Open Geospatial Consortium

Submission Date: 2022-12-16

Approval Date: yyyy-mm-dd

Publication Date: yyyy-mm-dd

External identifier of this OGC® document:
https://github.com/opengeospatial/CRS-Gridded-Geodetic-data-eXchange-Format/tree/master/specifiation

Internal reference number of this OGC® document: 22-051r1

Version: 1.0.1

Category: OGC® Implementation Standard

Editor: Roger Lott

The GGXF geodetic data grid exchange format

Warning

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Implementation Standard
Document stage:	Draft
Document language:	English

i. Abstract

The Geodetic data Grid eXchange Format (GGXF) is designed to be a single file format that may be used for a wide range of geodetic applications requiring interpolation of regularly gridded data, including (but not limited to):

- Transformation of latitude and longitude coordinates from one geodetic coordinate reference system to another;
- Transformation of gravity-related heights from one vertical coordinate reference system to another;
- Reduction of ellipsoid heights to the geoid, quasi-geoid or a surface of a vertical reference frame;
- The description of coordinate changes due to deformation.

The GGXF format has been designed specifically for carrying gridded geodetic parameters supporting coordinate transformations and point motion operations but has no restriction on the type of content that may be included.

ii. Keywords

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

ogcdoc, OGC document, GGXF, geodetic, grid, exchange format, CRS, coordinate reference system

iii. Preface

A single, standard, grid file format offers the following advantages:

- Grid producers do not have to create file formats themselves, provide their own software to read and interpolate their gridded data or concern themselves with lack of take-up of their data due to its proprietary distribution format;
- Survey and geographic information software vendors need to read only one grid file format, eliminating the need to repeatedly write programs to import different grids;
- End users can use a new grid file as soon as it became available, without having to wait for their application vendor to produce a software upgrade.

GGXF has been designed to cope with multiple levels of data resolution, engender computational efficiency, be straightforward for grid producers to populate and easy and efficient for application developers to use.

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.

iv. Security Considerations

No security considerations have been made for this Standard

v. Submitting organizations

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

Name	Affiliation
Elmar Brockmann	SwissTopo
Michael Craymer	NR Canada
Chris Crook	LINZ
Martin Desruisseaux	Geomatys
Kevin Kelly	ESRI
Roger Lott (editor)	IOGP
Chris Pearson	Trimble Inc
Dan Roman	US NGS
Keith Ryden	ESRI
Richard Stanaway	Quickclose Pty
Patrick Vorster	South Africa NGI

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

Contents

Introduction

The Geodetic data Grid eXchange Format (GGXF) is designed to be a single file format that may be used for a wide range of geodetic applications requiring interpolation of regularly gridded data, including (but not limited to):

- Transformation of latitude and longitude coordinates from one geodetic coordinate reference system to another;
- Transformation of gravity-related heights from one vertical coordinate reference system to another;
- Reduction of ellipsoid heights to the geoid, quasi-geoid or a surface of a vertical reference frame;
- The description of coordinate changes due to deformation.

The GGXF format has been designed specifically for carrying gridded geodetic parameters supporting coordinate transformations and point motion operations but has no restriction on the type of content that may be included. GGXF is designed to be extensible to all geodetic data that is presented as a grid.

A single, standard, grid file format offers the following advantages:

— Grid producers do not have to create file formats themselves, provide their own software to read and interpolate their gridded data or concern themselves with lack of take-up of their data due to its proprietary distribution format;

— Survey and geographic information software vendors need to read only one grid file format, eliminating the need to repeatedly write programs to import different grids;

— End users can use a new grid file as soon as it became available, without having to wait for their application vendor to produce a software upgrade.

GGXF has been designed to cope with multiple levels of data resolution, engender computational efficiency, be straightforward for grid producers to populate and easy and efficient for application developers to use.

The GGXF geodetic data grid exchange format

1 Scope

The Geodetic data Grid eXchange Format (GGXF) is a file format that may be used for a wide range of geodetic applications requiring interpolation of gridded data.

This document describes GGXF format version 1.0. It supports data values for one or more user-specified parameters at grid nodes of a regularly-spaced grid constructed in a coordinate reference system (CRS) which is spatially either 2-dimensional or 3-dimensional. It supports data population at varying density. It adheres to the data model and terminology described in OGC Abstract Specification Topic 2, Referencing by coordinates [2]. The functional model for deformation grids is described in [1].

This document does not support triangulated irregular networks (TINs).

This document is applicable to producers of gridded geodetic data and to developers of software using gridded geodetic data.

2 Normative references

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

ISO 8601-1, *Date and time — Representations for information interchange — Part 1: Basic rules*

ISO 19111, *Geographic information — Referencing by coordinates*

ISO 19115-1, *Geographic information — Metadata — Part 1: Fundamentals*

ISO 19125-1, *Geographic information — Simple Features Access — Part 1: Common Architecture*

ISO 19162, *Geographic information — Well-known text representation of coordinate reference systems*

ISO 80000-3, *Quantities and units — Part 3: Space and time*

OGC 22-010, Abstract Specification Topic ##, *Deformation Model Functional Model*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

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

3.1.1
compound coordinate reference system
coordinate reference system using at least two independent coordinate reference systems

Note 1 to entry: Coordinate reference systems are independent of each other if coordinate values in one cannot be converted or transformed into coordinate values in the other.

[SOURCE: ISO 19111:2019, 3.1.3]

3.1.2
coordinate epoch
epoch to which coordinates in a dynamic coordinate reference system are referenced

[SOURCE: ISO 19111:2019, 3.1.7]

3.1.3
coordinate operation
process using a mathematical model, based on a one-to-one relationship, that changes coordinates in a source coordinate reference system to coordinates in a target coordinate reference system, or that changes coordinates at a source coordinate epoch to coordinates at a target coordinate epoch within the same coordinate reference system

Note 1 to entry: Generalisation of coordinate conversion, coordinate transformation and point motion operation.

[SOURCE: ISO 19111:2019, 3.1.8]

3.1.4
coordinate transformation
coordinate operation that changes coordinates in a source coordinate reference system to coordinates in a target coordinate reference system in which the source and target coordinate reference systems are based on different datums

Note 1 to entry: A coordinate transformation uses parameters which are derived empirically. Any error in those coordinates will be embedded in the coordinate transformation and when the coordinate transformation is applied the embedded errors are transmitted to output coordinates.

Note 2 to entry: A coordinate transformation is colloquially sometimes referred to as a 'datum transformation'. This is erroneous. A coordinate transformation changes coordinate values. It does not change the definition of the datum. In this document coordinates are referenced to a coordinate reference system. A coordinate transformation operates between two coordinate reference systems, not between two datums.

[SOURCE: ISO 19111:2019, 3.1.12]

3.1.5
coordinate tuple
tuple composed of coordinates

Note 1 to entry: The number of coordinates in the coordinate tuple equals the dimension of the coordinate system; the order of coordinates in the coordinate tuple is identical to the order of the axes of the coordinate system.

[SOURCE: ISO 19111:2019, 3.1.13]

3.1.6
displacement
change in coordinates of a point due to deformation

[SOURCE: Deformation Model glossary]

3.1.7
easting
E
distance in a coordinate system, eastwards (positive) or westwards (negative) from a north-south reference line

[SOURCE: ISO 19111:2019, 3.1.21]

3.1.8
ellipsoidal height
geodetic height
h
distance of a point from the reference ellipsoid along the perpendicular from the reference ellipsoid to this point, positive if upwards or outside of the reference ellipsoid

Note 1 to entry: Only used as part of a three-dimensional ellipsoidal coordinate system or as part of a three-dimensional Cartesian coordinate system in a three-dimensional projected coordinate reference system, but never on its own.

[SOURCE: ISO 19111:2019, 3.1.24]

3.1.9
epoch
<geodesy> point in time

Note 1 to entry: In this document an epoch is expressed in the Gregorian calendar as a decimal year.

EXAMPLE 2017-03-25 in the Gregorian calendar is epoch 2017.23.

[SOURCE: ISO 19111:2019, 3.1.27]

3.1.10
geodetic coordinate reference system
two- or three-dimensional coordinate reference system based on a geodetic reference frame and having either a three-dimensional Cartesian or an ellipsoidal or a spherical coordinate system

Note 1 to entry: In this document a coordinate reference system based on a geodetic reference frame and having an ellipsoidal coordinate system is geographic.

[SOURCE: ISO 19111:2019, 3.1.31]

3.1.11
geodetic latitude
ellipsoidal latitude
φ
angle from the equatorial plane to the perpendicular to the ellipsoid through a given point, northwards treated as positive

[SOURCE: ISO 19111:2019, 3.1.32]

3.1.12
geodetic longitude
ellipsoidal longitude
λ
angle from the prime meridian plane to the meridian plane of a given point, eastward treated as positive

[SOURCE: ISO 19111:2019, 3.1.33]

3.1.13
geographic bounding box
geographic position of an area described by the maximum and minimum latitude and longitude bounding it.

Note 1 to entry: This is only an approximate reference so specifying the geographic coordinate reference system is unnecessary and bounding coordinates need only be provided with a precision of up to two decimal places of a degree.

3.1.14
geographic coordinate reference system
coordinate reference system that has a geodetic reference frame and an ellipsoidal coordinate system

[SOURCE: ISO 19111:2019, 3.1.35]

3.1.15
geoid height
distance from the surface of the reference ellipsoid of a specified geographic 3D coordinate reference system to a surface realizing the geoid, positive if upwards or outside of the reference ellipsoid

Note 1 to entry: In this document this term is used in a general sense for any geoid model. It includes gravimetric geoids (quasi geoids), hybrid height correction models, etc.

3.1.16
graticule
grid consisting of parallels of latitude and meridians of longitude

3.1.17
gravity-related height
H
height that is dependent on the Earth's gravity field

Note 1 to entry: This refers to, amongst others, orthometric height and Normal height, which are both approximations of the distance of a point above the mean sea level, but also may include Normal-orthometric heights, dynamic heights or geopotential numbers.

Note 2 to entry: The distance from the reference surface may follow a curved line, not necessarily straight, as it is influenced by the direction of gravity.

[SOURCE: ISO 19111:2019, 3.1.37]

3.1.18
height
distance of a point from a chosen reference surface positive upward along a line perpendicular to that surface

Note 1 to entry: A height below the reference surface will have a negative value.

Note 2 to entry: Generalisation of ellipsoidal height (h) and gravity-related height (H).

[SOURCE: ISO 19111:2019, 3.1.38]

3.1.19
interpolation CRS
coordinate reference system to which a grid is referenced and in which interpolation of the grid is performed

3.1.20
meridian
intersection of an ellipsoid by a plane containing the shortest axis of the ellipsoid

Note 1 to entry: This term is generally used to describe the pole-to-pole arc rather than the complete closed figure.

[SOURCE: ISO 19111:2019, 3.1.42]

3.1.21
northing
N
distance in a coordinate system, northwards (positive) or southwards (negative) from an east-west reference line

[SOURCE: ISO 19111:2019, 3.1.43]

3.1.22
offset
quantity that is added [to a coordinate]

Note 1 to entry: The value of the offset may be positive or negative.

3.1.23
period
extent in time

Note 1 to entry: A period is bounded by two different temporal positions.

[SOURCE: ISO 19108:2005, 4.1.27]

3.1.24
point motion operation
coordinate operation that changes coordinates within one coordinate reference system due to the motion of the point

Note 1 to entry: The change of coordinates is from those at an initial epoch to those at another epoch.

Note 2 to entry: In this document the point motion is due to tectonic motion or crustal deformation.

[SOURCE: ISO 19111:2019, 3.1.48]

3.1.25
projected coordinate reference system
coordinate reference system derived from a geographic coordinate reference system by applying a map projection

Note 1 to entry: May be two- or three-dimensional, the dimension being equal to that of the geographic coordinate reference system from which it is derived.

Note 2 to entry: In the three-dimensional case the horizontal coordinates (geodetic latitude and geodetic longitude coordinates) are projected to northing and easting and the ellipsoidal height is unchanged.

[SOURCE: ISO 19111:2019, 3.1.51]

3.1.26
root grid
grid where the file structure defines this grid to have no parent

3.1.27
secular motion
velocity (of a point) that is continuous over an inter-seismic time period

Note 1 to entry: Geodetic convention is to consider 2.5 years as the minimum period of continuous measurement required to define the velocity of a point. In areas where there are frequent earthquakes and slow-slip events the secular motion can only be defined between these events and the interval between seismic events is often less than 2.5 years.

3.1.28
sequence
finite, ordered collection of related items (objects or values) that may be repeated

[SOURCE: ISO 19111:2019, 3.1.55]

3.1.29
sibling grid
grid where the file structure defines at least one other grid to have the same parent

3.1.30
source coordinate epoch
coordinate epoch to which a coordinate set that is to be input into a point motion operation is referenced

Note 1 to entry: The 'start' epoch.

3.1.31
source coordinate reference system
source CRS
coordinate reference system to which a coordinate set that is to be input into a coordinate transformation is referenced

Note 1 to entry: The 'from' coordinate reference system.

3.1.32
target coordinate epoch
coordinate epoch to which a coordinate set that is output from a point motion operation is referenced

Note 1 to entry: The 'end' epoch.

3.1.33
target coordinate reference system
target CRS
coordinate reference system to which a coordinate set that is output from a coordinate transformation is referenced

Note 1 to entry: The 'to' coordinate reference system.

3.1.34
tuple
ordered list of values

Note 1 to entry: The number of values in a tuple is immutable.

[SOURCE: ISO 19136-1:2019, 3.1.70]

3.1.35
vertical coordinate reference system
one-dimensional coordinate reference system based on a vertical reference frame

[SOURCE: ISO 19111:2019, 3.1.70]

3.2 Abbreviations

CRS	coordinate reference system
GGXF	Geodetic data Grid eXchange Format
netCDF	(Unidata) network Common Data Form
URI	Uniform Resource Identifier
WKT	well-known text format

4 Conformance requirements

Any file claiming conformance to this GGXF specification shall satisfy the conformance requirements in Annex A. Conformance classes are shown in Table 1.

Table 1 — Conformance requirements for a GGXF file

Data content	Conformance requirements given in
core - required for: a) production of GGXF files, and b) consuming GGXF files	A.1
yaml - required by those producing or consuming GGXF YAML text files	A.2
netcdf - required by those producing or consuming GGXF netCDF binary files	A.3

The normative provisions in this specification are denoted by the URI

http://www.opengis.net/spec/ggxf/1.0

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

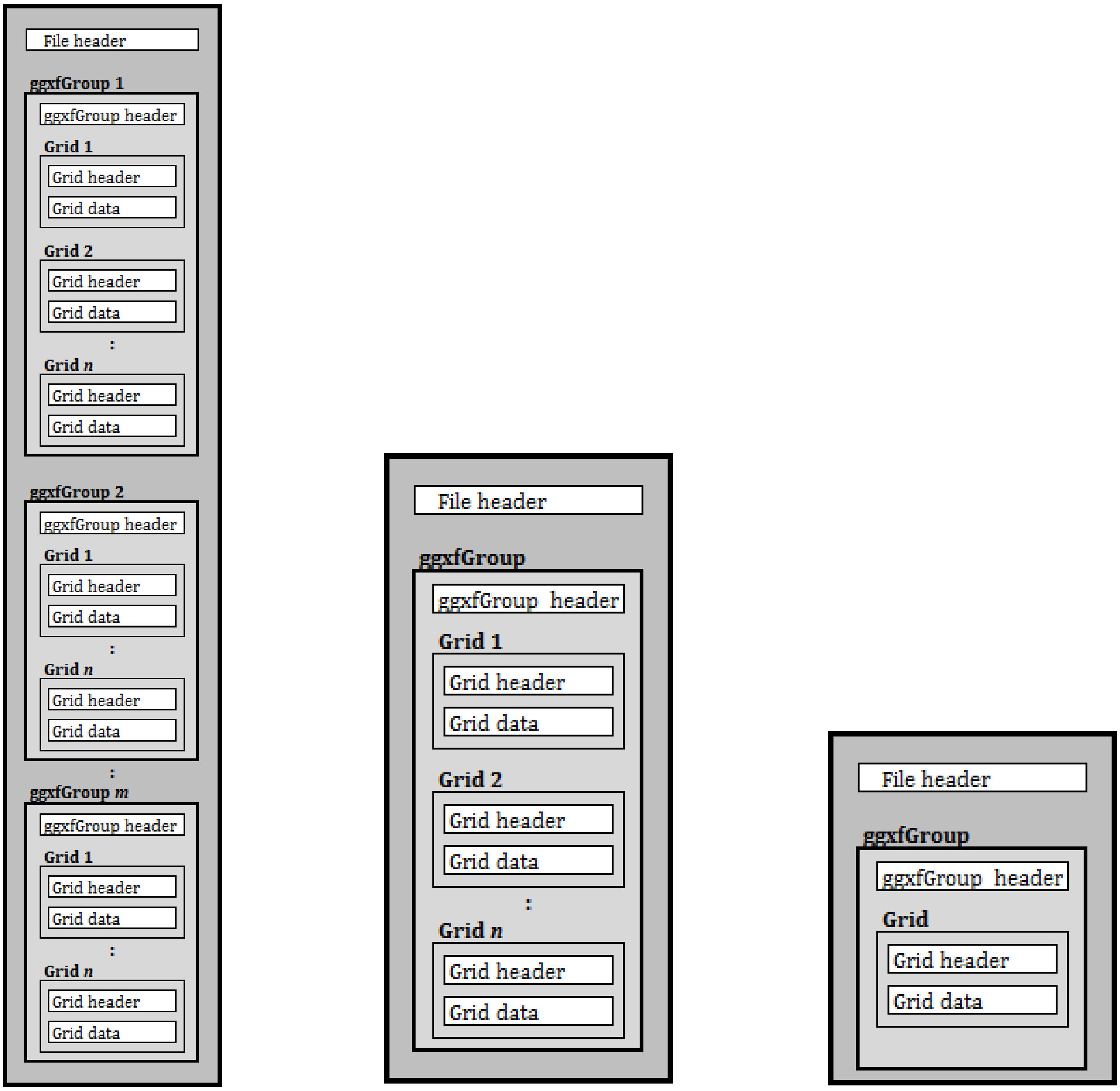
5 Overview of a GGXF file

5.1 File structure

A GGXF file consists of

- A file header giving metadata applicable to the whole file;
- One or more ggxfGroups, where each ggxfGroup consists of
  - A ggxfGroup header giving metadata applicable to the ggxfGroup;
  - One or more grids, each with:
    - A grid header containing metadata applicable to the grid;
    - An array of nodes, at each of which one or more parameter values are given.

In this document the term ggxfGroup is used to distinguish the concept from that of a group in a netCDF file. A conceptual diagram of the GGXF file structure is shown in Figure 1.



(a) general structure (b) multiple grids in a single ggxfGroup (c) single grid

Figure 1 — Conceptual structure of a GGXF file

Most gridded geodetic data does not require the full complexity of this file structure. Most requirements can be met through a subset of the full structure, through either a single grid (Figure 1c) or through a single ggxfGroup containing several nested grids (Figure 1b). The full structure (Figure 1a) permits complex data resulting from a series of deformation events such as earthquakes to be carried, with each event or component of an event described through a ggxfGroup.

5.2 GGXF Conventions

The metadata in the file, ggxfGroup and grid headers is presented as key-value pairs where the key is a fixed string defined through the 'GGXF Conventions' (Annex B) and the value may be a number, a string, a list, or further nested key-value pairs.

5.3 Geodetic content type

The GGXF file header identifies the type of content in the file using the keyword *content*. The *content* attribute is mandatory.

The content types listed in Table 2 are frequently encountered geodetic content. For these the value of the *content* attribute should be the content identifier defined in the GGXF Conventions (Annex B) Table B.3.

Table 2 — Content type

Content type
(coordinate offsets): Cartesian2dOffsets Cartesian3dOffsets geocentricTranslations geographic2dOffsets geographic3dOffsets verticalOffsets
deviationsOfThe Vertical
(geoid and height correction models): geoidModel hydroidModel (tidal surface)
velocityGrid
deformationModel

Most of these content types are associated with coordinate operation methods used to apply the data from the grid file in a coordinate operation. They are consistent with methods documented in geodetic parameter registries such as EPSG [3] and the ISO Geodetic Registry [4].

If the content of the file is not one of those given in the GGXF Conventions, producers should use "producerDefinedContent" as the value for *content* and provide a *producerDefinedContentCitation*.

Applications that implement GGXF files are not obliged to handle producer-defined content.

If the content has wide geodetic applicability, the producer should request the OGC Coordinate Reference System standards working group to consider revising the GGXF Conventions to include this content type.

Three further metadata attributes are also mandatory in a GGXF file: *title*, *abstract* and *filename*. The *abstract* attribute may be used for elaborating the file content.

5.4 Interpolation CRS

A GGXF grid is constructed in any 2D or higher-dimension spatial coordinate reference system that is compliant with ISO 19111, for example in a geographic (latitude and longitude) or projected (easting and northing) CRS. This CRS is called the Interpolation CRS. For a GGXF file containing multiple grids, all grids must be constructed within the same Interpolation CRS.

Note: The Interpolation CRS should not be confused with the source and target CRSs used in coordinate operations utilising the interpolated geodetic data. For some coordinate operations they may be related, for example for 2-dimensional ellipsoidal coordinate offsets the Interpolation CRS may be the source CRS. Other coordinate operations may have an Interpolation CRS completely unrelated to source or target CRS, for example offsets between two one-dimensional vertical CRSs will require a 2D Interpolation CRS.

The Interpolation CRS applicable to the file is identified in the GGXF file header through a well-known text (WKT) string. This format optionally permits the inclusion of a reference to a definition of the CRS in a register of geodetic parameters, from which further metadata regarding the CRS may be obtained. If there is a conflict between the well-known text description and the information derived through the external registry, the WKT description prevails.

