conventional use in deformation models the time functions are defined as a functions of decimal years (eg velocities are metres/year). All date/time values such as calculation epoch, velocity reference epoch) are converted to decimal years for use in the following formulae.

The conversion of a UTC epoch *yyyy-mm-dd*T*HH:MM:SS*Z to decimal years is done by first converting the epoch to UTC. 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 second at the end of the year. 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, or if even if they do it may take some time before the leap second is incorporated into the libraries installed by a particular user. Since this difference makes no practical difference to deformation model calculations it is proposed that the calculation of the decimal year is considered compliant whether or not it includes it — there are two nominally correct answers.

The time function is defined as the sum of one or more of the following generic functions. In this table some functions include optional parameters indicated by italics.

Table 1

TIME FUNCTION TYPE	PARAMETERS	FORMULA ($T_{I,MIN} \leq T < T_{I,MAX}$)
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
piecewise linear	Start epoch t_s	$f(t) = f_s$ for $t < t_s$
	Start scale factor f_s	$f(t) = (f_s.(t_e - t) + f_e.(t-t_s))/(t_e-t_s)$ for $t_s \leq t < t_e$
	End epoch t_e	$f(t) = f_e$ for $t \geq t_e$
	End scale factor f_e	
exponential	Start epoch t_s	$f(t) = 0$ for $t < t_s$
	Start amplitude α_0	$f(t) = \alpha_0 + (\alpha_\infty - \alpha_0).(1 - \exp(-(t-t_s)/\theta))$ for $t \geq t_s$
	Final amplitude α_∞	$f(t) = f(t_e)$ for $t > t_e$ if t_e is defined
	Decay constant θ	
	End epoch t_e	
logarithmic	Start epoch t_s	$f(t) = 0$ for $t < t_s$
	Scale factor α	$f(t) = \alpha.\ln(1 + (t-t_s)/\tau)$ for $t_s \leq t$
	Time constant τ	$f(t) = f(t_e)$ for $t > t_e$ if t_e is defined
	End epoch t_e	
acceleration	Function reference epoch t_0	$f(t) = a.(t - t_0)^2$
	Acceleration a	$f(t) = f(t_s)$ for $t < t_s$ if t_s is defined
	Start epoch t_s	$f(t) = f(t_e)$ for $t > t_e$ if t_e is defined
	End epoch t_e	
hyperbolic tangent	Start epoch t_s	$f(t) = 0$ for $t < t_s$
	End epoch t_1	$f(t) = A.\tanh((t - t_r)/\tau)$ for $t \geq t_s$ and $t < t_e$
	Time constant τ	$f(t) = \alpha$ for $t \geq t_e$
	Scale factor α	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)	$f(t) = \alpha\cos(f(t - t_0)/2\pi) + \beta\sin(f(t - t_0)/2\pi)$
	Function reference epoch t_0	
	Cosine amplitude α	
	Sine amplitude β	

To improve simplicity and readability of the time function definition two common special cases are supported as a specialisations of the piecewise linear function:

Table 2

FUNCTION TYPE	PARAMETERS	FORMULA ($T_{I,MIN} \leq T < T_{I,MAX}$)
step	Step epoch t_s	$f(t) = 0$ when $t < t_s$,
		$f(t) = 1$ when $t \geq t_s$
reverse step	Step epoch t_s	$f(t) = -1$ when $t < t_s$,
		$f(t) = 0$ when $t \geq t_s$

Note: a velocity function including a start and end epoch is functionally identical to the piecewise linear function. However these two options are provided to support quite different cases.

Typically the velocity function will not be specified with both a start time and an end time. It is representing secular deformation and the displacements in the spatial model are the deformation accumulating in one year regardless of the start and end time.

The piecewise linear function is to support a specific time bounded event. It more meaningfully represents specific deformation events as the spatial model displacements reflect the total deformation. The step functions are specialisations of it in which the start and end epoch are equal (which is not possible with a velocity function, as the velocity is infinite). The main role of the piecewise linear function in deformation model time functions is that multiple such functions can be combined to approximate any time evolution.