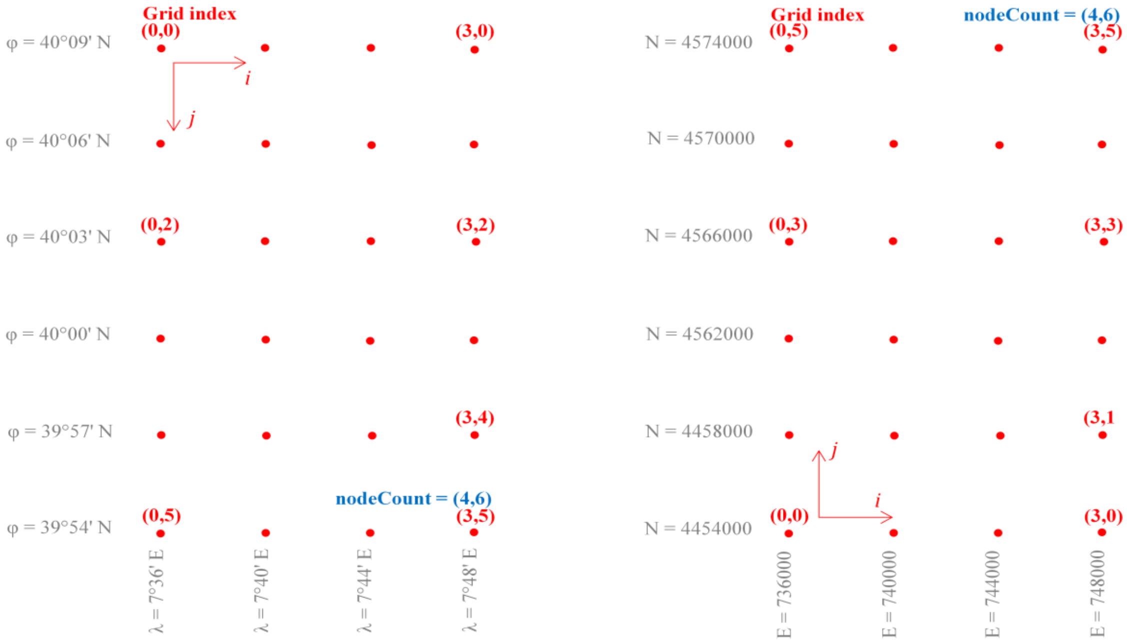
For many geodetic purposes the Interpolation CRS will be geographic 2D. Then:

Recommendation 1	When the Interpolation CRS is geographic, if it is desired to make the spacing of latitude and longitude node intervals similar in linear distance, make the node spacing in angular units in longitude ($\Delta\lambda$) approximately equal to $\Delta\varphi/\cos(\varphi_m)$ where $\Delta\varphi$ is the node spacing in latitude and φ_m is the latitude of the middle of the grid.

A geographic CRS is unsuitable for grids covering very high latitudes, and in polar areas a grid constructed in an appropriate projected CRS may be more suitable. When the Interpolation CRS is not geographic, the location of the grid in latitude and longitude terms can be identified only after conversion of Interpolation CRS coordinates.

5.5 Grid array

A GGXF grid is an array of nodes arranged in *i* columns by *j* rows (and, for a 3-dimensional grid, by *k* planes). Each node of the grid is identified by integer coordinates; the first column and row (and in the 3D case, plane) have an index value of 0 and other index values indicate distance along the axis. The node spacing along each axis in the Interpolation CRS must be constant. There is no requirement for the node spacing to be equal across different axes. Figure 2a illustrates a 2-dimensional GGXF grid of order 6 rows x 4 columns where the Interpolation CRS is geographic, the spacing in latitude is 3 arc-minutes and the spacing in longitude is 4 arc-minutes. In this example the grid indexing uses a graphic display paradigm, with grid node coordinates starting in the top left corner at (0,0) and increasing by 1 for each node. Figure 2b illustrates a 2-dimensional GGXF grid of order 6x4 where the Interpolation CRS is projected and the spacing on both axes is 4 km. In this example the grid node (0,0) is in the bottom left corner of the grid. In the grid header the maximum extent of each axis is given as a *NodeCount*, shown in blue. NodeCount = (maximum index) + 1.



(a) Interpolation CRS is geographic 2D
Grid origin in top left

(b) Interpolation CRS is projected 2D
Grid origin in bottom left

Figure 2 — Relationship of GGXF grid to Interpolation CRS

Performance for searching within a file may be impacted by the shape of the boundary of the grid. In a GGXF file 2D grids must be rectangular and 3D grids must be cuboid, that is, the minimum and maximum coordinates of a grid axis shall have equal coordinate differences in the axis units. In Figure 2a, due to the convergence of the meridians the distance along the parallel nearer to the pole is less in linear units than the distance along the parallel closer to the equator. But the grid is constructed in an Interpolation CRS in degrees, and in angular units the longitude differences of 12' at the minimum and maximum grid extent are equal: this grid is rectangular in the Interpolation CRS.

5.6 Conversion between grid array indices and Interpolation CRS coordinates

The indices (*i,j[,k]*) on each axis of the grid take an integer value from 0 to (*nodeCount* - 1) where *nodeCount* is the maximum number of nodes along that axis (see Figure 2). For each grid in the GGXF file, the conversion from that grid's node indices (*i,j[,k]*) to the file's Interpolation CRS coordinates (X,Y[,Z]) is defined by an affine transformation. In matrix form for a 2D grid:

$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} A_1 & A_2 & A_0 \\ B_1 & B_2 & B_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

Note: Here X and Y are the first and second coordinates in the Interpolation CRS tuple and should not be mistaken for axis abbreviations or implication of positive direction of the axes.

The elements of the middle matrix are derived from the coordinates of the corners of the Interpolation CRS envelope and the distance between nodes in Interpolation CRS units. Their values depend upon the corner of the grid that is the grid origin and the positive direction and axis order of the Interpolation CRS axes: see Annex D.1.

The affine transformation in conjunction with the maximum node count along each axis of the grid defines the extent (envelope) of the grid expressed in the Interpolation CRS. Indices are a measure of distance from the grid origin. The maximum node count indicates the total number of nodes along the axis. The number of inter-node intervals on the axis to be used in calculating grid extent in the Interpolation CRS is (*nodeCount* - 1).

5.7 Multiple grids

A ggxfGroup defines the value of a set of parameters by interpolating values from one or more grids of data. While these logically form one single grid it is often advantageous to use multiple grids. This may be because the area being described is an irregular shape, or because a subset of the area covered by the grid needs a finer grid spacing than the rest to provide the required accuracy of interpolated values. GGXF supports this by allowing a ggxfGroup to be described by multiple grids in a hierarchical structure, as shown in Figure 3a.

In this structure the ggxfGroup contains three top level grids A, D, and J. These are called the root grids of the group. Root grids have no parent grid. Grids A and D include child grids which provide more detailed information. For grid A these are grids B and C, and for grid D they are grids E and F. Grid E itself has children G and H which provide more detailed information about it. Parent-child relationships are identified in the structural view of Figure 3a by arrows leading from each parent to its children. In the YAML and netCDF implementations of GGXF, the parent-child relationship is defined by the file structure, not by spatial extent or by explicit attributes.

A child grid can have only one parent grid. Child grids having the same parent are termed siblings. The root grids of the GGXF group are also termed siblings – the ggxfGroup is their common parent. In this example in Figure 3a, A, D, and J are siblings, B and C are siblings, E and F are siblings, and G and H are siblings.

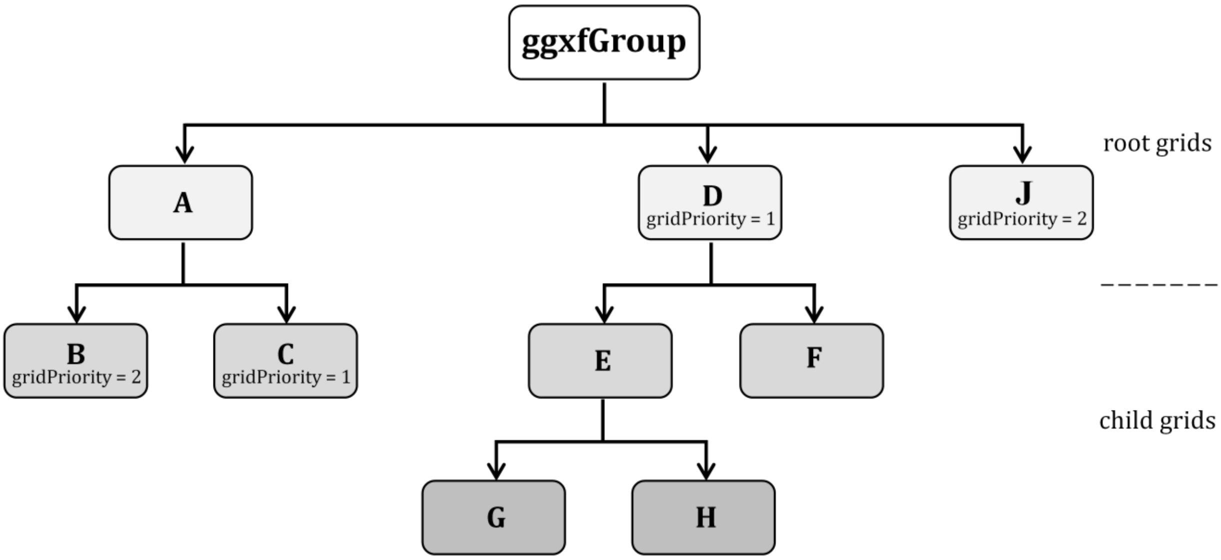


Figure 3a — Grid structure – Structural view

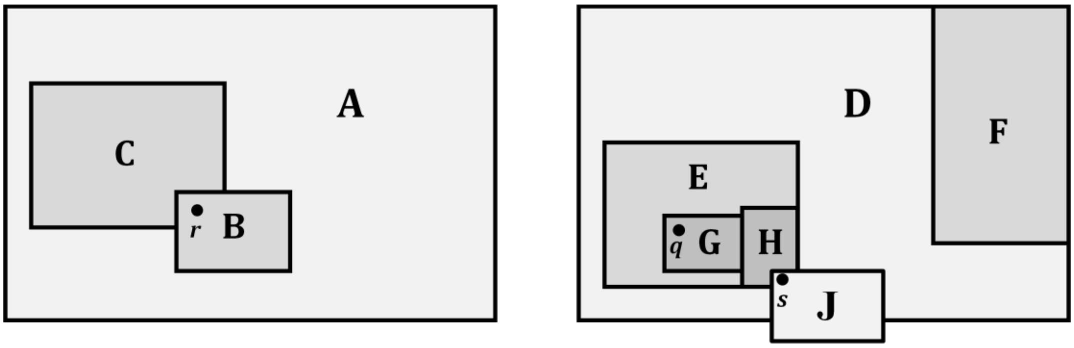


Figure 3b — Grid structure – Spatial view

Each grid has a rectangular spatial extent in the Interpolation CRS. Figure 3b shows how these grids are organised spatially. Child grids are providing detail of their parent grid so it follows that they must be spatially contained within their parent grid. Grid J cannot be a child of grid D because its extent is not fully contained within the extent of grid D. Sibling grids may intersect. In this example the root grid siblings D and J intersect, and the child grid siblings B and C intersect. Child grid siblings E and F do not intersect, nor do siblings G and H[1].

When the parameters are calculated at a specific location, only one of the grids is used. This will be one of the grids that spatially includes that location. Child grids always take priority over their parent. For example, at location *r* in the area where grids B and C intersect, grid A will not be used to evaluate the parameters as its children B and C take priority. With multiple levels of nesting the most deeply nested grid takes priority. In Figure 3b, location *q* within the extent of grid G also falls within the extent of grids D and E but, because of the parent-child structure, for a location such as *q* grids D and E are not used for evaluation.

There may be more than one such child grid at the location being evaluated. For example, the grids A, B, and C all contain location *r* within the area in which grids B and C intersect. In order to choose between intersecting sibling grids such as B and C, GGXF requires that each intersecting sibling grid

[1] Adjacency along a common boundary, such as grids G and H, is not considered to be an intersection.

has an integer attribute called *gridPriority* and for each of the intersecting siblings this must be a different value. Higher grid priority values have higher priority of use. In the example of a point such as *r* in the area where B and C intersect, grid B has gridPriority 2, and grid C has gridPriority 1, so grid B will be used. gridPriority values across different families are not required to be unique.

An intersecting sibling with the highest priority and all of its children takes priority over an intersecting sibling with the lowest priority and all of its children. For example, consider a point *s* within grid J and also within the spatial extents of grids D, E, and H. Grids D and J are intersecting siblings (both are root grids) so both must have a gridPriority attribute value. Grid J has gridPriority 2 and grid D has gridPriority 1, so grid J with the higher gridPriority is used in preference to grid D and also in preference to D's child grid E and its grandchild grid H. Should grid J itself have child grids (not the case here), these would be used in preference to J.

In a scenario in which grid H is required to be the grid with highest priority and so be used in preference to grid J where these grids overlap, simply giving grid H a high gridPriority within the grid structure in Figure 3a will have no effect, because priority of the oldest generation takes precedence. Changing the priority of grid H can only be achieved by changing the grid structure. One rearrangement that would achieve this is to make grid H a root grid and then give it a higher priority than both D and J, as shown in Figure 3c.

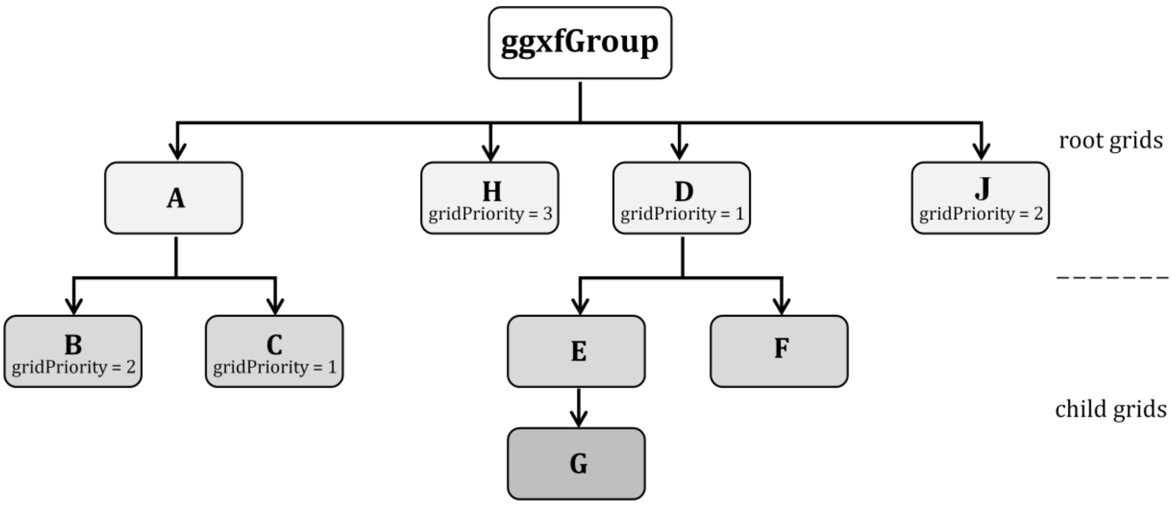


Figure 3c — Grid structure – Alternative structural view

Grid priority only applies between overlapping siblings. There is no requirement for grids that are not intersecting siblings to have the gridPriority attribute. Grid A is a sibling of intersecting sibling grids D and J (and in the alternative structure in Figure 3c, also of grid H). Grid A does not intersect any of its siblings so does not require a gridPriority value. Should it be given one (e.g. for consistency in symmetry) it will not be used. In the structural diagrams in Figures 3a and 3c, grid priority is shown for the grids for which it is a mandatory attribute – in Figure 3a intersecting sibling grids D and J, B and C, in Figure 3c intersecting sibling grids D and J and H, B and C.

The rearrangement of structure from that in Figure 3a to that in Figure 3c without any change in the spatial extent of the grids shown in Figure 3b demonstrates that spatial containment does not necessarily imply that the grid is a nested child. It is the structure of the files, not spatial extent, that controls nesting.

In many practical situations grids are constructed to not overlap, because overlapping can cause issues with continuity across the grid boundaries. None of examples E.1 through E.6 in Annex E require the use of gridPriority as they contain no intersecting sibling grids. A practical example of the application of grid priority is shown in Annex E example E.7.

A common requirement of parameter values interpolated from a ggxfGroup is that they are mathematically continuous across the spatial extent of the ggxfGroup. Where a ggxfGroup contains multiple grids there will be boundaries across which the interpolation of the ggxfGroup transitions from one grid to another. To ensure continuity, on these boundaries the parameter values interpolated from the grid on one side of the boundary should be the same as those interpolated from the grid on the other side.

Recommendation 2	Where there is a transition from one grid being used for interpolation to another, to ensure continuity the parameter values interpolated from the grid on one side of the boundary should be the same as those interpolated from the grid on the other side.

5.8 Parameters

5.8.1 General

In this document a parameter is one of a set (or tuple) of values being carried in the GGXF file at grid nodes. The number of parameters in the tuple is unlimited.

The *parameters* in the GGXF file are declared in the file header. The sequence in which these parameters are listed in the file header is significant, indicating the order in which they are given at each grid node.

The declaration includes attributes of the parameter. The attributes of *parameterName*, *unitName* and *unitSiRatio* are mandatory. Two further attributes, *sourceCrsAxis* and *noDataFlag*, are conditional. The attributes of a parameter optionally may be extended to include the minimum and maximum values of the parameter in the file, the uncertainty measure for the parameter and *groupAdditionMethod* which for parameters that are split across multiple ggxfGroups defines how their values are combined. Within a GGXF file the attributes of a parameter (units, etc.) are unchanged across all grids in the file.

In a GGXF file having multiple ggxfGroups, each ggxfGroup may have a *gridParameters* attribute specifying a subset of all parameters in the ggxfGroup, together with their order in the ggxfGroup. Within each ggxfGroup, every grid node must have the same tuple of parameters.

5.8.2 Parameter name

Each parameter must have a *parameterName*. This is the identifier for the parameter. GGXF Conventional keywords for the identifiers of those parameters supporting the content types listed in Table 2 are given in Annex B.3 Table B.4.

5.8.3 Parameter unit

Each parameter must have its units declared. The GGXF Conventions in Annex B.3 Table B6 define unit identifier (*unitName*) and ratio to SI base unit (*unitSiRatio*) for units frequently encountered in geodetic data. Units other than these may be used, but when parameters have units listed in the GGXF Conventions then the unit ID and its SI Ratio exactly as given in the Conventions should be used. If unitSiRatio is given for a unit included in the Conventions and the ratio values differ, the value in the Conventions takes precedence.

Recommendation 3	When parameters are in units documented in the GGXF Conventions, the unit ID and ratio to SI standard unit as given in the Conventions should be used.
Recommendation 4	Use arc-minutes or arc-seconds as the unit for parameters that are sexagesimal divisons of a degree.

5.8.4 Source CRS Axis

The parameter attribute *sourceCrsAxis* is included to assist applications implementing GGXF file content in a coordinate operation. It identifies the coordinate in the source CRS to which the parameter should be applied through a zero-based sequence number corresponding to the source CRS axis order. For all parameters that are variables in a coordinate operation, *sourceCrsAxis* must be included in the parameter declaration.

Example 1: In a GGXF file with content type = geographic2dOffsets in which parameters latitudeOffset and longitudeOffset are to be applied to a source CRS in which the first axis is geodetic latitude and the second axis is geodetic longitude, the sequence values for sourceCrsAxis for the latitudeOffset and longitudeOffset parameters are 0 and 1 respectively.

Example 2: In a GGXF file with content type = deformationModel in which parameters displacementEast, displacementNorth and displacementUp are to be applied to a source CRS in which the first axis is geodetic latitude, the second axis is geodetic longitude and the third axis is ellipsoidal height, the sequence values for sourceCrsAxis for the displacementEast, displacementNorth and displacementUp parameters are 1, 0 and 2 respectively.

Example 3: In a GGXF file with content type = geoidModel in which the first axis is geodetic latitude, the second axis is geodetic longitude and the third axis is ellipsoidal height, the value of the attribute sourceCrsAxis for the parameter geoidHeight is 2.

The *sourceCrsAxis* attribute is not applicable for content types that do not directly support coordinate operations, for example content type = deviationsOfTheVertical.

5.8.5 Missing data values

Where data within a grid is missing, this document requires the declaration of a 'no data' value attribute to the relevant parameter. The *noDataFlag* value is one that cannot be mistaken for a genuine value of the parameter. The value used will depend upon the data values present in the file. If all grids in the file have all nodes completely populated, the *noDataFlag* is not required.
However, having nodes with no data values causes difficulties in interpolation around the node. Different implementations may give different results. This can be avoided by producers populating missing data with estimated values.

Recommendation 5	To assist implementations reading the GGXF file, data producers are discouraged from using the no data flag and instead to populate all nodes of a grid with estimated values, giving extrapolated values a high uncertainty.
Recommendation 6	To cover an irregularly-shaped area and avoid a large number of nodes with no data, use multiple root grids. Figure 4 illustrates a hypothetical arrangement.

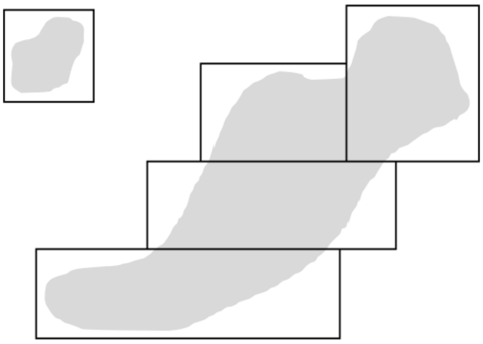


Figure 4 — Multiple grids covering an irregular area

Recommendation 7	When interpolating grids values, implementations should raise an exception, return a no-data value, or otherwise alert users to calculations which require interpolating a no-data value.

5.8.6 Node coordinates as parameters

This document permits the tuple of values at a grid node to include as parameters the coordinate values of the node (referenced to the Interpolation CRS). The GGXF conventions (Annex B) use the keyword *node** for these parameters, for example *nodeLatitude*. Whilst this information may be very useful to human readers of a text extract of grid content, implementations reading a GGXF file do not require these values to be explicitly given – they are implied through the grid node indices and affine transformation coefficients. Including these parameters increases the size of the GGXF file and, particularly for large grids, may lead to degraded implementation performance. They may however be useful during the production of GGXF text files using the extenal file option (see 6.2.3).

Recommendation 7	To minimise file size, coordinates of the grid nodes should not be included as parameters in grids in a GGXF binary file. However, producers creating GGXF binary files using a GGXF YAML text file with ggxf-csv external grid files are encouraged to include node coordinates in the ggxf-csv grid files to allow software to validate the row and column counts and the affine coefficients. These node coordinates are not required to be included in the resultant GGXF binary file.

5.8.7 Parameter uncertainty

This document does not require parameter uncertainty to be included in grid data, but if required GGXF can support doing so through two mechanisms.

a) For any parameter a complimentary parameter carrying the uncertainty of that parameter at each grid node may be given. For example, in a GGXF file with a content type of 'geographic2dOffsets' (see GGXF Conventions Annex B Table B.3), the mandatory parameters 'latitude offset' and 'longitude offset' may be accompanied by two further parameters, 'latitude offset uncertainty' and 'longitude offset uncertainty' (see GGXF Conventions Table B.4). The uncertainties are declared in the file header as parameters in the normal way.

b) A single (constant) value for the uncertainty of a parameter throughout a ggxfGroup may be declared in the ggxfGroup header. For example, in a deformation model the parameter 'displacementHorizontalUncertainty' may be declared in the file header and in a ggxfGroup header a constant value for it may be declared. The displacement horizontal uncertainty then is not explicitly defined at grid nodes, but a GGXF application will treat it as if it is defined at every grid node with the declared constant value value. This is described in 5.8.9.

For both mechanisms, the statistical measure used for the uncertainty is declared through one of the uncertainty measure attributes given in the GGXF Conventions (Annex B Table B.10).

5.8.8 ParameterSet

This optional attribute may be used for controlling the array(s) in which parameters are stored in the GGXF netCDF implementation. This is discussed further in 6.3.

5.8.9 Parameters in a ggxfGroup

5.8.9.1 Use of multiple ggxfGroups

Typically, in a GGXF file with a single ggxfGroup, every grid node will hold the ordered set of all parameters defined in the file header. For parameters that have different characteristics multiple ggxfGroups can be used. For example, the spatial characteristics of the horizontal and vertical components of a *geographic3dOffset* can be quite different, and this may be reflected by using grids having different node intervals. Because the node intervals differ, the three offsets cannot be described in a single grid without the inclusion of nodes with missing data (see 5.8.5). However a producer could put all three parameters into the same grid by interpolating the coarser grid to the node intervals of the finer grid, at the expense of increasing file size. An alternative would be for the producer to provide them in a GGXF file with two ggxfGroups, one defining the *latitudeOffset* and *longitudeOffset* through one grid and the other defining the *ellipsoidalHeightOffset* through a different grid.

A GGXF file may also have multiple ggxfGroups which define common parameters. For example, in a deformation model each ggxfGroup is characterised by its time behaviour. It may contain many ggxfGroups containing parameters *displacementEast*, *displacementNorth*, and *displacementUp*. Each group will have a different time function. See Annex E example E.5 for a deformation model GGXF file containing multiple groups in which the *gridParameters* differ.

5.8.9.2 Combining parameters from multiple groups

In a GGXF file containing multiple ggxfGroups, any parameter occurring in multiple ggxfGroups must have its values from each ggxfGroup combined to calculate the full parameter value that is used for the coordinate operation described by the GGXF file. By default, parameter values are combined by adding them together. If a ggxfGroup contains a time function then the parameter values interpolated from the grids in the ggxfGroup are multiplied by this value before they are added.

For uncertainty parameters it may be more appropriate to combine values as the root mean square of the values from each ggxfGroup. A parameter definition in the GGXF file header may include a *groupAdditionMethod* attribute which can take values *addition* (the default) or *rootMeanSquare* that can be used to specify the method.

5.8.9.3 Parameter values beyond grid extent

Each ggxfGroup in a GGXF file defines parameters within the spatial extent defined by its grids. ggxfGroups may have different spatial extents. The GGXF file supports coordinate operations at any location within the spatial extents of any of its ggxfGroups. At a location that is only within a subset of the ggxfGroups, the parameter values from the other ggxfGroups are treated as zero.

At a location that is not within the spatial extents of any ggxfGroup in the file, parameter values are undefined. The GGXF file does not support coordinate operations at that location.

Parameter values are also undefined at a location if calculating the parameter requires using a no data value in any of the ggxfGroups that apply at that location.

For example, a deformation model may contain two ggxfGroups. One represents the long term secular deformation across the entire extent of the producer's jurisdiction. The other represents the coseismic deformation due to an earthquake and is only defined in the region affected by that event. At a point outside the extent of the earthquake ggxfGroup, the deformation is defined solely by the secular deformation ggxfGroup. At this point he earthquake ggxfGroup makes no contribution – its contribution to each parameter is zero.

5.8.9.4 gridParameters

In a GGXF file with multiple ggxfGroups, only a subset of the parameters declared in the file header may be relevant to a particular ggxfGroup. That ggxfGroup can include a *gridParameters* attribute specifying the subset of parameters represented at each grid node, and their order. If *gridParameters* is not specified, then the grids in the ggxfGroup will hold the set of parameters declared in the file header, in the order in which they are defined.

Recommendation 9	When all parameters given in the file header are used in the ggxfGroup, the *gridParameters* attribute should be omitted from the ggxfGroup header.

Annex E examples E.1 through E.4 contain a single ggxfGroup and omit the *gridParameters* in the ggxfGroup header.

5.8.9.5 constantParameters

A ggxfGroup may also include a *constantParameters* attribute defining constant values for one or more parameters in that ggxfGroup. This facilitates a smaller file size as the values are omitted from the set of node parameters but is treated as if the constant value at each node of each grid.

A *constantParameters* attribute can be specified only in a ggxfGroup that also contains a *gridParameters* attribute. A parameter cannot be specified in both the *gridParameters* and *constantParameters* attributes. The *gridParameters* and *constantParameters* attributes together define which parameters are represented in the ggxfGroup. However, the attributes of these parameters (units, etc.) are as defined in the file header.

5.8.9.6 Time functions applied to parameters

A ggxfGroup may include an attribute *timeFunctions* that defines a time evolution of the parameter values interpolated from the grids in the ggxfGroup. This is currently supported only for deformationModel GGXF files, but in principle other content types could include time functions.

The time function may be evaluated from one or more components; these are evaluated separately and added together to calculate the total time function. For example, a time function for post-seismic displacements after an earthquake may include an exponential and a logarithmic component.

If a ggxfGroup includes a time function, then a time must be specified to evaluate the parameters. The time function evaluates a scalar value at that time. The parameter values interpolated at a location (or defined through the ggxfGroup *constantParameters* values) are multiplied by this time function value to give the value of those parameters at that location and time.

5.9 Data extent

5.9.1 General

The spatial and, if relevant, temporal extent of all data in a GGXF file is given in the file header as discovery metadata.

For both technical and legal reasons the extent over which the file content should be used may be less than the full extent of data in the file. This document requires the description of the applicability of the file content. This may (and often will) differ from the spatial extent of the gridded data.

Example: A single grid produced by a US agency covering both Alaska and the conterminus lower 48 states will include a good proportion of Canada, but the US data should not be used there. Canada has its own equivalent dataset, and in Canada that dataset should be used.

The description uses the Extent provisions from ISO 19115-1. A text description and a geographic bounding box are mandatory, temporal extent is conditional and geographic bounding polygon and vertical extent are optional.

The extent of each grid in the file is not described this way. It is described through the affine transformation outlined in 5.6 and in Annex D.1 in conjunction with the node maxima attributes; see Annex E.1.

However, there may be circumstances where for discovery purposes it would be useful to give the geographic extent of individual ggxfGroups or grids, for example, when the Interpolation CRS is projected rather than geographic. In this document this is permitted, using the *contentBox* attribute, but the capability is not expected to be widely used.

5.9.2 Geographic bounding box

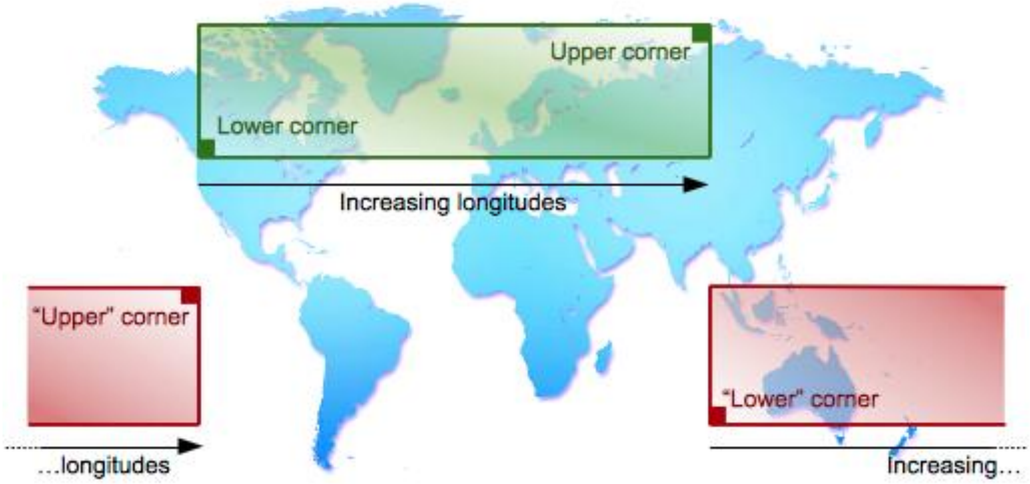


Figure 5 — Bounding boxes

ISO 19115-1 requires that bounding box longitudes are in the range −180 =< λ =< +180 degrees, positive eastwards from the prime meridian. In this document the interpretation made is "start from lower bound longitude λ_1, then move through increasing longitude values until we reach λ_2". The relative values of *westBoundLongitude* and *eastBoundLongitude* describe whether the bounding box crosses the 180° meridian. In the −180° =< λ =< +180° range, if $\lambda_1 < \lambda_2$ then the grid does not cross the 180° meridian (green box in Figure 5), but when $\lambda_1 > \lambda_2$ then the grid crosses the 180° meridian (red bounding box in Figure 5). The example in Annex E.5 includes a bounding box crossing the 180° meridian, the other examples in Annex E show the more frequent case of bounding boxes not crossing the 180° meridian.

5.9.3 Extent description

A free text description of the geographic, vertical and/or temporal extent of file contents must be given in the file header. When a grid or grids cover an irregular area, the valid application of the file content may be significantly reduced from that described by the bounding box, for example within national limits, and the text description may be used to indicate this. If relevant, the description of main subdivisions may be included.

5.9.4 Geographic bounding polygon

Geographic bounding box is a coarse description of the applicability of data. It will often include areas around its periphery where data is not applicable. A more refined description can be made through a bounding polygon. However, detailed polygons can be large and working with them complex. This document permits the inclusion of a polygon in the file header but it is not mandatory to provide it, and if a polygon is provided it is not mandatory for implementations to utilise it. A polygon may consist of any number of exterior and interior linear rings. If provided, a polygon must be represented through Well-known text (WKT) and adhere to the geometry requirements of ISO 19125-1. If the polygon crosses the 180° meridian, polygon vertex longitude coordinates should be continuous across the 180° meridian in whichever of the ranges (−180° =< λ =< +360°) and (−360° =< λ =< +180°) has minimal extent beyond the range (−180° =< λ =< +180°).

5.9.5 Temporal validity

The temporal validity describes the period over which the parameters in the GGXF file are applicable, given in the file header as a start date and/or an end date, or as a start epoch and/or an end epoch.

The distinction between date and epoch is the data type: epoch is given as a number of years represented by a real number, date is given as a date/time using ISO 8601 syntax.

Temporal validity is 'discovery metadata' for the entire dataset. Deformation model datasets also define time function for each ggxfGroup that describe how that the displacement defined in the ggxfGroup applies at a particular time.

5.9.6 Vertical extent

Limits on vertical validity of file content may be given in the file header. Minimum and maximum values are in the context of the positive direction of the vertical CRS axis (heights or depths). The vertical CRS is identified through Well-Known Text.

5.10 Miscellaneous metadata

5.10.1 Interpolation method

This document does not specify the method for interpolation of the gridded data but allows the file provider to recommend a method. Including this information guides implementers; it contributes to ensuring that implementations output results consistent with the expectations of grid producers. Bi-linear interpolation of a 2D grid and tri-linear interpolation of a 3D grid will suffice for most geodetic purposes. The interpolation method is an optional attribute and when not present bi-linear interpolation of a 2D grid and tri-linear interpolation of a 3D grid should be assumed.

The interpolation methods given in Table B.10 are frequently encountered in geodetic content and these identifiers should be used where appropriate, but this list is not exclusive and other interpolation methods may be recommended. If a producer wishes an interpolation method not in this table to be used by implementers, the *userDefinedMethodName* and a *userDefinedMethodFormulaCitation* to its formulae should be provided.

Recommendation 10	The method that is recommended to be used for interpolation of the grid(s) in a ggxfGroup should be given in the ggxfGroup header.

5.10.2 Nominal operation accuracy

operationAccuracy is a single value indicating the typical error of target coordinates derived through application of this grid content assuming no errors in source coordinates. It is discovery metadata for the file content as a whole. Coordinate Transformation and Point Motion Operation descriptions compliant with ISO 19111 are required to include this information.

This operation accuracy parameter is independent of the uncertainty of a parameter in the grid. Uncertainty of any parameter in the grid may be included as an additional parameter for the grid; see 5.8.4.

Recommendation 11	A GGXF file with content supporting coordinate transformation should indicate the nominal operation accuracy value applicable to the data in the file.

5.10.3 Citation

The GGXF file header should include metadata giving details of the party responsible for producing the file. The elements of the CI_Citation and CI_Responsibility classes from ISO 19115-1 – *Metadata fundamentals* should be used for this. To include a Digital Object Identifier (DOI), use the CI_Citation attribute *identifier* and then the MD_Identifier attributes *code* and *codeSpace*=DOI.

Recommendation 12 The producer and distributer of the file, file date of issue, and if applicable a Digital Object Identifier (DOI) for the file content should be included in the GGXF file header metadata.

5.10.4 License

For the protection of both supplier and consumer of GGXF data, license information that clearly defines the data copyright (if any is declared) as well as use restrictions, citation obligations, export limitations, etc. should be included in the header metadata. There are numerous existing licenses that may be suitable to reference as well as organization specific licenses that may need to be used. Custom or generic licenses may be referenced by URL using a *licenseURL* attribute in the header metadata. Alternatively, the entirety of the license text may be included using the *license* attribute in the header metadata.

Recommendation 13 Metadata describing the terms under which the GGXF file is distributed (license information) should be included in the file header.

5.10.5 Permanent tide

If earth tides are significant to the geodetic parameters described in the GGXF file, the tide system used may be indicated in a ggxfGroup header.

5.10.6 Check data

Coordinates and/or parameter values at one or more points within the of the GGXF file may be included in the GGXF file header to be used as verification data using the *checkPoints* attribute. The check points can be at grid nodes or within grid cells; they should be within the *contentApplicabilityExtent* of the data in the GGXF file. Parameters are identified using the *parameterName* given in the file header. Coordinates referenced to the source CRS (or if no source CRS is given in the file header, referenced to the interpolation CRS) are used to identify the location of the checkpoint. For GGXF file content supporting coordinate operations, the coordinates in the target CRS or target coordinate epoch act as test points for verification of the values in the GGXF file and their application through the coordinate operation. Check data is included in examples E.3 and E.5.

5.10.7 Comments

Miscellaneous comments about the data in the file may be made using the keyword *comment*. A single *comment* may be made in each element of the embedded structure: one comment in the file header, one in each ggxfGroup header, each grid header, each parameter definition, each time function definition, etc.

Note: Multiple comments within one element are invalid YAML and netCDF; whether and how they are handled will depend upon the reader.

6 Encoding

6.1 Introduction

This document provides for two encodings:

i) text;
ii) binary.

The GGXF YAML text file is envisaged primarily as an intermediate stage used by file producers in the assembling of grid data. It is not intended for public distribution. The text format is both human and machine readable, but for human reading will be practical only for small grids, or small extracts of large grids. The GGXF YAML text file has an option for referencing grid data in separate files rather than including the grid data within the GGXF YAML text file. The header information in the GGXF YAML text file is then consolidated and easier to read and edit. The text file uses syntax in compliance with YAML version 1.2. YAML may be validated using various available tool such as https://codebeautify.org/yaml-validator.

The GGXF binary file includes header records and all grids; it does not permit reference to external grid files. This document requires netCDF to be used as the carrier for the GGXF binary encoding.

Recommendation 14	Producers are encouraged to publish data using the GGXF binary format rather than the GGXF text format.

Specific provisions for the YAML and netCDF encodings are given below.

6.2 YAML Encoding

6.2.1 Grid data

The GGXF YAML format supports two alternative methods of handling grids and, if relevant, content extent polygon. For small grids and polygons the data can be included directly in the GGXF YAML file. This is a direct mapping of the GGXF structure to YAML. When the text file contains grid (and if relevant polygon) data and does not reference external files, there is a 1:1 mapping between text file content and the GGXF binary file.

However, some YAML readers and text editors may not handle very large grids. To cater for this, in place of the grid (or polygon) data the GGXF YAML format may include a reference to an external file. This may also be useful for producers generating grids with software that doesn't generate GGXF YAML files. When the text file contains references to external grid or polygon files, there is a near complete mapping between the GGXF text and binary formats; the difference is that the text format is the GGXF headers plus references to external files.

Recommendation 15	For large grids producers are encouraged to have these grids in external files rather than directly in the YAML text file.

The purpose of a text GGXF file with external grid data is to allow specific software to compile the dataset to a netCDF binary GGXF file for publication. This specification defines one supported format of external grid data files (see 6.2.3.2). A GGXF software provider may choose to support other external grid data formats. A producer cannot assume that a GGXF text file referencing other external formats will be readable by software other than that for which it is written

6.2.2 YAML grid data syntax

In the GGXF YAML format, all parameters at a node are given sequentially with each value separated by a comma and space (YAML will ignore the space).

In the YAML syntax the grid data is represented as a nested array structure. The structure consists of an array in which each element represents a row in the grid. Within the row array, each element is an array representing the columns in the row, and each element of that array is an element representing the parameter values at each node in the column.

Example: a 3 x 5 grid with 2 parameters per node, each node having a parameter value in green and in blue (i.e. total of 30 values):

```
[[[1.00, -2.70],[1.20, -2.50],[1.40, -2.30],[1.60, -2.10],[1.80, -1.90]],
 [[1.21, -2.74],[1.40, -2.52],[1.65, -2.31],[1.83, -2.13],[2.00, -1.92]],
 [[1.40, -2.78],[1.80, -2.53],[2.00, -2.33],[2.10, -2.14],[2.20, -1.93]]]
```

In this structure, to access the highlighted second parameter of the fourth column of the third row in the grid would use an index [2][3][1] (noting that YAML arrays are indexed from 0). When the YAML grid data is loaded into the multidimensional grid structure used to represent GGXF grids it is indexed on [row(j),column(i),parameter], so the same element is indexed [3,2,1].

GGXF allows the YAML structure to be flattened if it suits a data producer. In particular if there is only one parameter at each node then it may be represented as a single value, rather than an array of one value. For example, a geoid height can be shown as a simple value 1.3, rather than an array [1.3].

More generally the brackets around each set of parameter value and around the data for each row may also be omitted, provided it is done consistently in the entire grid. Bracketing around only the parameters at each node, or around only each row of nodes, is not permitted as it may be ambiguous as to which it is.

Example: the grid above may be written as an array of values as follows. The order of elements in the grid is not changed, it is simply an array of numbers.

```
[1.00,-2.70,1.20,-2.50,1.40,-2.30,1.60,-2.10,1.80,-1.90,1.21,-2.74,1.40,-2.52,1.65,
 -2.31,1.83,-2.13,2.00,-1.92,1.40,-2.78,1.80,-2.53,2.00,-2.33,2.10,-2.14,2.20,-1.93]
```

In this structure the highlighted element would be accessed by index [27]. However, when it is loaded into the multidimensional structure its index is unchanged and is [3,2,1]. The software reading the YAML file is responsible for correctly loading the YAML structure based on the grid shape (number of rows and columns) and the number of parameters at each node.

6.2.3 YAML external grid data

6.2.3.1 General

The GGXF binary file format requires grid data to be accompanied with header data in a single file. However, the GGXF YAML text file format optionally permits the grid data to be held in one or more external files referenced from the GGXF YAML file. This feature is designed for data producers to simplify building GGXF data sets from other software. GGXF software can be used to convert the GGXF YAML format to a GGXF netCDF file for distribution. GGXF YAML files referencing external grid data files should not be used for distribution.

The external grid files are referenced by the *dataSourc*e keyword in the GGXF YAML grid specification. It is used in place of the *data* keyword. *dataSource* identifies the external location of the grid data (typically through a file name or url), its format, and possibly a subset of data that are to be read from the file through a set of key-value pairs. This should include *dataSourceType* which identifies the format of its grid data. The *dataSource* attribute is specific to the text file reader implementation and cannot be used in the GGXF binary format.

When there is a change to the sequence of parameters between the GGXF binary file (as described through the GGXF YAML headers) and the external file, it is made by the application creating the GGXF binary file. In accordance with the recommendations in 5.8.8, any parameters with the identifier *node** should not be included in the translation into the GGXF binary file.

This specification describes one simple text file format for external files (6.2.3.2). However the GGXF YAML file may reference external grids in other formats (6.2.3.3).

6.2.3.2 GGXF external grid text file format

GGXF supports one format for external grid files - the *ggxf-csv* text file format - and one set of keywords to reference this. The ggxf-csv format is based on the commonly used comma separated value (CSV) file format but it allows the file to use either comma or space or tab as the separator. The *dataSourceType* for a GGXF external grid data file is "*ggxf-csv*" with an additional key *separator* identifying the character used to separate values in the file. The ggxf-csv text file format is defined in the GGXF Conventions in Annex B Table B.17.