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

6.3. COMBINATION OF COMPONENTS

To calculate the total deformation at a time and location, the displacement and uncertainties due to each component are calculated independently and then combined using the formulae below to obtain the total displacement and uncertainty at a location. The total displacement is then applied to the coordinate.

The same input position coordinate is used for each component — the components are not applied sequentially (ie the coordinate is not updated by the first component before being used to calculate the deformation on the second component). See the discussion below on using the same position for each component.

At a given time and location the elements from each component are combined to determine the overall displacement and errors.

The displacement elements de, dn, dh are combined by simply adding their values calculated for each component. For example, if there are n components for which the spatial representation

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 model value for de is

de = $f_1.de_1$ + $f_2.de_2$ + … + $f_n.de_n$

The error values eh, ev are combined by determining the root sum of squares (RSS) of the values determined for each component. So for example

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

6.4. APPLYING THE OFFSET TO A COORDINATE

For each physical point on the surface for which the deformation model is defined the model defines the trajectory of that point in the target coordinate system. The trajectory is defined for any valid epoch by adding the displacement calculated from the model to a reference coordinate for the point — a time invariant coordinate for the point at which displacement is zero. The reference coordinate is not directly measurable — it is accessed by measuring a position in the target coordinate system and then subtracting the displacement from the deformation model at the epoch of measurement using the inverse formula below.

The method used to add the calculated displacement to the reference coordinate is defined in the deformation model metadata. Two methods are proposed — *addition* and *geocentric addition*. The *addition* method is relatively simple and adds the offset to the coordinates, converting metres to degrees first if necessary. The *geocentric* method is an alternative method that may be used near the poles if the grid latitude spacing is relatively large and accounts for the potentially significant differences in the north and east directions at each grid node. It is only applicable if the offsets are defined in metres and the coordinate system is a geographic (latitude/longitude) system. The *addition* and *geocentric addition* methods are detailed below.

6.4.1. Addition method

The method of the calculated east/north/up displacement to a coordinate depends on the units of the displacement and the type of the source and target coordinate system. Also for geographic coordinate systems the method described here does not apply very 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 ordinate.

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

If the source and target coordinate systems are geographic coordinate systems and the east and north displacement units are metres then the displacement components must be converted to degrees before they are added to longitude and latitude. 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 model is using a geographic coordinate system and offsets measured in metres. In this case the horizontal offset is converted to a geocentric (XYZ) offset, added to the geocentric coordinate, and then converted back to geographic coordinates. The vertical coordinate is always calculated by simple addition of the vertical displacement to the reference coordinate height.

This method may be applicable for coordinates near the pole, where simple addition of offsets to the longitude/latitude may not give the desired result. This is shown in the figure below in which 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 eastwards vector at the reference coordinate. Close to the pole the eastward vector is different to changing the longitude coordinate.. .

Figure 3
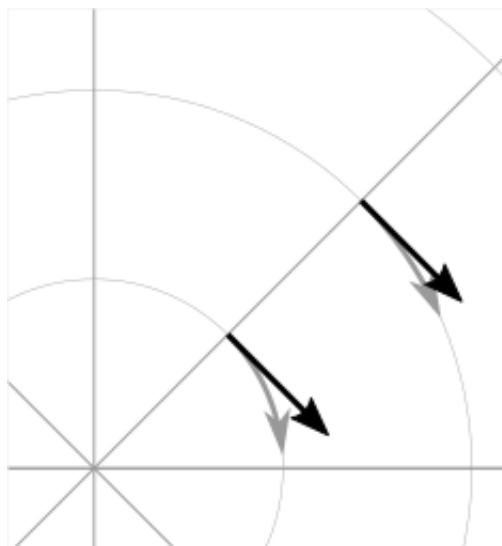
Moving away from the pole this issue rapidly becomes insignificant. For a point at distance R from the pole with a displacement d, the difference is approximately d*(1-cos(d/R)), or approximately $d^3/2R^2$. So for example a 1m eastward displacement 10km from the pole would have an error of 10^{-8}m. This is only an issue very close to the pole!