The content of these ggxf-csv files is logically part of the GGXF YAML text file that references them. The metadata such as units given in the GGXF YAML text file applies to these parameters, except that the sequence of parameters in the ggxf-csv file may differ from the sequence defined in the GGXF YAML file, and the ggxf-csv file may contain node coordinates as additional parameters. The inclusion of node coordinates in the ggxf-csv file is not a requirement but they may be used by software for validating the grid affine coefficients.

See E.1.3 for examples of GGXF YAML files with externally-referenced grid data in the ggxf-csv text format supported in this document.

6.2.3.3 User-defined external grid file format

There is no restriction on the file format for external grid files referenced through *dataSource*. They may be in a producer-specific format. The definition of *dataSourceType* other than ggxf-csv and any additional keys specific to that source type is outside the scope of GGXF. If the data source is a single file then it is recommended that the key *gridFilename* is used to identify the file. Software implementing a custom data source may define other keys in the *dataSource* attribute to describe how the grid data is provided from the external data source. Translating such grids into the GGXF binary format will require a custom extension to an application reading the GGXF YAML file. This is outside the scope of GGXF. The external file content need not conform to GGXF requirements but if this is the case then the custom software will need to include any data conversion necessary so that in the GGXF binary file the grid and the metadata in the GGXF YAML headers match.

See E.5 and the GitHub link in E.7 for examples of a GGXF YAML files with grid data in a referenced external file using user-defined attributes for *dataSource*.

6.3 netCDF Encoding

6.3.1 Introduction

The Unidata network Common Data Form (netCDF) is used as the carrier for the GGXF binary encoding. There are several versions of netCDF: GGXF uses netCDF-4.

The netCDF user guide recommends using the netCDF-CF (climate and forecasting) conventions. However, the CF-conventions used for defining the grid *domain* do not apply to GGXF files. GGXF files do not use netCDF coordinate variables for defining grid axis coordinates. Instead, in GGXF the domain coordinate reference system and its relationship with the file's grid coordinates are defined solely by the interpolationCrsWkt and affineCoeffs attributes.

This GGXF specification does have a small subset of attributes in common with the netCDF-CF conventions; examples include some discovery metadata in the file header using ACDD attributes (see 6.3.2) and using netCDF data packing attributes *scale_factor* and *add_offset* (see 6.3.5.4).

6.3.2 Relationship of GGXF Conventions to ACDD

The Attribute Convention for Data Discovery (ACDD) defines a list of netCDF global attributes recommended for describing a netCDF dataset to metadata discovery systems. ACDD is encouraged in some non-geodetic communities utilising netCDF. Software tools are available to use these attributes for extracting metadata from netCDF datasets and exporting to metadata formats such as Dublin Core and ISO 19115. To facilitate the application of these software tools to a GGXF file, when implementing a GGXF file in netCDF the mapping of attribute names of certain parameters shall follow that in Table B.14.

Note that Table B.14 does not include all ACDD attributes. Applications reading a GGXF netCDF file may ignore any attributes that are encountered but which are not found in the GGXF Conventions.

Recommendation 16	When reading a GGXF binary file any attributes encountered that are not in the GGXF Conventions may be ignored.

6.3.3 GGXF netCDF structure

The netCDF-4 structure has the following main components:

- netCDF group - a structured data object which may contain other groups.
- netCDF variable - an array attribute of a group with an arbitrary number of dimensions. The size of the array is defined by netCDF dimensions in its direct or indirect parent groups.

- netCDF attribute - a named attribute which containing either a single value or a list of values. netCDF attributes may be assigned to netCDF groups and to netCDF variables.
- netCDF dimension - a special integer attribute of a netCDF group that is used to define the dimensions of a variable. It is defined in a group and may be referenced in that group and all its children and their descendents.

netCDF groups are analogous to directories in a file system. Each netCDF file has a unique "root" group. Each group is identified by a name. The root group is named "/". Each group is identified by the path "<parent_path_name>/<group_name>", except that if the parent is the root group then the path is "/<group_name>".

The structural elements of a GGXF file are located in a netCDF file as follows:

- The GGXF file header is the netCDF root group at *"/"*
- A ggxfGroup header is a netCDF group that is a direct child of the root group at *"/<ggxfGroupName>"*
- GGXF grid headers, including the grid data, are netCDF groups that are either children of a GGXF group header or of another GGXF grid header following the nesting structure of the grids in each ggxfGroup.
  - Top level (root) grids in a nested grid structure area are located at *"/<ggxfGroupName>/<rootGridName>"*.
  - First generation child grids are located at *"/<ggxfGroupName>/<rootGridName>/<childGridName>".*

6.3.4 Representation of structured data

6.3.4.1 General

Attributes in the file header, ggxfGroup header, and grid header are represented as netCDF attributes of the corresponding netCDF group except for

i) the *iNodeCount*, *jNodeCount*, and *kNodeCount* parameters which are stored as dimensions in the netCDF group holding the GGXF grid.
ii) structured parameters which are discussed in the following subclause.

Attributes which are simple strings or numeric values are stored as netCDF attributes of the same type.

Attributes which are lists of strings or numbers (such as “gridParameters”, “affineCoefficients”) are stored as netCDF list attributes.

6.3.4.2 Structured data

GGXF also defines attributes that are structured. A structured attribute is one which holds a set of name/pair value or an ordered list of values. Each of the constituent values may itself be a set of name/pair values or an ordered list of values. Examples of structured data include the parameters attribute in the file header which has several attibutes (name, units, etc) and the timeFunctions attribute of a deformationModel content type.

These GGXF structured attributes are represented using multiple attributes in the netCDF group, one for each string or numeric value held within the structured attribute. The netCDF attributes are named using a naming convention that appends a string encoding the structure to the GGXF attribute name.

- GGXF attributes containing name/value pairs are replaced by an attribute for each pair named by appending "." and the pair name to the name of the attribute that holds it.
- GGXF attributes containing a list of values are replaced by
  i) an attribute defining the number of elements in the list, named by appending ".count" to the name of the attribute that holds the list;
  ii) an attribute for each element of the list named by appending a zero-based sequence number ".0", ".1", ... to the name of the attribute that holds the list.

The following example illustrates this for the *parameters* attribute of the GGXF file header. This example has four parameters, each with the (minimum mandatory) three attributes of name, unit identifier and unit SI ratio.

```
:parameters.count = 4 ;
:parameters.0.parameterName = "latitudeOffset" ;
:parameters.0.unit = "arc-second" ;
:parameters.0.unitSiRatio = 4.84813681109536e-06 ;
:parameters.1.parameterName = "longitudeOffset" ;
:parameters.1.unit = "arc-second" ;
:parameters.1.unitSiRatio = 4.84813681109536e-06 ;
:parameters.2.parameterName = "latitudeOffsetUncertainty" ;
:parameters.2.unit = "metre" ;
:parameters.2.unitSiRatio = 1.0 ;
:parameters.3.parameterName = "longitudeOffsetUncertainty" ;
:parameters.3.unit = "metre" ;
:parameters.3.unit = 1.0 ;
```

The following example illustrates this for deformation model time functions. The timeFunctions attribute of a ggxfGroup contains a list of values each of which is a set of name/pair values. This example includes a step function and a ramp function, these having different attributes.

```
:timeFunctions.count = 2 ;
:timeFunctions.0.functionType = "step" ;
:timeFunctions.0.eventDate = "2009-07-15T00:00:00Z" ;
:timeFunctions.0.functionReferenceDate = "2010-01-01T00:00:00Z" ;
:timeFunctions.1.functionType = "ramp" ;
:timeFunctions.1.startDate = "2009-07-15T00:00:00Z" ;
:timeFunctions.1.endDate = "2009-10-01T00:00:00Z" ;
:timeFunctions.1.functionReferenceDate = "2010-09-01T00:00:00Z" ;
:timeFunctions.1.scaleFactor = 0.3 ;
```

6.3.5 Representation of grid data

6.3.5.1 ParameterSet

In the GGXF logical model each GGXF grid contains a single array containing the tuple of parameters defined at each grid node.

In the GGXF netCDF implementation this is represented by one or more NetCDF variables, each holding the grid node values for a subset of parameters. A variable holding two or more GGXF parameters is referred to as a "vector variable". The parameter attribute *parameterSet* is used to define the netCDF variable to which a GGXF parameter is assigned. When two or more GGXF parameters are assigned the same value for their parameterSet attribute, they are combined into a single netCDF vector variable. For example, if three GGXF parameters displacementNorth, displacementEast, displacementUp all have parameterSet value “displacement”, they are all combined into a single netCDF variable called 'displacement'. If four GGXF parameters latitudeOffset, latitudeOffsetUncertainty, longitudeOffset and longitudeOffsetUncertainty are assigned parameterSet attributes of offset, offsetUncertainty, offset and offsetUncertainty respectively, the GGXF parameters latitudeOffset and longitudeOffset will be combined in a NetCDF vector variable called offset whilst the GGXF parameters latitudeOffsetUncertainty and longitudeOffset will be combined in a second netCDF vector variable called offsetUncertainty.

The *parameterSet* attribute is optional. If it is not defined, then the *parameterName* is used as the name of the netCDF variable holding the parameter. For example, if a parameterSet value was not given to either of two GGXF parameters latitudeOffset and longitudeOffset, they are held in two separate netCDF variables called 'latitudeOffset' and 'longitudeOffset' respectively.

Recommendation 17	Consider defining different parameter sets for parameters and their uncertainties, for example 'offset' and 'offsetUncertainty'. Applications not needing to utilise the uncertainty data can then skip the offsetUncertainty variable.
Recommendation 18	In creating the name of a parameter set, avoid 'count' as this is may create a conflict with its use in the definition of structured data and in netCDF vector variable naming.

6.3.5.2 Parameter sequence

The order of parameters in the netCDF variables is defined in the file header. This may be overridden by the optional *gridParameters* attribute in the ggxfGroup containing the grid. *gridParameters* may define a subset of all parameters which are in the group and/or define a new sequence.

Note that where the grid data is used in a transformation between a source CRS and a target CRS, the parameters in the netCDF file are not necessarily in the same order as the axes in the source CRS to which the grid data is to be applied. The source CRS axis corresponding to each parameter should be defined in the GGXF parameter definitions in the GGXF file header - see 5.8.5.

6.3.5.3 netCDF variable size

The size of a netCDF variable is defined by netCDF dimension attributes. Each variable will have a dimension for each interpolation coordinate axis. These netCDF dimensions are called iNodeCount, jNodeCount, (kNodeCount for 3-dimensional interpolation grids) and are defined in the netCDF group representing the grid. The values correspond to the node count.

Vector variables have an additional netCDF dimension representing the number of parameters at each grid node. This is defined in the netCDF group containing the grid. The name of this dimension is arbitrary.

Recommendation 19	It is recommended that the dimension defined in netCDF group for the number of parameters in a grid variable should be the parameterSet name appended with “Count”. For example, the dimension of the parameterSet "displacement" should be named “displacementCount".

Each netCDF variable in a grid therefore will have dimensions (iNodeCount,jNodeCount) for variable respresenting a single GGXF parameter, or (iNodeCount,jNodeCount,*set*Count) for a variable representing a vector of GGXF parameters in the parameterSet *set*.

6.3.5.4 Data packing and numeric type

In a GGXF netCDF file the variables holding grid data can be of any numeric type. There is no requirement that all the variables in a GGXF netCDF file are of the same numeric type.

The variables may have netCDF conventional attributes *scale_factor* and *add_offset* to define a scale factor and offset used to facilitate packing the data into a smaller numeric type. The *scale_factor* and *add_offset* attributes are assigned individually to each netCDF variable in a GGXF file. The values need not be the same for each variable.

When scaled data is written, the producing application should first subtract the offset and then divide by the scale factor. The units of a variable should be representative of the unpacked data.

When reading a GGXF netCDF file, after the data values of a packed variable have been read, they are to be multiplied by the *scale_factor* and have *add_offset* added to them. If both attributes are present, the data is scaled before the offset is added.

6.3.5.5 No data value

Each parameter in a GGXF file may have a *noDataFlag* defined that is used at grid nodes where the value of the parameter is missing or not defined. Any parameter that has missing values somewhere in the GGXF file must have the *noDataFlag* attribute defined.

In a netCDF GGXF file this value is not used in the variables that hold grid data. Instead, each variable may have a *missing_value* attribute defined. This is a conventional netCDF attribute. It is not used in a GGXF YAML file. *noDataFlag* is in domain of the parameter, whereas *missing_value* is in the domain of the netCDF variable. It is used to represent any missing value in the variable. The *missing_value* attribute must be of the same numeric type as the data in the netCDF variable on which it is defined, which may be packed data.

Each variable in a GGXF netCDF file may use a different value for the *missing_value* attribute. If all the parameter values are defined at every node in the netCDF variable, then the variable does not require a *missing_value* attribute.

Recommendation 20	If a grid has missing values a producing application should determine a suitable *missing_value* for each grid data variable it creates. It should use this value in place of the noDataFlag in the variable. The *missing_value* attribute of a variable should be either larger than the largest actual packed data value or smaller than the smallest actual packed data value in the variable.
Recommendation 21	A reading application encountering values matching the *missing_value* attribute in a variable should identify and treat the corresponding unpacked values as undefined.

Annex A
(normative)

Conformance requirements

A.1 Core requirements

The GGXF core requirements are those that shall be satisified by both data providers and data users. To conform to the core requirements of this document a GGXF file shall as a minimum include requirements given in Table A.1. Where appropriate, text in italic font gives the keyword from the GGXF Conventions. All requirements below are identified by partial URIs which are relative to the base

http://www.opengis.net/spec/ggxf/1.0/

The identifiers given in this document use camelCase for clarity, but they are not case sensitive.

Table A.1 — GGXF core requirements

Identifier	Requirement
req/core/conventions	**GGXF Conventions** A GGXF file shall adhere to the GGXF Conventions. The GGXF file header shall begin with the GGXF Convention identifier *version* to which the file conforms. The name and version number for files conforming to this document is "GGXF-1.0".
req/core/fileHeader	**File header** A GGXF file shall contain a header describing the data in the file. The header shall have three elements: (i) a file header containing metadata applicable to the whole file describing the file content and its provenance, (ii) for each ggxfGroup in the file, a ggxfGroup header containing metadata applicable to that ggxfGroup, and (iii) for each grid in the file, a grid header containing metadata applicable to that grid.
req/core/groupIdentifier	**ggxfGroup identifier** Each ggxfGroup shall have an identifier which is unique within the file. It shall be given in the ggxfGroup header. The ggxfGroup identifier shall be specified throught the GGXF Conventions identifier *ggxfGroupName* and be the first attribute in the ggxfGroup header block. It shall be a Unicode Identifier.
req/core/gridIdentifier	**Grid identifier** Each grid shall have an identifier which is unique within the file. It shall be given in the grid header and be specified through the GGXF Conventions identifier *gridName*. It shall be a Unicode Identifier.
req/core/content	**Content type** A GGXF file shall contain a single content type. This shall be specified in the file header by the *contentType* attribute.

Identifier	Requirement
req/core/fileMetadata	**Title, abstract and file name** The GGXF file header shall include attributes *title*, *abstract* and *filename*.
req/core/interpolationCrs	**Interpolation CRS** All grids in a GGXF file shall be constructed in the same Interpolation CRS. This Interpolation CRS shall be any 2D or higher-dimension coordinate reference system that is compliant with ISO 19111. The GGXF file header shall include a description of the Interpolation CRS using well-known text (WKT), using the GGXF Conventions identifier *interpolationCrsWkt*. The well-known text shall be compliant with ISO 19162 (OGC 18-010). If the WKT string contains a reference to an external registry describing the CRS, if there is a conflict between the well-known text description and the information derived through the http URI, the WKT description shall prevail.
req/core/sourceTargetCrs	**Source and target CRS** (Conditional) When the GGXF content is to be used in a coordinate operation, the GGXF file header shall include a description of the Source CRS and Target CRS in well-known text (WKT) using the GGXF Conventions identifiers *sourceCrsWkt* and *targetCrsWkt* respectively. The well-known text shall be compliant with ISO 19162 (OGC 18-010). If a WKT string contains a reference to an external registry describing the CRS, if there is a conflict between the well-known text description and the information derived through the http URI, the WKT description shall prevail.
req/core/geogExtent	**Geographic extent** The GGXF file header shall include a description of the geographic applicability of file content using the GGXF Conventions identifiers "*contentApplicabilityExtent.boundingBox.southBoundLatitude*", "*contentApplicabilityExtent.boundingBox.westBoundLongitude*", "*contentApplicabilityExtent.boundingBox.northBoundLatitude*", "*contentApplicabilityExtent.boundingBox.westBoundLongitude*" and "*contentApplicabilityExtent.extentDescription*".
req/core/boundingPolygon	**Bounding polygon** (Conditional) The GGXF file header may include a refined description of the applicable geographic extent using the GGXF Conventions identifier *contentApplicabilityExtent.boundingPolygon*. If provided, the polygon representation shall be in Well-known text (WKT) and adhere to the geometry requirements of ISO 19125-1, the vertices shall be in geographic coordinates (latitude and longitude), and if the polygon crosses the 180° meridian the polygon vertex longitude coordinates shall be continuous across the 180° meridian in either one of the ranges (−180° =< λ =< +360°) and (−360° =< λ =< +180°), preferably that which has minimal extent beyond the range (−180° =< λ =< +180°).

Identifier	Requirement
req/core/temporalExtent	**Temporal extent** (Conditional) If one or more grids in a GGXF file have limits on their temporal applicability, the full temporal extent of the file content shall be given in the file header through either a) *startDate* and/or *endDate* or b) *startEpoch* and/or *endEpoch* If start date is given and end date is missing, the temporal validity of the data in the file continues into the future from the start date. If start epoch is given and end epoch is missing, the temporal validity of the data in the file continues into the future from the start epoch. If end date is given and start date is missing, the temporal validity of the data in the file applies before and up to the end date. If end epoch is given and start epoch is missing, the temporal validity of the data in the file applies before and up to the end epoch. If temporal extent is not given, there is no constraint on temporal validity of the data in the file.
req/core/producer	**Data producer metadata** (Conditional) When citation or responsible party metadata is included in the file header, for any attributes not explicitly given in the GGXF Conventions the ISO 19115-1 UML role names shall be used as keywords.
req/core/DOI	**Digital Object Identifier** (Conditional) When a Digital Object Identifier (DOI) is included in file header metadata, the CI_Citation attribute *identifier* and then the MD_Identifier attributes *code* and *codeSpace*=DOI shall be used.
req/core/grid	**GGXF grid** A GGXF grid shall be an array of i columns by j rows (and, for a 3-dimensional grid, by k planes), where $i,j,[k] > 1$. The indices of the grid nodes shall start at zero and increase by 1 for each column, row, and plane. A 2D grid shall be rectangular in its Interpolation CRS and a 3D grid shall be cuboid in its Interpolation CRS. Note: 'rectangular' and 'cuboid' mean that the lengths of opposite edges of the grid have the same distance in Interpolation CRS units, for example an arc-rectangle with north and south sides along different parallels of latitude but both extending 20 degrees of longitude. On any one axis of a GGXF grid, the spacing of nodes in the Interpolation CRS shall be constant. For a grid of dimensions i columns by j rows and with p parameters at each node there shall be $i.j.p$ values in the block of data for the grid.

Identifier	Requirement
req/core/affineCoeffs	**Affine Transformation** For each GGXF grid the relationship of the grid indices to Interpolation CRS coordinates shall be described in the grid header through affine transformation coefficients (*affineCoeffs*).
req/core/nodeCount	**Node Count** For each axis of the grid, the maximum node count shall be given in the grid header (*iNodeCount*, *jNodeCount* [*kNodeCount*].
req/core/interpolationMethod	**Grid interpolation method** GGXF file producers should specify the required grid *interpolationMethod* in the ggxfGroup header information. GGXF file user software shall apply the interpolation method specified in the ggxfGroup header. If no interpolation method is specified, bilinear shall be assumed.
req/core/nestedGrid	**Nested grids** (Conditional) For ggxfGroups with nested grids, the extent of a child grid shall be contained within the extent of its parent grid. They may share a common external boundary. The parent-child relationships within the grids belonging to a ggxfGroup shall be defined through the ggxfGroup structure. To evaluate ggxfGroup parameters in nested grids, a child grid shall be used in preference to its parent.
req/core/gridPriority	**Grid priority** (Conditional) When sibling grids intersect, different integer *gridPriority* values for each intersecting sibling shall be given in the grid header. Where sibling grids intersect, to evaluate the ggxfGroup parameters a sibling grid with a higher gridPriority and all its direct or indirect child grids shall be used in preference to the sibling with lower gridPriority and all its direct or indirect child grids.
req/core/param/attributes	**Parameters and their attributes** All parameters in the GGXF file shall be defined in the GGXF file header. For each parameter, its *parameterName*, *unitName* and the *unitSiRatio* shall be given. If unitSiRatio is given for a unit included in the GGXF Conventions and its value differs from that in the Conventions, the value in the Conventions shall take precedence. This list of parameter attributes may be extended to include *minimumValue*, *maximumValue*, *uncertaintyMeasure*, and the *parameterSet* in which the parameter is grouped in a data array.
req/core/uncertaintyMeasure	**Uncertainty Measure** (Conditional) When uncertainty estimates are included as parameters, the measure for uncertainty as given in the GGXF Conventions shall be included in the parameter definition.

Identifier	Requirement
req/core/param/sequence	**Parameter Sequence** The sequence in which the parameters are listed shall be the sequence of the parameters at each grid node. This sequence may be amended for a ggxfGroup. In a ggxfGroup containing only a subset of the parameters declared for the whole GGXF file, the subset and its sequence shall be identified in a *groupParameters* attribute in the ggxfGroup header.
req/core/param/count	**Parameter Count** Each node of all grids within a ggxfGroup shall contain the same tuple of parameters. For a grid of dimensions *i* columns by *j* rows and with *p* parameters at each node there shall be *i.j.p* values in the block of data for the grid.
req/core/param/mandatory	**Mandatory parameters** As a minimum, those parameters given in the GGXF Conventions as being mandatory for the content type shall be included in the tuple.
req/core/param/sourceCrsAxis	**Source CRS Axis** (Conditional) For parameters used in a coordinate operation, the sequence value of the *sourceCrsAxis* to which the parameter is applied shall also be included.
req/core/param/missingData	**Missing Data** (Conditional) When a parameter is missing a value at any grid node in the GGXF file, the value for a *noDataFlag* specific to the parameter shall also be included. A *noDataFlag* value shall be one that cannot be mistaken for a genuine parameter value. The flag may be omitted if a value is provided at every node of every grid in the file at which the parameter applies.
req/core/param/nodeCoords	**Node coordinates** (Conditional) If coordinate values for nodes are included as parameters, they shall match the values calculated from the node indices using the affine transformation.
req/core/param/constantValue	**Parameter with Constant Value** (Conditional) In a ggxfGroup containing a parameter with a constant value throughout all grids in the ggxfGroup, the parameter and its constant value shall be identified in the ggxfGroup header. Any parameter declared to have a constant value in a ggxfGroup shall be omitted from the tuple of parameters at grid nodes in the grids in the ggxfGroup. Any parameter declared to have a constant value in a ggxfGroup shall be treated as if were defined at grid nodes.
req/core/permanentTide	**Permanent Tide** (Conditional) If relevant to the ggxfGroup content, the permanent *tideSystem* used shall be indicated in the ggxfGroup header.

A.2 Requirements for mapping GGXF schema to a GGXF YAML text file

Requirements for mapping the GGXF schema into the GGXF YAML text file format are given in Table A.2. Where appropriate, text in italic font gives the keyword from the GGXF Conventions. All requirements are identified by partial URIs which are relative to the base http://www.opengis.net/spec/ggxf/1.0/. The terms 'mapping' and 'collection' in Table A.2 are as defined in the YAML specification.

Table A.2 — GGXF YAML text file requirements

Identifier	Requirement
req/yaml/version	**Version** GGXF YAML text files shall adhere to the requirements of YAML version 1.2.
req/yaml/filename	**File name** GGXF YAML text files shall have the file name extension ".yaml".
req/yaml/structure	**File structure** A GGXF YAML text file shall be structured as a mapping containing the GGXF file header attributes A GGXF YAML text file shall include an attribute "*ggxfGroups*" containing a collection of mappings. Each ggxfGroup shall be represented by a mapping in the *ggxfGroups* collection. Each ggxfGroup shall contain an attribute "*grids*" containing a collection of mappings. Each grid header in the ggxfGroup shall be represented by a mapping in the "*grids*" collection. Each mapping representing a grid may contain an attribute "*childGrids*" containing a collection of mappings. Each child grid of a grid shall be represented by a mapping in the "*childGrids*" collection.

Identifier	Requirement
req/yaml/gridData	**Grid data** For each grid, the GGXF YAML file shall contain either an attribute "*data*" containing the array of the parameter values in the grid within the YAML text file, or an attribute "*dataSource*" defining an external data source from which the array of parameter values shall be read. A grid "*data*" attribute shall contain an entry for each interpolation axis value and for each parameter in the grid. The data is represented as a nested array structure of dimensions $[n_j][n_i][n_p]$ or $[n_k][n_j][n_i][n_p]$ where n_i, n_j, and n_k are the number of columns, rows, and planes in the grid respectively, as defined in the grid header, and n_p is the number of parameters at each node as defined in the ggxfGroup that contains the grid. The YAML structure may be flattened by combining any or all of the grid axes provided that the order of numeric values is not changed. For example a structure of dimension $[n_j][n_i][n_p]$ could be flattened to dimensions $[n_j][n_i \times n_p]$ or $[n_j \times n_i \times n_p]$. Regardless of the flattening of the nested array structure the order of the parameter values in the YAML representation is unaltered. The location of the pth parameter at node (i,j) shall be given by the formula $p + n_p \times (i + n_i \times j)$ where the indices i, j and p are all zero based indices. In a three dimensional grid with indices(i, j, k) the formula shall be: $p + n_p \times (i + n_i \times (j + n_j \times k))$
req/yaml/gridBracketing	**Grid value bracketing** Parameter values for the whole grid shall be within one pair of square brackets with all parameters at a node given sequentially and each value separated by a comma. Optionally, additional square bracketing may be inserted around (a) the set of parameters at each node and (b) each row of nodes.

Identifier	Requirement
req/yaml/ggxf-csv	**External grid value file** A ggxf-csv external grid data text file shall contain one header line followed by one line for each grid node in the order specified in req/yaml/gridData. The line containing the parameters for node (i,j) shall be: $2 + i + n_i \times j$ Note: 2 is added to the line number as there is 1 header line, and line numbers are conventionally 1 based – the first line is 1. In a three dimensional grid the containing the parameters for node (i,j,k) shall be at line number: $2 + i + n_i \times (j + n_j \times k)$ The values in the header line shall be the grid node parameter identifiers from the GGXF Conventions and shall define the order of parameter values in the subsequent lines. The set of identifiers shall include the grid node parameters defined in the ggxfGroup within which the external grid file is referenced and may also include node coordinate identifiers. The values on each subsequent line shall be the parameter values at a grid node in the order defined by the header line. Missing parameter values in the grid shall be identified by the *noDataFlag* value defined for the parameter in the GGXF file header. Each line shall have the same number of values. Values shall be separated by the character given through the *separator* key and optionally may be additionally padded with spaces. Each line shall terminated by the line feed (new line) character (U+000A). The line feed may be immediately preceded by a carriage return character (U+000D).

A.3 Requirements for mapping GGXF schema to a GGXF netCDF binary file

Requirements for mapping the GGXF schema into the GGXF netCDF binary file format are given in Table A.3. Where appropriate, text in italic font gives the keyword from the GGXF Conventions. All requirements are identified by partial URIs which are relative to the base http://www.opengis.net/spec/ggxf/1.0/

Table A.3 — GGXF netCDF binary file requirements

Identifier	**Requirement**
req/netcdf/filename	**File name** GGXF netCDF binary files shall have the file name extension ".ggxf".
req/netcdf/structure	**File structure** The GGXF file header attributes shall be held in the netCDF root group. Each ggxfGroup shall be represented by a netCDF group that is a child group of the root group. Every child group of the root group represents a ggxfGroup. Each grid shall be represented by a netCDF group. Each root grid of a ggxfGroup is represented by a netCDF group that is a child group of that representing the ggxfGroup. Each child grid of a grid is represented by a netCDF group that is a child group of the netCDF group representing its parent grid.
req/netcdf/attributes	**Attributes** Attributes of the file, ggxfGroup, and grid headers shall be represented as netCDF attributes of the corresponding netCDF group, except: • The node count attributes of a grid (*iNodeCount*, *jNodeCount*, *kNodeCount*) shall be held as netCDF dimensions of the netCDF group containing the grid. • The *ggxfGroupName* parameter of a ggxfGroup is held as the name of the netCDF group representing the ggxfGroup • The *gridName* parameter of a grid is held as the name of the netCDF group representing the grid GGXF attributes holding a string value, or a numeric value, or a list of string values, or a list of numeric values shall be represented by a single netCDF attribute. GGXF attributes holding more complex structured values shall be represented by multiple netCDF attributes as described in the GGXF Conventions. GGXF attributes in a netCDF file shall use the name specified in the GGXF Conventions.

Identifier	Requirement
req/netcdf/variable	**Parameter values** The array of parameter values of a grid shall be represented by one or more netCDF variables in the netCDF group representing the grid. Each variable shall be named by the *parameterSet* of the parameter it contains; if *parameterSet* is not defined then the variable shall be named by the *parameterName* of the parameter. Parameters of the ggxfGroup that have the same *parameterSet* name shall be combined into a single netCDF variable. The order of parameters in the variable shall be the order defined in the *gridParameters* attribute in the ggxfGroup containing the grid, or, if that is not defined, in the order defined in the *parameters* attribute in the GGXF file header. A grid variable containing the data for a single GGXF parameter shall have dimensions (*iNodeCount*, *jNodeCount*) or (*iNodeCount*, *jNodeCount*, *kNodeCount*). A grid variable containing data for two or more GGXF parameters shall have dimensions (*iNodeCount*, *jNodeCount*, *parameterSetCount*) where 'parameterSet' is replaced by the name of the *parameterSet*. The *parameterSetCount* value shall be defined as a dimension in the netCDF group representing the ggxfGroup containing the grid.

Annex B
(normative)

GGXF Conventions

B.1 Introduction

The GGXF Conventions are a controlled vocabulary of identifiers used in file, ggxfGroup and grid header records to describe file content. Most are key-value pairs. To conform to YAML requirements, identifiers (keywords) are case-sensitive. To aid readability, keywords formed from multiple words or abbreviations are given in upper camel case.

A consolidated list of all GGXF identifiers is given in Table B.1

Primary keywords and definitions are given in Table B.2. In some cases reference is made to supplementary Tables B.3 through B.11 which give controlled vocabularies for attributes. In addition, for citation information GGXF uses the UML role names from the ISO 19115 (Metadata) Citation and Responsible Party classes as keywords; selected example keywords are shown in Table B.12. Mapping of GGXF attributes to netCDF is given in Tables B.13 through B.16. These tables collectively form the GGXF Conventions. In each table the identifiers are listed alphabetically.

Clause	Table	Content
B.2 principle keywords	Table B.2	header attribute keywords
B.3 supplementary keywords	Table B.3	content identifier keywords
	Table B.4	parameter identifier keywords
	Table B.5	content tidal surface identifier keywords
	Table B.6	unit identifier keywords and ratio to SI standard unit
	Table B.7	time function identifier keywords
	Table B.9	interpolation method identifier keywords
	Table B.10	uncertainty measure identifier keywords
	Table B.11	permanent tide identifier keywords
B.4 citation keywords	Table B.12	example citation and responsible party attributes from ISO 19115-1
B.5 mapping GGXF to NetCDF	Table B.13	Categories for NetCDF attribute mapping
	Table B.14	GGXF file header attributes to netCDF variables
	Table B.15	ggxfGroup header attributes to netCDF variables
	Table B.16	GGXF grid header attributes to netCDF variables
B.6 external grid files	Table B.17	Definitions for optional external simple text file

To propose an extension to the GGXF Conventions, submit a change request to the OGC CRS SWG.

Table B.1 — GGXF conventions: keywords

GGXF keyword	Defined in Table	GGXF keyword	Defined in Table
1CED	B.10	deformationModel	B.3
1CEE	B.10	degree	B.6
1CEP	B.10	depthOffset	B.4
1DRMS	B.10	depthOffsetUncertainty	B.4
1SE	B.10	deviationEast	B.4
1SEP	B.10	deviationEastUncertainty	B.4
2CED	B.10	deviationNorth	B.4
2CEE	B.10	deviationNorthUncertainty	B.4
2CEP	B.10	deviationsOfTheVertical	B.3
2DRMS	B.10	digitalObjectIdentifier	B.2
2SE	B.10	displacementEast	B.4
3SE	B.10	displacementEastUncertainty	B.4
abstract	B.2	displacementHorizontalUncertainty	B.4
acceleration	B.7	displacementNorth	B.4
affineCoeffs	B.2	displacementNorthUncertainty	B.4
arc-minute	B.6	displacementUp	B.4
arc-second	B.6	displacementUpUncertainty	B.4
bicubic	B.9	eastBoundLongitude	B.2
bilinear	B.9	eastingOffset	B.4
biquadratic	B.9	eastingOffsetUncertainty	B.4
boundingBox	B.2	ellipsoidalHeightOffset	B.4
boundingPolygon	B.2	ellipsoidalHeightOffsetUncertainty	B.4
Cartesian2dOffsets	B.3	endDate	B.2
Cartesian3dOffsets	B.3	endEpoch	B.2
CD	B.5	epoch	B.2
centimetre	B.6	eventDate	B.2
checkPoints	B.2	eventEpoch	B.2
childGrids	B.2	exponential	B.7
citation	B.10	extentDescription	B.2
cm/yr	B.6	extentTemporal	B.2
comma	B.17	extentVertical	B.2
comment	B.2	extentVerticalCrsWkt	B.2
constantParameters	B.2	extentVerticalMaximum	B.2
content	B.2	extentVerticalMinimum	B.2
contentApplicabilityExtent	B.2	filename	B.2
contentBox	B.2	foot	B.6
conventionalTideFree	B.11	frequency	B.2
cyclic	B.7	functionReferenceDate	B.2
data	B.2	functionReferenceEpoch	B.2
dataSource	B.2	functionType	B.2
dataSourceType	B.2	geocentricTranslations	B.3
date	B.2	geocentricXOffset	B.4

GGXF keyword	Defined in Table
geocentricXOffsetUncertainty	B.4
geocentricYOffset	B.4
geocentricYOffsetUncertainty	B.4
geocentricZOffset	B.4
geocentricZOffsetUncertainty	B.4
geographic2dOffsets	B.3
geographic3dOffsets	B.3
geoidHeight	B.4
geoidHeightUncertainty	B.4
geoidModel	B.3
ggxfGroupName	B.2
ggxfGroups	B.2
ggxfVersion	B.2
ggxf-csv	B.17
gridFilename	B.2
gridName	B.2
gridParameters	B.2
gridPriority	B.2
grids	B.2
groupAdditionMethod	B.2
HAT	B.5
heightOffset	B.4
heightOffsetUncertainty	B.4
HHWLT	B.5
HW	B.5
hydroidHeight	B.4
hydroidHeightUncertainty	B.4
hydroidModel	B.3
hyperbolicTangent	B.7
iNodeCount	B.2
instantaneous	B.11
interpolationCrsWkt	B.2
interpolationMethod	B.2
interpolationMethodCitation	B.2
ISLW	B.5
jNodeCount	B.2
kNodeCount	B.2
LAT	B.5
latitudeOffset	B.4
latitudeOffsetUncertainty	B.4
license	B.2
licenseURL	B.2
LLWLT	B.5
logarithmic	B.7

GGXF keyword	Defined in Table
longitudeOffset	B.4
longitudeOffsetUncertainty	B.4
LW	B.5
m/s	B.6
m/yr	B.6
meanTide	B.11
MHHW	B.5
MHW	B.5
MHWST	B.5
milliarc-second	B.6
millimetre	B.6
mas/yr	B.6
meanCrust	B.11
metre	B.6
MLLW	B.5
MLLWST	B.5
MLW	B.5
MLWST	B.5
mm/yr	B.6
missing_value	B.2
MSL	B.5
noDataFlag	B.2
nodeDepth	B.4
nodeEasting	B.4
nodeEllipsoidalHeight	B.4
nodeGeocentricX	B.4
nodeGeocentricY	B.4
nodeGeocentricZ	B.4
nodeHeight	B.4
nodeLatitude	B.4
nodeLongitude	B.4
nodeNorthing	B.4
nodeSouthing	B.4
nodeWesting	B.4
noInterpolation	B.9
northBoundLatitude	B.2
northingOffset	B.4
northingOffsetUncertainty	B.4
operationAccuracy	B.2
parameterCheckValues	B.2
parameterMaximumValue	B.2
parameterMinimumValue	B.2
parameterName	B.2
parameters	B.2

GGXF keyword	Defined in Table
parameterSet	B.2
parameterValue	B.2
ppb	B.6
ppb/yr	B.6
ppm	B.6
ppm/yr	B.6
producerDefinedContent	B.3
producerContentTypeCitation	B.2
publicationDate	B.2
radian	B.6
rad/s	B.6
ramp	B.7
RMSE	B.10
scaleFactor	B.2
second	B.6
separator	B.17
sourceCoordinateEpoch	B.2
sourceCrsAxis	B.2
sourceCrsCoordinates	B.2
sourceCrsWkt	B.2
southBoundLatitude	B.2
southingOffset	B.4
southingOffsetUncertainty	B.4
space	B.17
startDate	B.2
startEpoch	B.2
step	B.7
tab	B.17
targetCoordinateEpoch	B.2
targetCrsCoordinates	B.2
targetCrsWkt	B.2
tidalSurface	B.2
tideFree	B.11
tideSystem	B.2
timeConstant	B.2

GGXF keyword	Defined in Table
timeFunctions	B.2
title	B.2
tricubic	B.10
trilinear	B.9
uncertaintyMeasure	B.2
unitName	B.2
unitSiRatio	B.2
unitType	B.2
unity	B.6
unity/s	B.6
userDefinedMethodExample	B.2
userDefinedMethodFormula	B.2
userDefinedMethodFormulaCitation	B.2
velocity	B.7
velocityEast	B.4
velocityEastUncertainty	B.4
velocityGrid	B.3
velocityNorth	B.4
velocityNorthUncertainty	B.4
velocityUp	B.4
velocityUpUncertainty	B.4
velocityX	B.4
velocityXUncertainty	B.4
velocityY	B.4
velocityYUncertainty	B.4
velocityZ	B.4
velocityZUncertainty	B.4
version	B.2
verticalOffsets	B.3
westBoundLongitude	B.2
westingOffset	B.4
westingOffsetUncertainty	B.4
year	B.6
zeroTide	B.11

B.2 GGXF Conventions: principal keywords

Table B.2 — GGXF conventions: header attribute identifier

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
abstract	Brief narrative summary of content of the GGXF file or ggxfGroup.	characterString	
affineCoeffs	Ordered list of coefficients A_1, A_2, A_0, B_1, B_2 and B_0 of the 2D affine parametric transformation used to convert the grid indices to the Interpolation CRS coordinates and vice versa. For a 3D grid the order in the sequence of coefficients shall be A_1, A_2, A_3, A_0, B_1, B_2, B_3, B_0, C_1, C_2, C_3 and C_0. Refer to clause 5 and Annex D.1.	sequence	
boundingBox	Geographic bounding box describing an extent through keys *southBoundLatitude*, *westBoundLongitude*, *northBoundLatitude* and *eastBoundLongitude*.	set	*southBoundLatitude*, *westBoundLongitude*, *northBoundLatitude*, *eastBoundLongitude*
boundingPolygon	List of points given in geographic (latitude longitude) coordinates describing an extent using the simple features geometry of ISO 19125-1, in particular through exterior linear ring(s) and optionally (for excluded enclaves) interior linear rings. For each ring the final vertex shall be the same as the first vertex.	characterString	Well-known text (WKT) syntax for geometry as defined in ISO 19125-1, Simple features.
checkPoints	Keyword in a GGXF file header to indicate the start of list of test data included to assist file users verify their reading of data at grid nodes or interpolated points. Each check point shall include *sourceCrsCoordinates*. It may also include *parameterCheckValues*. When the GGXF file *content* is used in coordinate operations the check point should also include *targetCrsCoordinates.*	set	
childGrids	Keyword in a YAML grid header to indicate the start of list of child grids contained within a parent grid.	set	

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
comment	Free-text comment that may be included in a file header, ggxfGroup header or grid header block.	characterString	
constantParameters	Keyword in a ggxfGroup header to indicate the start of a list of parameter identifiers for *parameters* in the ggxfGroup that have constant values. Each item of the list shall have two attributes, *parameterName* and *parameterValue*.	set	*parameterName*, *parameterValue*
content	Identification of the content of the GGXF file, given through a cryptic description of the coordinate operation method which utilises the data in the file.	characterString	Table B.3 — GGXF conventions: content identifier
contentApplicabilityExtent	Keyword in a GGXF file header to indicate the start of a list of extents of the technical and legal applicability of the file content. Content applicability extent may differ from the extent of the file or ggxfGroup content which is described through *contentBox*. *contentApplicabilityExtent* shall have two attributes, *extentDescription* and *boundingBox*. Optionally this list may be extended to include *boundingPolygon*, *extentTemporal* and *extentVertical*.	set	*boundingBox*, *boundingPolygon*, *extentDescription*, *extentTemporal*, *extentVertical*
contentBox	Geographic bounding box describing the extent of the ggxfGroup or grid content through *boundingBox*. Used when the interpolation CRS is not geographic. See also *contentApplicabilityExtent*.	set	*boundingBox*
data	Keyword in a GGXF YAML text file indicating the array of grid parameter values.	sequence	

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
dataSource	Keyword in a GGXF YAML text file grid definition indicating key-value pairs identifying an external file containing the grid data. *dataSource* must have one attribute *dataSourceType*. If the dataSourceType is ggxf-csv then dataSource must include attributes *gridFilename* and *separator*. Other user-specific attributes are permitted. They must be different to any keywords in these Conventions. These are implementation specific and not part of the GGXF Conventions. Note: *dataSource* is not valid in a GGXF binary file which must contain all header and grid data.	set	*dataSourceType*
dataSourceType	Identifier for the type of external file referenced in a GGXF YAML text file through *dataSource*. This document supports one value for this attribute, *ggxf-csv*; any other user-defined identifier is permitted but is not part of these Conventions.	characterString	Table B.17 — GGXF conventions: external grid file format
date	Instant in time given as a date/time data type in conformance with ISO 8601-1.	dateTime	ISO 8601-1, Representation of date and time.
digitalObjectIdentifier	Digital identifier (DOI) for the model described by the GGXF gridded data compliant with the requirements of ISO 26234.	characterString	ISO 19115-1, Metadata fundamentals, CI_Citation MD_Identifier
eastBoundLongitude	Discovery metadata describing the eastern longitude of a geographic bounding box (*contentApplicabilityExtent* and *contentBox*), expressed in decimal degrees (positive east) in the range -180 =< λ =< 180. If the geographic bounding box crosses the 180° meridian then *eastBoundLongitude* < *westBoundLongitude*, else *eastBoundLongitude* > *westBoundLongitude*.	real	-180 =< *eastBoundLongitude* =< 180 *eastBoundLongitude* ≠ *westBoundLongitude*