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

Standard formulae are used for the conversion of geographic coordinates to and from geocentric coordinates. The initial ellipsoidal height is set to zero before converting, and the resultant ellipsoidal height is discarded.

The horizontal components of displacement are converted to X,Y,Z components using the same formulae as described for the "geocentric_bilinear" method.

While this method can be used at any location it is not recommended other than close to the poles. It is computationally very expensive compared to simply adding the offsets to longitude and latitude.

6.5. CALCULATION OF THE INVERSE DEFORMATION MODEL

Calculating the inverse of the deformation model requires an iterative solution as the coordinate in the defintion coordinate reference system is required to evaluate the deformation model, but it is not known until the deformation has been calculated and applied to the input coordinate in the target coordinate reference system.

The iteration is done by

- using the input coordinate as an initial estimate for the output coordinate

- at each iteration:

  - apply the deformation model to the current estimate of the output coordinate

  - calculate the difference between the calculated coordinate and the input coordinate

  - subtract this difference from the current estimate solution to obtain the estimated solution for the next iteration

  - if this difference is less than the precision required for the inverse operation then finish

The calculation of the difference and the subtraction of the difference from the current estimate is done by the "addition" or "geocentric" method, as defined in the deformation model metadata. (Formulae are defined above.)

See the discussion below on the iterating the inverse calculation.

6.6. CALCULATION OF DEFORMATION BETWEEN TWO EPOCHS

Calculating the deformation between two times is straightforward for the displacement elements de, dn, and du as it is simply the difference between the values calculated at each time.

This approach is not appropriate for the error components eh, ev. Uncorrelated errors are combined as a root sum of squares, but the errors of displacements calculated for one component calculated at different times are clearly correlated.

While there is no mathematically correct way to define the errors without a much more complex error model, the following approach is recommended if these errors are required.

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

The eh and ev values from the spatial representation of each component are multiplied by these time function error factor values and then combined as the root sum of squares to give the total error of the deformation between the two epochs.

6.7. CALCULATION OF THE 14 PARAMETER TRANSFORMATION

If the model includes a 14 parameter transformation then this is applied to the coordinates after the deformation model is calculated and applied in a forward transformation. In an inverse transformation it is applied (in reverse) before the deformation model components area been applied to the coordinate.