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
endDate	*date* at the end of a time period. If omitted the temporal extent applies at all times after the temporal extent *startDate*. For a deformation model, the time function has a constant value at and after this time.	dateTime	*endDate* > *startDate*
endEpoch	*epoch* at the end of a time period. For a deformation model, the time function has a constant value at and after this time.	real	> 0 *endEpoch* > *startEpoch*
epoch	Instant in time given expressed in the Gregorian calendar as a decimal year. EXAMPLE 2017-03-25 in the Gregorian calendar is epoch 2017.23.	real	
eventDate	*date* used as parameter used in a deformation model the time function	dateTime	
eventEpoch	*epoch* used as a parameter in a deformation model the time function.	real	
extentDescription	Discovery metadata textual description of the geographic, vertical and temporal extents of the content of the GGXF file.	characterString	
extentTemporal	Discovery metadata describing the temporal extent of the applicability of file content, expressed through *startDate and endDate.*	set	*startDate*, *endDate* {count(startDate + endDate)≥1}
extentVertical	Discovery metadata describing the vertical extent of the applicability of file content, expressed through *extentVerticalMaximum*, *extentVerticalMinimum* and *extentVerticalCrsWKT*.	set	*extentVerticalMaximum*, *extentVerticalMinimum*, *extentVerticalCrsWkt*
extentVerticalCrsWkt	Description in well-known text conformant to ISO 19162 of the vertical coordinate reference system to which the values of *extentVerticalMaximum* and *extentVerticalMinimum* are referenced.	characterString	ISO 19162, Well known text representation of coordinate reference systems.
extentVerticalMaximum	The maximum vertical extent for the GGXF file content. Whether this is a height or a depth, and its units, are defined through *extentVerticalCrsWkt*.	real	extentVerticalMaximum > extentVerticalMinimum

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
extentVerticalMinimum	The minimum vertical extent for the GGXF file content. Whether this is a height or a depth, and its units, are defined through *extentVerticalCrsWkt*.	real	extentVerticalMinimum < extentVerticalMaximum
filename	Name and extension of a GGXF file.	characterString	It is required that the extension for GGXF text files be ".yaml" and for GGXF binary files the extension be ".ggxf"
frequency	Number of cycles per year of a deformation cyclic time function.	real	
functionReferenceDate	*Date* at which a deformation time function has value zero. A constant value may be added to a time function to set it to zero at this date. Some deformation time function types require a reference date for their definition. Each time function using a reference date has its own date; deformation time functions within a single ggxfGroup may have different reference dates.	dateTime	
functionReferenceEpoch	*Epoch* at which a deformation time function has value zero. A constant value may be added to a time function to set it to zero at this epoch. Some deformation time function types require a reference epoch for their definition. Each time function using a reference epoch has its own epoch; deformation time functions within a single ggxfGroup may have different reference epochs.	real	
functionType	GGXF identifier of the type of deformation time function.	characterString	Table B.7 — GGXF conventions: time function identifier
ggxfGroupName	Identifier which is unique within the GGXF file for this ggxfGroup.	characterString	Valid Unicode identifier, unique within the GGXF file,
ggxfGroups	Keyword in a GGXF file header to indicate the start of a list of ggxfGroups contained within the file.	sequence	*ggxfGroups*
ggxfVersion	Version of the GGXF format for the file.	real	> 0

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
gridFilename	Identifier of an external file containing the grid data referenced in a GGXF text file through *dataSource*. Typically a file name or url.	characterString	
gridName	Identifier for a grid which is unique within the grids in a GGXF file.	characterString	Valid Unicode identifier, unique within the GGXF file,
gridParameters	Ordered list of the parameters represented in the tuple of values at each node of each grid in the ggxfGroup. Each element of the list shall match the *parameterName* of a parameter defined in the file header *parameters* attribute. Required only when a subset of the *parameters* declared in the file header are present at nodes in the grid(s) in the ggxfGroup, or when the order of parameters at the nodes differs from the order defined in the file header. When gridParameters is not specified, nodes throughout the ggxfGroup have all parameters in the order specified in the *parameters* file header attribute. Note: Attributes for each parameter are defined in the file header through *parameter*.	orderedSet	
gridPriority	Attribute used to indicate priority of intersecting sibling grids. Grid priority only applies between intersecting siblings. It is required for all intersecting sibling grids. All intersecting root grids in a ggxfGroup and all intersecting sibling child grids in a nested structure are intersecting sibling grids.	integer	Unique within a set of intersecting siblings.
Grids	Keyword in a YAML ggxfGroup header to indicate the start of list of grids contained within the ggxfGroup.	sequence	
groupAdditionMethod	Optional attribute of a parameter definition specifying how parameter values are combined if they are defined in more than one ggxfGroup.	characterString	*addition* (the default), *rootMeanSquare*

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
interpolationCrsWkt	Description of the coordinate reference system to which grid nodes in the GGXF file are referenced, given in well-known text conformant to ISO 19162.	characterString	ISO 19162, Well known text representation of coordinate reference systems.
interpolationMethod	Identifier of interpolation algorithm recommended by data provider for interpolating data values at locations not coincident with grid nodes.	characterString	Table B.9 — GGXF conventions: interpolation method identifier
interpolationMethodCitation	Reference for interpolation algorithm recommended by data provider for interpolating data values at locations not coincident with grid nodes.	characterString	ISO 19115-1, Metadata fundamentals, CI_Citation
iNodeCount	The number of nodes on the *i-axis* of the grid. The maximum index (equal to the number of inter-node intervals) on this axis is (*iNodeCount* - 1).	integer	> 1
jNodeCount	The number of nodes on the *j-axis* of the grid. The maximum index (equal to the number of inter-node intervals) on this axis is (*jNodeCount* - 1).	integer	> 1
kNodeCount	The number of nodes on the *k-axis* of the grid. The maximum index (equal to the number of inter-node intervals) on this axis is (*kNodeCount* - 1).	integer	> 1
license	License under which the file is distributed.	characterString	
licenseURL	URL of the license under which the file is distributed. If the text in the file does not agree with the URL reference text the text in the file shall take precedence.	characterString	
missing_value	Attribute of a netCDF variable defining the value that indicates missing or undefined data in the variable. Must be either larger than the largest actual packed data value or smaller than the smallest actual packed data value in the variable. Note: not used in a GGXF YAML file - see *noDataFlag*.	real	
noDataFlag	Attribute of a parameter in a GGXF file to indicate missing or undefined data. Must be a value that cannot be mistaken for a genuine parameter value.	real	

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
northBoundLatitude	Discovery metadata describing the northern latitude of a geographic bounding box (*contentApplicabilityExtent* and *contentBox*), expressed in decimal degrees (positive north) in the range -90 =< φ =< 90 and *northBoundLatitude* > southBoundLatitude.	real	-90 =< *northBoundLatitude* =< 90 *northBoundLatitude* > *southBoundLatitude*
operationAccuracy	Nominal accuracy of the coordinate operation described through the file contents, expressed in metres. Gives position error estimates for target coordinates of this coordinate operation, assuming no errors in source coordinates. This is a single value for the complete file content. Uncertainty estimates throughout the grid may be given as separate parameters.	real	>= 0
parameterCheckValues	Value of a parameter at a *checkPoint*. The parameter shall be identified through its *parameterName*. Note: not to be confused with *parameterValue*.	real	Table B.4 — GGXF conventions: grid node parameter identifier
parameterMaximumValue	The maximum value of this parameter in the GGXF file.	real	
parameterMinimumValue	The minimum value of this parameter in the GGXF file.	real	
parameterName	GGXF identifier of the *parameter*.	characterString	Table B.4 — GGXF conventions: grid node parameter identifier

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
parameters	Keyword in a GGXF file header to indicate the start of an ordered list of all parameters at nodes in the GGXF file. For GGXF files with multiple ggxfGroups, different subsets of the listed parameters may be present in different ggxfGroups. Each parameter shall have three attributes: *parameterName*, its *unitName* and the *unitSiRatio*. When a parameter is used in a coordinate operation it shall have one further attribute, *soureCrsAxis*. When a parameter is missing a value at any grid node in the GGXF file, the parameter shall have a *noDataFlag* attribute. Optionally this list of parameter attributes may be extended to include *parameterMinimumValue*, *parameterMaximumValue*, *uncertaintyMeasure*, *parameterSet* and *groupAdditionMethod*. When multiple parameters are listed, the sequence in which they are given shall match the sequence of parameter values at grid nodes throughout the GGXF file except where the sequence is overriden by a *gridParameters* attribute within a ggxfGroup.	orderedSet	
parameterSet	The name of the variable in which the parameter is stored in the GGXF netCDF file. If parameterSet is omitted the parameter is assumed to be stored in a 1D array with the same name as the *parameterName*.	characterString	
parameterValue	A constant value of this parameter in the ggxfGroup. Note: not to be confused with *parameterCheckValues*.	real	
producerDefinedContentCitation	Citation for content type not explicitly catered for through these Conventions. The citation should inform users what the file content is and how it should be used.	characterString	ISO 19115-1, Metadata fundamentals, CI_Citation
publicationDate	Date when the data in the file was issued.	dateTime	

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
scaleFactor	Scalar value by which deformation time functions are multiplied. Mainly applicable where the deformation time function includes multiple components which are scaled relative to each other. Note: not to be confused with the map projection parameter or netCDF attribute having the same name.	real	
sourceCoordinateEpoch	Coordinate epoch of *sourceCrsCoordinates*.	real	
sourceCrsAxis	The zero-based sequence number of the axis in the source CRS to which a *parameter* is to be applied in a coordinate operation. Examples: (i) in a GGXF file supporting a coordinate operation applying latitudeOffset and longitudeOffset to a source CRS in which the first axis is geodetic latitude and the second axis is geodetic longitude, the values for sourceCrsAxis for the latitudeOffset and longitudeOffset parameters are 0 and 1 respectively. (ii) in a GGXF file supporting a coordinate operation applying geoidHeight to a source CRS in which the first axis is geodetic latitude, the second axis is geodetic longitude and the third axis is ellipsoidal height, the value of the attribute sourceCrsAxis for the parameter geoidHeight is 2.	integer	> -1
sourceCrsCoordinates	The tuple of coordinates at a check point referenced to the source CRS defined in the GGXF file header. If no source CRS is declared the coordinates(s) are referenced to the interpolation CRS. The sequence of coordinates shall be that defined for the CRS in the file header. If the source CRS coordinates have time dependency they shall be accompanied by a *sourceCoordinateEpoch* attribute.	sequence	
sourceCrsWkt	Description of the coordinate reference system to which parameter values interpolated from the GGXF grid are applied in the coordinate operation described through the file, given in well-known text conformant to ISO 19162.	characterString	ISO 19162, Well known text representation of coordinate reference systems.

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
southBoundLatitude	Discovery metadata describing the southern latitude of a geographic bounding box (*contentApplicabilityExtent* and *contentBox*), expressed in decimal degrees (positive north) in the range -90 =< φ =< 90 and *southBoundLatitude* < *northBoundLatitude*.	real	-90 =< *southBoundLatitude* =< 90 *southBoundLatitude* < *northBoundLatitude*
startDate	*date* at the start of a time period. If omitted the temporal extent applies before and up to the temporal extent *endDate*. For a deformation model, the time function has a constant value before and up to this time.	dateTime	*startDate* < *endDate*
startEpoch	*epoch* at the start of a time period. For a deformation model, the time function has a constant value before and up to this time.	real	> 0
targetCoordinateEpoch	Coordinate epoch of *targetCrsCoordinates*.	real	
targetCrsCoordinates	The tuple of coordinates at a check point referenced to the target CRS defined in the GGXF file header. The sequence of coordinates shall be that defined for the CRS in the file header. If the target CRS coordinates have time dependency they shall be accompanied by a *targetCoordinateEpoch* attribute.	sequence	
targetCrsWkt	Description of the coordinate reference system that is the target in the coordinate operation described through the file given in well-known text conformant to ISO 19162.	characterString	ISO 19162, Well known text representation of coordinate reference systems.
tidalSurface	Surface derived from a *hydroidModel* to which depths are referenced.	characterString	Table B.5 — GGXF conventions: tidal surface identifier
tideSystem	Permanent tide system relevant to geodetic parameters in the GGXF file, described through *tideSystem*.	characterString	Table B.11 — GGXF conventions: permanent tide identifier
timeConstant	Time duration expressed in years used in the definition of some types of deformation time function.	real	

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
timeFunctions	Keyword in a ggxfGroup header to indicate the start of list of time functions applied to grid data within the ggxfGroup. Each time function shall have a *functionType* and one or more additional attributes. The time functions are summed to calculate the scale factor applying to a ggxfGroup at a given date/epoch.	sequence	Table B.7 — GGXF conventions: deformation time function identifier
title	Name by which the dataset is known	characterString	
uncertaintyMeasure	Statistical probability measure of an uncertainty parameter.	characterString	Table B.10 — GGXF conventions: uncertainty measure identifier
unitName	The identifier of the unit in which a *parameter* is given.	characterString	Table B.6 — GGXF conventions: unit identifier
unitSiRatio	The ratio of a unit to the SI base unit of that unit type, given as units per SI base unit. Examples: an *arc-second* = ((pi/180) / 3600) has a unit SI ratio of 4.84813681109536E-06 radian. A *foot* has a unit SI ratio of 0.3048 metre. *Parts per million* has a unit SI ratio of 1.0E-6 unity. If unitSiRatio is given for a unit included in Annex B.6 of these Conventions and the SI ratio values differ, the value in these conventions takes precedence.	real	Table B.6 — GGXF conventions: unit identifier
userDefinedMethodExample	Example of input and output from a user-defined operation method used to apply data in the GGXF file.	characterString	
userDefinedMethodFormula	Formula (text string) for a user-defined transformation or point motion operation method used to apply data in the GGXF file.	characterString	
userDefinedMethodFormulaCitation	Citation for formula(s) for a user-defined transformation or point motion operation method used to apply data in the GGXF file.	characterString	ISO 19115-1, Metadata fundamentals, CI_Citation
version	Version identifier for the data.	characterString	

Attribute identifier (keyword)	**Definition**	**Data type**	**Domain** (valid values)
westBoundLongitude	Discovery metadata describing the western longitude of a geographic bounding box (*contentApplicabilityExtent* and *contentBox*), expressed in decimal degrees (positive east) in the range -180 =< λ =< 180. If the geographic bounding box crosses the 180° meridian then *westBoundLongitude* > *eastBoundLongitude*, else *westBoundLongitude* < *eastBoundLongitude*.	real	-180 =< *westBoundLongitude* =< 180 *westBoundLongitude* ≠ *eastBoundLongitude*

B.3 GGXF Conventions: keywords for supplementary attributes

For the header attribute *content* (refer to Table B.2), the GGXF conventional identifier in Table B.3 should be used. For each content type, those parameters which are mandatory are given in this table and defined in Table B.4. These mandatory parameters are listed in alphabetical order: in the GGXF file the order is not constrained but whatever it is must be declared in the file or ggxfGroup header. For each of these parameters, a corresponding uncertainty parameter is optional. For example, if the content type requires a *latitudeOffset* parameter, then it also supports an optional *latitudeOffsetUncertainty* parameter.

Table B.3 — GGXF conventions: content identifier

File content identifier	Definition	Mandatory parameters
Cartesian2dOffsets	Differences in coordinates between two 2-dimensional CRSs having Cartesian coordinates. Interpolated offsets are parameters to be added to the coordinates referenced to the source coordinate reference system: $X_T = X_S + \delta X$ $Y_T = Y_S + \delta Y$	Two offsets, these being consistent with the source CRS positive axis directions, for example: *eastingOffset* *northingOffset* or *southingOffset* *westingOffset*
Cartesian3dOffsets	Differences in coordinates between two 3-dimensional CRSs having Cartesian coordinates. Interpolated offsets are parameters to be added to the coordinates referenced to the source coordinate reference system: $X_T = X_S + \delta X$ $Y_T = Y_S + \delta Y$ $Z_T = Z_S + \delta Z$ Note: use *geocentricTranslations* when the coordinate reference systems are geodetic with geocentric Cartesian coordinate system.	Three offsets, these being consistent with the source CRS positive axis directions, e.g.: *eastingOffset* *heightOffset* *northingOffset* or *heightOffset* *southingOffset* *westingOffset,*
deformationModel	Describes secular and episodic geodynamic motion. The functional model required for use with this version of the GGXF format is given in [1].	Each group of a deformation model must define one or both of the following sets of parameters: *displacementEast* *displacementNorth* or *displacementUp* and/or their uncertainties. When uncertainties are included, *displacementHorizontalUncertainty* shall not be included if *displacementEastUncertainty* or *displacementNorthUncertainty* is included. Each group must include a *timeFunctions* attribute defining one or more time functions. Refer to Table B.7

File content identifier	Definition	Mandatory parameters
deviationsOfTheVertical	Deviation of the gravity vector from the ellipsoidal normal, resolved into north and east components. Positive when the downward gravity vector is to the south and west of the ellipsoid normal. Notes: (i) deviation of the vertical is sometimes called deflection of the vertical. (ii) The surface at which the deviation applies can be included in a *comment* or in the *abstract*.	*deviationEast* *deviationNorth*
geocentricTranslations	Differences in coordinates between two 3-dimensional geodetic CRSs having geocentric Cartesian coordinates. Interpolated offsets are parameters to be added to the coordinates referenced to the source coordinate reference system: $X_T = X_S + \delta X$ $Y_T = Y_S + \delta Y$ $Z_T = Z_S + \delta Z$ Note: A specialised form of *Cartesian3dOffsets*.	*geocentricXOffset* (δX) *geocentricYOffset* (δY) *geocentricZOffset* (δZ)
geographic2dOffsets	Differences in geodetic latitude and geodetic longitude between two geographic CRSs. Interpolated offsets are parameters to be added to the latitude and longitude referenced to the source geographic coordinate reference system: $\varphi_T = \varphi_S + \delta\varphi$ $\lambda_T = \lambda_S + \delta\lambda$	*latitudeOffset* ($\delta\varphi$) *longitudeOffset* ($\delta\lambda$)
geographic3dOffsets	Differences in geodetic latitude, geodetic longitude and ellipsoid height between two geographic CRSs. Interpolated offsets are parameters to be added to the latitude, longitude and ellipsoidal height referenced to the source geographic coordinate reference system: $\varphi_T = \varphi_S + \delta\varphi$ $\lambda_T = \lambda_S + \delta\lambda$ $h_T = h_S + \delta h$	*latitudeOffset* ($\delta\varphi$) *longitudeOffset* ($\delta\lambda$) *ellipsoidalHeightOffset* (δh)

File content identifier	Definition	Mandatory parameters
geoidModel	Height of a height reference surface above the ellipsoid surface associated with a specified geographic 3D CRS, positive upwards from the reference ellipsoid. In this specification this content type is used for geoid models, quasi-geoid models, and height correction (hybrid) models without distinction. The interpolated offset ('geoid' height, ζ) is a subtractive correction to the ellipsoidal height in the specified (source) geographic coordinate reference system: $H_T = h_S - \zeta$ Notes: (i) Differences in gravity-related height between two vertical CRSs should be described as *verticalOffsets*. (ii) The type of geoid model can be included in the abstract or in a comment immediately following content type (see example E.2).	*geoidHeight* (ζ)
hydroidModel	Height difference between the ellipsoid surface associated with a specified geographic 3D CRS and an identified tidal surface of a vertical coordinate reference system, positive upwards from the reference ellipsoid. The interpolated offset (height correction) is a subtractive correction to the ellipsoidal height in the specified (source) geographic coordinate reference system: $H_T = h_S - C$ Because it is normally applied to a target CRS in the depth domain the height correction model relationship to ellipsoidal height has its sign reversed: $D_T = C - h_S$ In a marine context depth of the seabed below the tidal surface is derived from observed water depth and observed three-dimensional position including ellipsoidal height, both observations to a common vessel reference point, and the ellipsoid height of the tidal surface interpreted from the hydroid model. $D = (D_{obs} – h_{obs}) + C$	*hydroidHeight* (C) A GGXF file containing hydroid model content must include an attribute *tidalSurface*. See Table B.5 for valid tidal surface identifiers.

File content identifier	Definition	Mandatory parameters
velocityGrid	Describes secular motion within a specified CRS from a specified coordinate epoch through velocity values. The components into which the velocity is resolved (for example: east, north, up) are described through the tuple of grid node parameters in the ggxfGroup *gridParameters* attribute or if *gridParameters* is not defined by the ordered list in the file *parameters* attribute. See Table B.4.	Two or three parameters. Either *velocityEast* *velocityNorth* or *velocityEast* *velocityNorth* *velocityUp* or *velocityX* *velocityY* *velocityZ*
verticalOffsets	Differences in gravity-related height or depth between two vertical CRSs. For source and target CRSs with height coordinate systems the interpolated offset (height difference) is a parameter to be added to the gravity-related height referenced to the source vertical coordinate reference system: $H_T = H_S + \delta H$ For source and target CRSs with depth coordinate systems the interpolated offset (depth difference) is a parameter to be added to the depth referenced to the source vertical coordinate reference system: $D_T = D_S + \delta D$	One offset, this being consistent with the source CRS positive axis direction, i.e.: *heightOffset* (δH) or *depthOffset* (δD)
producerDefinedContent	Content is not given above. Producers should provide details of the content including parameters and how they should be treated through a citation. Implementations are not obliged to handle producer-defined content.	*producerDefinedContentCitation*

The set of parameters at grid nodes is described in ggxfGroup headers as an ordered list in the *gridParameters* attribute, or if that is not defined by the ordered list in the file header *parameters* attribute. All grids within the ggxfGroup must have the same set of parameters. GGXF conventional keywords for the identifiers of these parameters are given in Table B.4.

Table B.4 — GGXF conventions: grid node parameter identifier

Parameter identifier	Definition
depthOffset	Parameter to be added to a depth that is referenced to the source vertical CRS to obtain the depth referenced to the target CRS. Note: default unit is that of the source CRS.
depthOffsetUncertainty	Uncertainty estimate of the *depthOffset*. The measure is described through *uncertaintyMeasure*.
deviationEast	East-west component (η) of the *deviation of the vertical* at the Earth's surface, positive eastwards (positive when the plumbline intersects the celestial sphere east of the ellipsoidal normal, i.e. when the downward gravity vector is deviated to the west of the ellipsoid normal). η = (astronomic longitude Λ minus geodetic longitude λ) * cos(geodetic latitude φ).
deviationEastUncertainty	Uncertainty estimate of the *deviationEast*. The measure is described through *uncertaintyMeasure*.
deviationNorth	North-south component (ξ) of the *deviation of the vertical* at the Earth's surface, positive northwards (positive when the plumbline intersects the celestial sphere north of the ellipsoidal normal, i.e. when the downward gravity vector is deviated to the south of the ellipsoid normal). ξ = (astronomic latitude Φ minus geodetic latitude φ)
deviationNorthUncertainty	Uncertainty estimate of the *deviationNorth*. The measure is described through *uncertaintyMeasure*.
displacementEast	Easterly component of a displacement, to be added to the source CRS longitude or easting to obtain the longitude or easting in the target CRS. Note: default unit is that of the source CRS.
displacementEastUncertainty	Uncertainty estimate of the *displacementEast* component. The measure is described through *uncertaintyMeasure*.
displacementHorizontalUncertainty	Single uncertainty estimate covering both *displacementEast* and *displacementNorth* components. The measure is described through *uncertaintyMeasure*.
displacementNorth	Northerly component of a displacement, to be added to the source CRS latitude or northing to obtain the latitude or northing in the target CRS. Note: default unit is that of the source CRS.
displacementNorthUncertainty	Uncertainty estimate of the *displacementNorth* component. The measure is described through *uncertaintyMeasure*.
displacementUp	Upwards component of a displacement, to be added to the source CRS height to obtain the height in the target CRS. Sometimes called vertical displacement. Note: default unit is that of the source CRS.
displacementUpUncertainty	Uncertainty estimate of the *displacementUp*. The measure is described through *uncertaintyMeasure*.
eastingOffset	Parameter to be added to an easting that is referenced to the source CRS to obtain the easting referenced to the target CRS. Note: default unit is that of the source CRS easting.
eastingOffsetUncertainty	Uncertainty estimate of the *eastingOffset*. The measure is described through *uncertaintyMeasure*.

Parameter identifier	Definition
ellipsoidalHeightOffset	Parameter to be added to an ellipsoidal height in the Source CRS to obtain the ellipsoidal height in the Target CRS. Note: default unit is that of the source CRS ellipsoidal height.
ellipsoidalHeightOffsetUncertainty	Uncertainty estimate of the *ellipsoidalHeightOffset*. The measure is described through *uncertaintyMeasure*.
geocentricXOffset	Parameter to be added to a geocentric Cartesian X coordinate that is referenced to the source CRS to obtain a geocentric_X coordinate referenced to the target CRS.
geocentricXOffsetUncertainty	Uncertainty estimate of the *geocentricXoffset*. The measure is described through *uncertaintyMeasure*.
geocentricYOffset	Parameter to be added to a geocentric Cartesian Y coordinate that is referenced to the source CRS to obtain a geocentric_Y coordinate referenced to the target CRS.
geocentricYOffsetUncertainty	Uncertainty estimate of the *geocentricYoffset*. The measure is described through *uncertaintyMeasure*.
geocentricZOffset	Parameter to be added to a geocentric Cartesian Z coordinate that is referenced to the source CRS to obtain a geocentric_Z coordinate referenced to the target CRS.
geocentricZOffsetUncertainty	Uncertainty estimate of the *geocentricZoffset*. The measure is described through *uncertaintyMeasure*.
geoidHeight	Parameter to be subtracted from an ellipsoidal height that is referenced to the source CRS to obtain a gravity-related height referenced to the target vertical CRS. Note: this attribute does not distinguish between gravimetric geoid, quasi-geoid or hybrid geoid models.
geoidHeightUncertainty	Uncertainty estimate of the *geoidHeight*. The measure is described through *uncertaintyMeasure*.
heightOffset	Parameter to be added to a gravity-related height that is referenced to the source vertical CRS to obtain a gravity-related height referenced to the target CRS.
heightOffsetUncertainty	Uncertainty estimate of the *heightOffset*. The measure is described through *uncertaintyMeasure*.
hydroidHeight	Parameter to be subtracted from an ellipsoidal height that is referenced to the source CRS to obtain a gravity-related height referenced to the target vertical CRS and tidal datum. The tidal surface is identified through the *tidalSurface* attribute (Table B.5).
hydroidHeightUncertainty	Uncertainty estimate of the *hydroidHeight*. The measure is described through *uncertaintyMeasure*.
latitudeOffset	Parameter to be added to the source CRS geodetic latitude to obtain the target CRS geodetic latitude. Note: when geodetic latitude is in degrees, *latitudeOffset* is permitted to be in *arc-minutes* or *arc-seconds*
latitudeOffsetUncertainty	Uncertainty estimate of the *latitudeOffset*. The measure is described through *uncertaintyMeasure*.
longitudeOffset	Parameter to be added to the source CRS geodetic longitude to obtain the target CRS geodetic longitude.
longitudeOffsetUncertainty	Uncertainty estimate of the *longitudeOffset*. The measure is described through *uncertaintyMeasure*.

Parameter identifier	Definition
nodeDepth	Depth at a node of a grid where the Interpolation CRS is a compound CRS including a vertical component carrying depths. Note: see 5.7.7 recommendations 2 and 3.
nodeEasting	Easting at a node of a grid where the Interpolation CRS is projected. Note: see 5.7.7 recommendations 2 and 3.
nodeEllipsoidalHeight	Ellipsoidal height (geodetic height) at a node of a grid where the Interpolation CRS is geographic 3D. Note: see 5.7.7 recommendations 2 and 3.
nodeGeocentricX	X coordinate at a node of a grid where the Interpolation CRS is geodetic with geocentric Cartesian coordinate system. Note: see 5.7.7 recommendations 2 and 3
nodeGeocentricY	Y coordinate at a node of a grid where the Interpolation CRS is geodetic with geocentric Cartesian coordinate system. Note: see 5.7.7 recommendations 2 and 3.
nodeGeocentricZ	Z coordinate at a node of a grid where the Interpolation CRS is geodetic with geocentric Cartesian coordinate system. Note: see 5.7.7 recommendations 2 and 3.
nodeHeight	Gravity-related height at a node of a grid where the Interpolation CRS is a compound CRS including a vertical component carrying gravity-related heights. Note: see 5.7.7 recommendations 2 and 3.
nodeLatitude	Geodetic latitude at a node of a grid where the Interpolation CRS is geographic. Note: see 5.7.7 recommendations 2 and 3.
nodeLongitude	Geodetic longitude at a node of a grid where the Interpolation CRS is geographic. Note: see 5.7.7 recommendations 2 and 3.
nodeNorthing	Northing at a node of a grid where the Interpolation CRS is projected. Note: see 5.7.7 recommendations 2 and 3.
nodeSouthing	Southing at a node of a grid where the Interpolation CRS is projected. Note: see 5.7.7 recommendations 2 and 3.
nodeWesting	Westing at a node of a grid where the Interpolation CRS is projected. Note: see 5.7.7 recommendations 2 and 3
northingOffset	Parameter to be added to the source CRS northing to obtain the target CRS northing.
northingOffsetUncertainty	Uncertainty estimate of the *northingOffset*. The measure is described through *uncertaintyMeasure*.
southingOffset	Parameter to be added to the source CRS southing to obtain the target CRS southing. Applies to transformations between projected CRSs with an axis positive southwards.
southingOffsetUncertainty	Uncertainty estimate of the *southingOffset*. The measure is described through *uncertaintyMeasure*.
velocityEast	Easterly component of the velocity vector at a point.
velocityEastUncertainty	Uncertainty estimate of the *velocityEast*.
velocityNorth	Northerly component of the velocity vector at a point.
velocityNorthUncertainty	Uncertainty estimate of the *velocityNorth*.

Parameter identifier	Definition
velocityUp	Up component of the velocity vector at a point.
velocityUpUncertainty	Uncertainty estimate of the *velocityUp*.
velocityX	Geocentric X component of the velocity vector at a point.
velocityXUncertainty	Uncertainty estimate of the geocentric X velocity (*velocityX*).
velocityY	Geocentric Y component of the velocity vector at a point.
velocityYUncertainty	Uncertainty estimate of the geocentric Y velocity (*velocityY*).
velocityZ	Geocentric Z component of the velocity vector at a point.
velocityZUncertainty	Uncertainty estimate of the geocentric Z velocity (*velocityZ*).
westingOffset	Parameter to be added to the source CRS westing to obtain the target CRS westing. Applies to transformations using *Cartesian2Doffsets* between projected CRSs with an axis positive westwards.
westingOffsetUncertainty	Uncertainty estimate of the *westingOffset*.

When the file content (refer to Table B.3) is *hydroidModel*, the *tidalSurface* to which ellipsoidal heights are reduced through applying the hydroid model should be described through the GGXF tidal surface identifier given in Table B.5. The definitions are from the International Hydrographic Organization (IHO).

Table B.5 — GGXF conventions: content tidal surface identifier

Identifier	Tidal Surface Definition
CD	Chart Datum. The plane of reference to which charted depths and drying heights are related. In tidal areas CD is chosen to show the least depth of water found in any place under 'normal' meteorological conditions.
HAT	Highest Astronomical Tide. The highest tide level which can be predicted to occur under average meteorological conditions and under any combination of astronomical conditions.
HHWLT	Higher High Water Large Tide. The average of the highest high waters, one from each of 19 years of observations.
HW	High Water. The highest level reached at a place by the water surface in one tidal cycle. When used on inland (non-tidal) waters it is generally defined as a level which the daily mean water level exceeds less than 5% of the time.
ISLW	Indian Spring Low Water. The level below MSL equal to the sum of the amplitudes of the harmonic constituents M2, S2, K1 and O1. It approximates mean lower low water spring tides (MLLWS).
LAT	Lowest Astronomical Tide. The lowest tide level which can be predicted to occur under average meteorological conditions and under any combination of astronomical conditions.
LLWLT	Lower Low Water Large Tide. The average of the lowest low waters, one from each of 19 years of observations.
LW	Low Water. The lowest level reached by the water surface in one tidal cycle. When used in inland (non-tidal) waters it is generally defined as a level which the daily mean water level would fall below less than 5% of the time.
MHHW	Mean Higher High Water. The average height of the higher high waters at a place over a 19-year period.
MHW	Mean High Water. The average height of the high waters at a place over a 19-year period.
MHWST	Mean High Water Spring Tides. The average height of the high waters of spring tides.
MLLW	Mean Lower Low Water. The average height of the lower low waters at a place over a 19-year period.
MLLWST	Mean Lower Low Water Spring Tides. The average height of the lower low water spring tides at a place.
MLW	Mean Low Water. The average height of all low waters at a place over a 19-year period.
MLWST	Mean Low Water Spring Tides. The average height of the low waters of spring tides.
MSL	Mean Sea Level. The average height of the surface of the sea at a tide station for all stages of the tide over a 19-year period, usually determined from hourly height readings measured from a fixed predetermined reference level.

The units in which parameters are given in the GGXF grid is described through an ordered list of unit identifier (*unitName*) and the ratio of the unit to the SI standard unit for that unit type (*unitSiRatio*), refer to Table B.2. The inclusion of ratio to the standard SI unit permits the use of units not in the GGXF Conventions. GGXF conventional identifiers for units are given in Table B.6. The GGXF conventional identifier should be used when it is applicable to file content. This list is not exclusive and may be extended using ISO 80000-3, Quantities and units - Part 3: Space and time, or other parts of ISO 80000 as required.

Table B.6 — GGXF conventions: unit identifier and ratio to SI standard unit

Unit identifier	Definition	SI standard unit	*unitSiRatio* (Ratio to SI standard unit)
arc-minute	Angle unit of one sixtieth (1/60) of a *degree*.	radian	0.000290888208665722
arc-second	Angle unit of one sixtieth (1/60) of an *arc-minute*.	radian	4.84813681109536E-06
centimetre	Length unit of one hundredth (1/100) of a *metre*.	metre	0.01
cm/yr	Length rate of *centimetre* per *year*.	m/s	3.16887651727315E-10
degree	Angle unit of one three hundred and sixtieth (1/360) part of a circle.	radian	0.0174532925199433
foot	Length unit by definition 0.3048 *metre* exactly. Sometimes referred to as the International foot. Note: other types of foot with different ratio to the metre exist. In GGXF these must not use the four character string 'foot' as their unit identifier.	metre	0.3048
m/s	SI standard unit for length rate, *metre* per *second*.	m/s	1.0
m/yr	Length rate of *metre* per *year*.	m/s	3.16887651727315E-08
mas/yr	Angle rate of *milliarc-second* per *year*.	rad/s	1.53631468932076E-16
metre	SI standard unit for length.	metre	1.0
milliarc-second	Angle unit of one thousandth (1/1000) of an *arc-second*.	radian	4.84813681109536E-09
millimetre	Length unit of one thousandth (1/1000) of a *metre*.	metre	0.001
mm/yr	Length rate of *millimetre* per *year*.	m/s	3.16887651727315E-11
ppb	parts per billion. Scale unit of where billion is one thousandth (1/1000) of a million.	unity	0.000000001
ppb/yr	parts per billion per year.	unity/s	3.16887651727315E-17
ppm	Scale unit of parts per million.	unity	0.000001
ppm/yr	Scale rate of *parts per million* per *year*.	unity/s	3.16887651727315E-14
radian	SI standard unit for angle.	radian	1.0
rad/s	SI standard unit for angle rate, *radian* per *second*.	rad/s	1.0
second	SI standard unit for time.	second	1.0
unity	Standard unit for scale.	unity	1.0
unity/s	Standard unit for scale rate, *unity* per *second*.	unity/s	1.0

Unit identifier	Definition	SI standard unit	*unitSiRatio* (Ratio to SI standard unit)
year	Time unit. In the GGXF conventions it is a fixed duration in seconds as defined by the International Union of Geological Sciences (IUGS) and International Union of Pure and Applied Chemistry (IUPAC). This approximation is adequate for giving unit SI ratios for angle, length and scale rates. Source: Pure and Applied Chemistry, Vol. 83, No. 5, pp. 1159–1162, 2011.	second	31556925.445

Deformation model time functions are defined in [1]. In a deformation model GGXF file, each ggxfGroup header defines one or more time functions. The sum these is applied as a scale factor to all grids within the ggxfGroup. Each time function has a *functionType* attribute defining the function type and other attributes depending that function type. GGXF conventional keywords for the time function types are given in Table B.7 and the attributes applicable to each type are summarised in Table B.8

Table B.7 — GGXF conventions: deformation time function identifier

Time function identifier	Definition
acceleration	Deformation model time function describing deformation growing proportionally to the square of elapsed time. An acceleration time function shall have one attribute: either *functionReferenceDate or functionReferenceEpoch*. Additionally it may have up to three further attributes: *scaleFactor*, *startDate* or *startEpoch* and *endDate* or *endEpoch.*
cyclic	Deformation model time function describing deformation proportional to the sine of elapsed time. A cyclic time function shall have two attributes: either *functionReferenceDate or functionReferenceEpoch, and frequency*. Additionally it may have up to three further attributes: *scaleFactor, startDate* or *startEpoch* and *endDate* or *endEpoch.*
exponential	Deformation model time function describing deformation proportional to the exponential of time elapsed after an event. An exponential time function shall have two attributes: either *startDate* or *startEpoch,* and *timeConstant.* Additionally it may have up to four further attributes: *scaleFactor, startDate* or *startEpoch*, *endDate* or *endEpoch*, and *functionReferenceDate* or *functionReferenceEpoch.*
hyperbolicTangent	Deformation model time function describing deformation proportional to the hyperbolic tangent of time elapsed before or after an event. A hyperbolic tangent time function shall have two attributes: either *eventDate* or *eventEpoch,* and *timeConstant.* Additionally it may have up to four further attributes: *scaleFactor, startDate* or *startEpoch*, *endDate* or *endEpoch*, and *functionReferenceDate* or *functionReferenceEpoch.*
logarithmic	Deformation model time function describing deformation proportional to the logarithm of time elapsed after an event. A logarithmic time function shall have two attributes: either *eventDate* or *eventEpoch,* and *timeConstant.* Additionally it may have up to four further attributes: *scaleFactor, startDate* or *startEpoch*, *endDate* or *endEpoch*, and *functionReferenceDate* or *functionReferenceEpoch.*
ramp	Deformation model time function describing deformation growing proportional to elapsed time from a start time to an end time. A ramp time function shall have two attributes: either *startDate* or *startEpoch*, and *endDate* or *endEpoch.* Additionally it may have up to two further attributes: *scaleFactor,* and *functionReferenceDate* or *functionReferenceEpoch.*
step	Deformation model time function describing an instantaneous change to displacement at an event time. A step time function shall have one attribute, *eventDate* or *eventEpoch*. Additionally it may have up to four further attributes: *scaleFactor, startDate* or *startEpoch*, *endDate* or *endEpoch*, and *functionReferenceDate* or *functionReferenceEpoch.*
velocity	Deformation model time function describing deformation growing proportionally to elapsed time. A velocity time function shall have one attribute, either *functionReferenceDate or functionReferenceEpoch*. Additionally it may have up to three further attributes: *scaleFactor, startDate* or *startEpoch* , and *endDate* or *endEpoch*.

Table B.8 — GGXF conventions: attributes required by time functions

Time function identifier (functionType)	Mandatory attributes							*Optional attributes*			
	function type	start date or epoch	end date or epoch	event date or epoch	time constant	function reference date or epoch	frequency	*start date or epoch*	*end date or epoch*	*function reference date or epoch*	*scale factor*
acceleration	√					√		*?*	*?*		*?*
cyclic	√					√	√	*?*	*?*		*?*
exponential	√			√	√			*?*	*?*	*?*	*?*
hyperbolic tangent	√			√	√			*?*	*?*	*?*	*?*
logarithmic	√			√	√			*?*	*?*	*?*	*?*
ramp	√	√	√							*?*	*?*
step	√			√				*?*	*?*	*?*	*?*
velocity	√					√		*?*	*?*		*?*

For issues associated with calendar arithmetic using dates refer to annex D of ISO 19111 [2].

Recommendation 22 To avoid problems with calendar arithmetic, use epoch rather than date.

For the header attribute *interpolationMethod* (refer to Table B.2), the interpolation methods given in Table B.9 are frequently encountered in geodetic use and these attribute values should be used where appropriate. The recommended formula are given through the formula citation. Note that where multiple citations are given, application of their formula may lead to different interpolation results, particularly near the edge of grids. Bilinear and trilinear interpolation methods are unambiguous. This list of interpolation methods is not exclusive and may be extended through the use of the interpolation method identifier (*userDefinedMethodName*) and *userDefinedMethodFormulaCitation*.

Table B.9 — GGXF conventions: interpolation method identifier

Interpolation method identifier	Description	Example Formula Citation
bicubic	Cubic polynomial interpolant used for interpolating two variables on a two-dimensional rectilinear grid.	*Press et al.*, 2007 [15], pp.136-139 *Russell*, 1995 [16]; *Shi et al.*, 2005 [17].
bilinear	Interpolation of two variables on a rectilinear 2D grid performed using linear interpolation first in one direction, and then again in the other direction.	*Press et al.*, 2007 [15], pp.132-134 *Young and Gregory*, 1988 [20], pp.342-343 *Russell*, 1995 [16]
biquadratic	Quadratic polynomial interpolant used for interpolating two variables on a two-dimensional rectilinear grid.	*Smith*, 2022 [18] *Russell*, 1995 [16] *Shi et al.*, 2005 [17].
tricubic	Cubic polynomial interpolant used for interpolating three variables on a three-dimensional rectilinear grid.	*Lekien and Marsden*, 2005 [14] *Arata*, 1995 [12]
trilinear	Linear interpolation of three variables on a three-dimensional rectilinear grid performed using linear interpolation sequentially in each of the three directions.	*Wagner*, 2008 [19] *Haynes et al.*, 2007 [13]
citation	The interpolation method is not included in the GGXF Conventions but is described through a *userDefinedMethodFormulaCitation* conformant to ISO 19115-1, Metadata fundamentals. Be aware that GGXF file reading software may not handle this!	

For the header attribute *uncertaintyMeasure* (refer to Tables B.2 and B.5), the GGXF identifier in Table B.10 should be used. Measures have been taken from references [6], [7].

Table B.10 — GGXF conventions: uncertainty measure identifier

Uncertainty measure identifier	Definition
One-dimensional Accuracy Indicators	
1SE	**S**tandard **E**rror of the mean. 68% probability.
2SE	Twice the **S**tandard **E**rror. 95% probability.
3SE	Three times the **S**tandard **E**rror. 99% probability.
RMSE	**R**oot **M**ean **S**quare **E**rror. Standard deviation of residuals. 68% probability. Note: Standard deviation (SD) is not a measure of accuracy, but of precision. Standard error of the mean yields a confidence interval, which more correctly expresses accuracy.
Two-dimensional Accuracy Indicators	
1CEE	Standard **C**onfidence **E**llips**E**. An ellipse, centred on the mean, whose boundary is expected to include the true value with 39% probability.
2CEE	Standard **C**onfidence **E**llips**E**. An ellipse, centred on the mean, whose boundary is expected to include the true value with 95% probability.
1CEP	**C**ircular **E**rror **P**robable. The radius of a circle, centred on the mean, whose boundary is expected to include the true value with 50% probability.
2CEP	**C**ircular **E**rror **P**robable. The radius of a circle, centred on the mean, whose boundary is expected to include the true value with 95% probability.
1DRMS	**D**istance **R**oot **M**ean **S**quared. Square root of the trace of a two-dimensional covariance matrix. 66% probability.
2DRMS	Twice the **D**istance **R**oot **M**ean **S**quared. Square root of the trace of a two-dimensional covariance matrix. 97% probability.
Three-dimensional Accuracy Indicators	
1CED	Standard **C**onfidence **E**llipsoi**D** . An ellipsoid, centred on the mean, whose boundary is expected to include the true value with 20% probability.
2CED	Standard **C**onfidence **E**llipsoi**D** . An ellipsoid, centred on the mean, whose boundary is expected to include the true value with 95% probability.
1SEP	**S**pherical **E**rror **P**robable. The radius of a sphere, centred on the mean, whose boundary is expected to include the true value with 50% probability.