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

ANNEX A (INFORMATIVE) EXPLANATIONS AND EXPANDED DISCUSSION POINTS

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
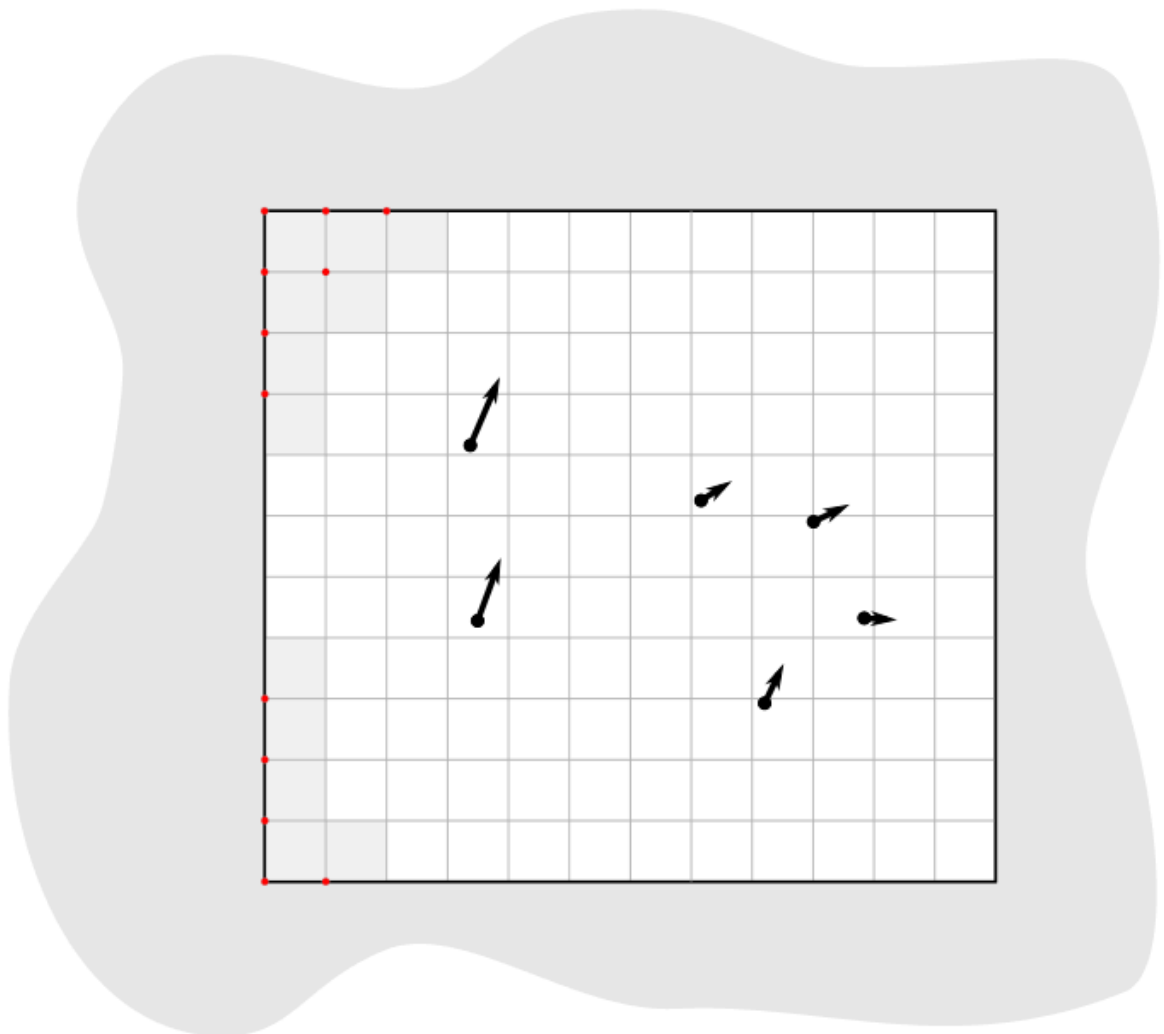
This may occur where the area in which deformation is defined is an irregular shape. It might include coastal regions where the deformation of the seabed is not measured, or it may be that it crosses a jurisdictional boundary. As the gridded model is by definition a rectangular area it will include these 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.
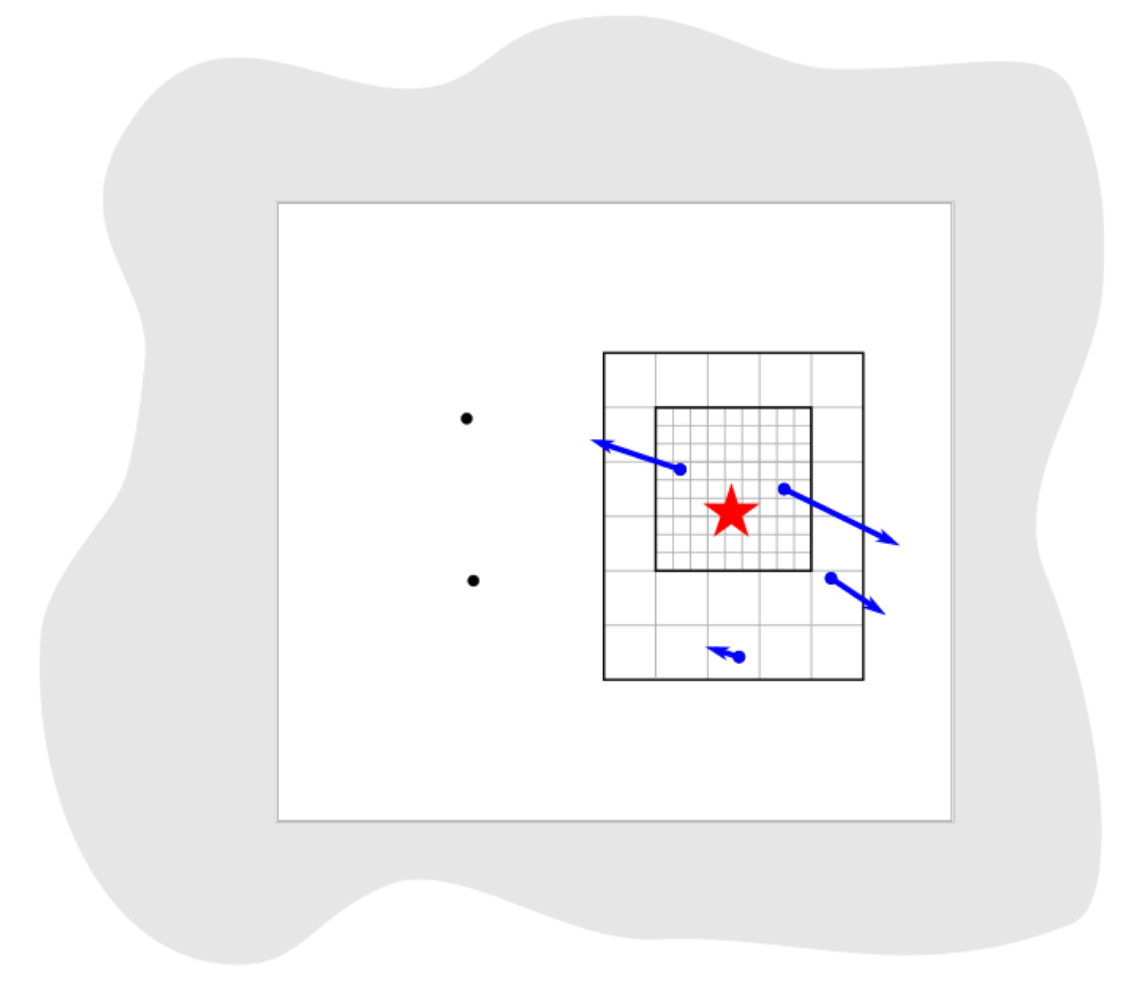
A.4. QUALITY PARAMETERS AT SPATIAL MODEL NODES

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
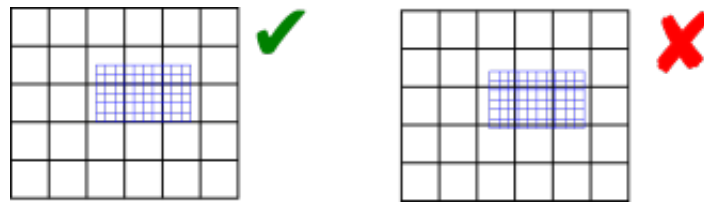
A question for implementers is how to transform data that extends beyond the deformation model. If, as is likely, the deformation is not zero at the edge of the model then there is discontinuity across the boundary. There are a number of possible approaches to handling this in the deformation model functional 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.

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
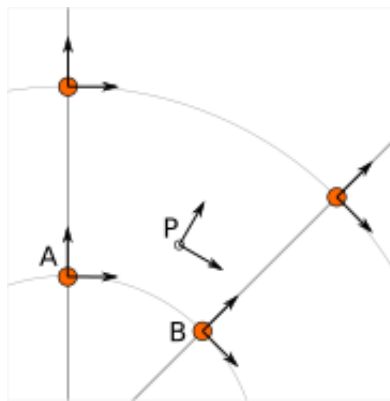
To estimate the error that could be incurred using simple bilinear interpolation and not accounting for this difference we can consider a case where the deformation is 1m northwards at point A, and zero at point B in the diagram above. Let the longitude grid spacing be λ_S radians. If the calculation 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. 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