For the header attribute *tideSystem* (refer to Table B.2), the permanent tide identifiers given in Table B.11 shall be used. These attributes are from the IERS Conventions 2010 chapter 1.1 [ref].

Table B.11 — GGXF conventions: permanent tide identifier

Earth tide identifier	Definition
conventionalTideFree	*instantaneous* geopotential or instantaneous crust position after removal of total tidal effects through the application of conventional Love numbers.
instantaneous	Observed geopotential or observed crust location.
meanCrust	*conventionalTideFree* crust with permanent deformation due to tidal potential restored using conventional Love numbers.
meanTide	*zeroTide* geopotential with permanent part of the tide-generating potential restored.
tideFree	*zeroTide* geopotential or *meanCrust* with permanent deformation produced by the tidal potential removed using the secular or fluid-limit value for the relevant Love number.
zeroTide	*conventionalTideFree* geopotential with permanent deformation due to tidal potential restored using conventional Love numbers.

B.4 GGXF Conventions: keywords for citation and responsible party attributes

The GGXF file header should include metadata giving details of the party responsible for producing the file. The CI_Citation and CI_Responsibility classes from ISO 19115-1 - *Metadata fundamentals* should be used for this. The attributes from these ISO 19115-1 classes are part of the GGXF Conventions with their UML role names being used as GGXF keywords. The 19115-1 UML diagram is shown in Figure B.1 and selected example attributes in Table B.12. Their use is exemplified in Annex E.

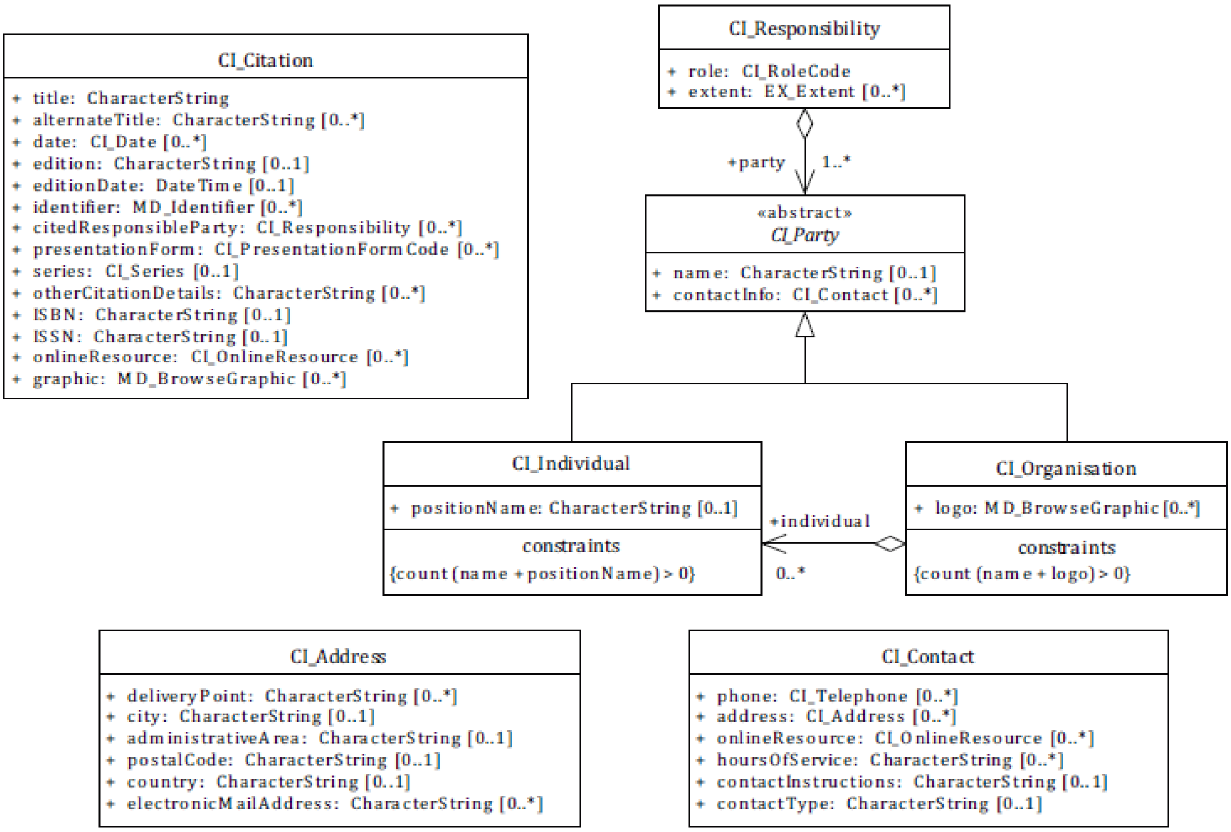


Figure B.1 - Citation and responsible party information classes from ISO 19115-1

Table B.12 — GGXF conventions: example citation attributes from ISO 19115-1

GGXF keyword	**Definition** (from ISO 19115-1)	**Data Type**	**Domain**
city	city of the location	characterString	
country	country of the address	characterString	
deliveryPoint	address line for the location EXAMPLE Street number and name, suite number, etc.	characterString	
onlineResourceLinkage	location (address) for on-line access using a Uniform Resource Locator/ Uniform Resource Identifier address or similar addressing scheme such as http://www.statkart.no/isotc211	characterString	Text restricted to URL (see IETF RFC 3986)
partyName	name of the party (individual or organization) In GGXF this is interpreted to be the authority producing the GGXF file.	characterString	
postalCode	ZIP or other postal code	characterString	
publicationDate	date identifies when the resource was issued	dateTime	
title	name by which the cited information is known	characterString	

B.5 GGXF Conventions: mapping of GGXF identifiers to netCDF attributes

When GGXF uses a concept that previously has been adopted in netCDF and associated conventions, where possible the GGXF Conventions have adopted the netCDF identifier. But some netCDF attribute identifiers are insufficiently specific for geodetic purposes and in these cases the GGXF Conventions have defined a different identifier. When implemented in netCDF these GGXF identifiers are changed to the nearest equivalent netCDF attribute name. Annex B.5 describes this mapping.

Tables B.14 to B.16 give the mapping to netCDF attributes of GGXF file header, ggxfGroup header and grid header identifiers respectively. In these tables the attributes are given a Category indicating from where the name originates. The Category acronyms used in Tables B.14 to B.16 are given in Table B.13.

Note that GGXF uses upperCamelCase for its identifier whilst netCDF favours snake_case. All GGXF Convention identifiers that do not appear in the following Tables B.14 to B.16 are used in netCDF with the GGXF upperCamelCase identifier unchanged.

Table B.13. Categories in netCDF attribute mapping

Category	Description
acdd	Represented as a netCDF attribute. Attribute name comes from the Attribute Convention for Data Discovery (ACDD).
ggxf	Represented as a netCDF attribute. Attribute name comes from the GGXF Conventions.
netcdf	Represented as a netCDF attribute. Attribute name comes from the netCDF conventions
dimension	Represented as a netCDF dimension
directory	Represented in the netCDF group name used to identify the group in the netCDF file. (This is like a directory name file name in a file system)
variable	Represented by one or more netCDF variables. The mapping of grid data is described in more detail in 6.3.
yaml	The attribute is specific to the GGXF YAML format and is not represented in the GGXF netCDF encoding.

GGXF Conventions attribute names for file header attributes (refer to B.2 and B.4) that are changed for use in a GGXF netCDF file are given in Table B.14.

Table B.14 — GGXF conventions: Mapping of GGXF file header attributes to netCDF attributes

GGXF attribute	Category	netCDF attribute name
abstract	acdd	summary
electronicMailAddress	acdd	creator_email
filename	acdd	source_file
ggxfGroups	yaml	
ggxfVersion	netcdf	Conventions
onlineResourceLinkage	acdd	publisher_url
partyName	acdd	institution
publicationDate	acdd	date_issued
version	acdd	product_version
Structured attributes[2]		
contentApplicabilityExtent.boundingBox.eastBoundLongitude	acdd	geospatial_lon_max
contentApplicabilityExtent.boundingBox.northBoundLatitude	acdd	geospatial_lat_max
contentApplicabilityExtent.boundingBox.southBoundLatitude	acdd	geospatial_lat_min
contentApplicabilityExtent.boundingBox.westBoundLongitude	acdd	geospatial_lon_min
contentApplicabilityExtent.boundingPolygon	acdd	geospatial_bounds
contentApplicabilityExtent.extentDescription	ggxf	extentDescription
contentApplicabilityExtent.extentTemporal.endDate	acdd	time_coverage_end
contentApplicabilityExtent.extentTemporal.startDate	acdd	time_coverage_start
contentApplicabilityExtent.extentVertical.extentVerticalCrsWkt	ggxf	extentVerticalCrsWkt
contentApplicabilityExtent.extentVertical.extentVerticalMaximum	acdd	geospatial_vertical_max
contentApplicabilityExtent.extentVertical.extentVerticalMinimum	acdd	geospatial_vertical_min
identifier.codeSpace=DOI	ggxf	DOI

GGXF Conventions attribute names for ggxfGroup header attributes (refer to B.2) that are changed for use in a GGXF netCDF file are given in Table B.15.

Table B.15 — GGXF conventions: Mapping of GGXF group header attributes to netCDF variables

GGXF attribute	Category	netCDF representation
ggxfGroupName	directory	Defined by the name of the netCDF group representing the ggxfGroup.
grids	yaml	Child netCDF groups of the group representing the ggxfGroup.

[2] Other attributes which are structured lists of values are implemented as described in 6.3.4.2.

GGXF Conventions attribute names for grid header attributes (refer to B.2) that are changed for use in a GGXF netCDF file are given in Table B.16.

Table B.16 — GGXF conventions: Mapping of GGXF grid header attributes to netCDF variables

GGXF attribute	Category	netCDF representation
data	variable	One or more netCDF variables named by the *parameterSet*, or if *parameterSet* is not defined by netCDF variables named by the *parameterName* attribute of the parameter they contain.
dataSource	yaml	Not applicable.
gridName	directory	Defined by the name of the netCDF group representing the grid.
childGrids	yaml	Child netCDF groups of the group representing the grid.

B.6 GGXF Conventions: external grid file format

The GGXF binary file format requires grid data to be accompanied with header data in a single file. The GGXF YAML text file format optionally permits the grid data to be held in an external file referenced from the GGXF YAML file. There is no restriction on the file format that data producers may use for such external grid files. In general these are not supported by GGXF. An optional very simple text format for external grid files which is supported by the GGXF format is defined in Table B.17. It is referenced from a GGXF YAML text file *dataSource* attribute when *dataSourceType* = *ggxf-csv*.

Table B.17 — GGXF conventions: external grid file format

GGXF identifier	Description
comma	Universal Coded Character Set (UCS) character code U+002C, used as the value separator in a ggxf-csv file.
ggxf-csv	Text file containing one header line followed by one line for each grid node in the row major order specified in req/yaml/gridData. Each line contains one or more values. The values in the header line are the grid node parameter identifiers from the GGXF Conventions (see Table B.4) and define the order of parameter values in the subsequent lines. The set of identifiers must include the grid node parameters defined in the ggxfGroup within which the external grid file is referenced. It may include node coordinate identifiers. The values on each subsequent line are the parameter values at a grid node in the order defined by the header line. Missing parameter values in the grid are identified by the *noDataFlag* value defined for the parameter in the GGXF file header. Each line must have the same number of values. Values must be separated by the character given through the *separator* key and optionally may be additionally padded with spaces. Each line is terminated by the line feed (new line) character (U+000A). The line feed may be immediately preceded by a carriage return character (U+000D)
separator	Character used as a value separator in the ggxf-csv text file. Supported values are *comma*, *space* and *tab*. Values are separated by either a single *comma* or *tab* (which may be padded with spaces) or by one or more *space* characters
space	Universal Coded Character Set (UCS) character code U+0020, used as the value separator in a ggxf-csv file.
tab	Universal Coded Character Set (UCS) code U+0009 (horizontal tabulation), used as the value separator in a ggxf-csv file.

See Annex E.1.3 for examples of the ggxf-csv file format.

Annex C
(informative)
Formulas

C.1 Affine transformation between GGXF grid and Interpolation CRS

For a 2D grid the conversion from grid indices (*i,j*) to Interpolation CRS coordinates (X,Y) is performed using the parametric form of an affine transformation:

$$X = A_0 + (A_1 . i) + (A_2 . j)$$
$$Y = B_0 + (B_1 . i) + (B_2 . j)$$

Here X and Y are the first and second coordinates in the Interpolation CRS tuple and must not be mistaken for axis abbreviations or imply positive direction of the axes. Alternatively, in matrix form:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} A_0 \\ B_0 \end{pmatrix} + \begin{pmatrix} A_1 & A_2 \\ B_1 & B_2 \end{pmatrix} * \begin{pmatrix} i \\ j \end{pmatrix}$$

By combining the translation matrix with the scale matrix this may be re-written as[3]:

$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} A_1 & A_2 & A_0 \\ B_1 & B_2 & B_0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

This combined form is convenient because the inverse of this matrix (for the conversion from Interpolation CRS coordinates to grid indices) has the same form. If there are many affine transformations to concatenate, their matrices can be multiplied and the result still has the same form. Note that:

- The order of elements in the leftmost matrix must match axis order of the Interpolation CRS.
- The order of elements in the rightmost matrix must match grid index order. In this document that order is fixed to (*i,j*) for all cases, regardless of the Interpolation CRS axis order.
- The values and signs of the elements of the middle matrix depend upon (a) the corner of the grid at which is the grid origin, (b) the positive directions of the axes of the Interpolation CRS, and (c) the order of the axes of the Interpolation CRS. There are sixteen permutations as shown in Table D.1.

Table C.1 — Matrix element values for conversion of grid indices to Interpolation CRS coordinates

Grid origin	Axes positive directions	Axis order	Matrix element A_1	A_2	A_0	B_1	B_2	B_0
Top left (northwest corner in a north-orientated grid)	North and east	North, East	0	$-\Delta N$	N_{max}	ΔE	0	E_{min}
		East, North	ΔE	0	E_{min}	0	$-\Delta N$	N_{max}
	South and east	South, East	0	ΔS	S_{min}	ΔE	0	E_{min}
		East, South	ΔE	0	E_{min}	0	ΔS	S_{min}
	South and west	South, West	0	ΔS	S_{min}	$-\Delta W$	0	W_{max}
		West, South	$-\Delta W$	0	W_{max}	0	ΔS	S_{min}
	North and west	North, West	0	$-\Delta N$	N_{max}	$-\Delta W$	0	W_{max}
		West, North	$-\Delta W$	0	W_{max}	0	$-\Delta N$	N_{max}

[3] These coefficients map to ESRI World file (.tfw) parameters A to F as: A_1=A, A_2=B, A_0=C, B_1=D, B_2=E and B_0=F.

Grid origin	Axes positive directions	Axis order	Matrix element A_1	A_2	A_0	B_1	B_2	B_0
Bottom left (southwest corner in a north-orientated grid)	North and east	North, East	0	ΔN	N_{min}	ΔE	0	E_{min}
		East, North	ΔE	0	E_{min}	0	ΔN	N_{min}
	South and east	South, East	0	-ΔS	S_{max}	ΔE	0	E_{min}
		East, South	ΔE	0	E_{min}	0	-ΔS	S_{max}
	South and west	South, West	0	-ΔS	S_{max}	-ΔW	0	W_{max}
		West, South	-ΔW	0	W_{max}	0	-ΔS	S_{max}
	North and west	North, West	0	ΔN	N_{min}	-ΔW	0	W_{max}
		West, North	-ΔW	0	W_{max}	0	ΔN	N_{min}

where

ΔN, ΔS, ΔE and ΔW are the grid node separations in Interpolation CRS units along the north or south and east or west axes.

N_{max}, S_{min}, E_{min} and W_{max} are the maximum or minimum coordinates of the Interpolation CRS envelope. They are always on the left-hand side of a north-up 2D envelope or geographic bounding box, at either the 'top left' or 'bottom left' corner.

For the conversion from Interpolation CRS coordinates to grid indices (*i*,*j*):

$$\begin{pmatrix} i \\ j \\ 1 \end{pmatrix} = \begin{pmatrix} A_1' & A_2' & A_0' \\ B_1' & B_2' & B_0' \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

Table C.2 — Matrix element values for conversion of Interpolation CRS coordinates to grid indices

Grid origin	Axes positive directions	Axis order	Matrix element A_1'	A_2'	A_0'	B_1'	B_2'	B_0'
Top left (northwest corner in a north-orientated grid)	North and east	North, East	0	1/ΔE	$-E_{min}$ /ΔE	-1/ΔN	0	N_{max} /ΔN
		East, North	1/ΔE	0	$-E_{min}$ /ΔE	0	-ΔN	N_{max} /ΔN
	South and east	South, East	0	1/ΔE	$-E_{min}$ /ΔE	1/ΔS	0	$-S_{min}$ /ΔS
		East, South	1/ΔE	0	$-E_{min}$ /ΔE	0	1/ΔS	$-S_{min}$ /ΔS
	South and west	South, West	0	-1/ΔW	W_{max} /ΔW	1/ΔS	0	$-S_{min}$ /ΔS
		West, South	-1/ΔW	0	W_{max} /ΔW	0	1/ΔS	$-S_{min}$ /ΔS
	North and west	North, West	0	-1/ΔW	W_{max} /ΔW	-1/ΔN	0	N_{max} /ΔN
		West, North	-1/ΔW	0	W_{max} /ΔW	0	-1/ΔN	N_{max} /ΔN
Bottom left (southwest corner in a north-orientated grid)	North and east	North, East	0	1/ΔE	$-E_{min}$ /ΔE	1/ΔN	0	$-N_{min}$ /ΔN
		East, North	1/ΔE	0	$-E_{min}$ /ΔE	0	-ΔN	$-N_{min}$ /ΔN
	South and east	South, East	0	1/ΔE	$-E_{min}$ /ΔE	-1/ΔS	0	S_{max} /ΔS
		East, South	1/ΔE	0	$-E_{min}$ /ΔE	0	-1/ΔS	S_{max} /ΔS
	South and west	South, West	0	-1/ΔW	W_{max} /ΔW	-1/ΔS	0	S_{max} /ΔS
		West, South	-1/ΔW	0	W_{max} /ΔW	0	-1/ΔS	S_{max} /ΔS
	North and west	North, West	0	-1/ΔW	W_{max} /ΔW	1/ΔN	0	$-N_{min}$ /ΔN
		West, North	-1/ΔW	0	W_{max} /ΔW	0	1/ΔN	$-N_{min}$ /ΔN

For 3D (or higher) dimension grids, a row and a column are added for each additional dimension. For dimension n, the affine transform is a square matrix of size $(n+1)\times(n+1)$. For a 3D GGXF grid with i columns, j rows and k planes, for the conversion grid indices (i,j,k) to Interpolation CRS coordinates (X,Y,Z) the matrix is:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} A_1 & A_2 & A_3 & A_0 \\ B_1 & B_2 & B_3 & B_0 \\ C_1 & C_2 & C_3 & C_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} i \\ j \\ k \\ 1 \end{pmatrix}$$

Here X, Y and Z are the first, second and third coordinates in the Interpolation CRS tuple and must not be mistaken for axis abbreviations or imply positive direction of the axes.

Then for the conversion from Interpolation CRS coordinates to grid indices:

$$\begin{pmatrix} i \\ j \\ k \\ 1 \end{pmatrix} = \begin{pmatrix} A_1' & A_2' & A_3' & A_0' \\ B_1' & B_2' & B_3' & B_0' \\ C_1' & C_2' & C_3' & C_0' \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

C.2 Offsets

A common source of confusion in coordinate transformations is whether a transformation is defined by the offset of the coordinate axes in which coordinates are defined, or by the consequent change to the coordinates representing a location. These definitions are numerically equal but of opposite sign. It is impossible to completely separate the content of a grid file from applications that write or read that content. Consequently, there are constraints on GGXF file content that are set by expectation of the reading application's behaviour. One of these is the sign of the direction in which offsets are applied.

Mathematically, if the origin of a one-dimensional coordinate system is shifted along the positive axis and placed at a point with ordinate A, then the transformation formula is:

$X_{new} = X_{old} - A$

However, it is common practice in coordinate reference system transformations to apply the shift as an addition, with the sign of the shift parameter value having been suitably reversed to compensate for the practice. Hence transformations allow calculation of coordinates in the target system by <u>adding</u> a correction parameter to the coordinate values of the point in the source system:

$X_t = X_s + A$

where X_s and X_t are the values of the coordinates in the source and target coordinate systems and A is the value of the transformation parameter to transform source coordinate reference system coordinate to target coordinate reference system coordinate.

Offset methods are reversible. For the reverse transformation, the offset parameter value is applied with its sign reversed.

Not all coordinate operations methods utilise offsets. For example a geoid model utilises the well-known geodetic relationship $h = H + N$ and to obtain the gravity-related height H the geoid height N interpolated from a GGXF file is subtracted from the ellipsoidal height h, that is $H = h - N$.

The application of offsets is included in Annex E examples E.3 and E.4, whilst E.2 exemplifies a GGXF file containing geoid heights.

Annex D
(informative)

Consolidated list of recommendations

Recommendation 1 (clause 5.3)	When the Interpolation CRS is geographic, if it is desired to make the spacing of latitude and longitude node intervals similar in linear distance, make the node spacing in angular units in longitude ($\Delta\lambda$) approximately equal to $\Delta\varphi/\cos(\varphi_m)$ where $\Delta\varphi$ is the node spacing in latitude and φ_m is the latitude of the middle of the grid.
Recommendation 2 (clause 5.7)	Where there is a transition from one grid being used for interpolation to another, to ensure continuity the parameter values interpolated from the grid on one side of the boundary should be the same as those interpolated from the grid on the other side.
Recommendation 3 (clause 5.8.3)	When parameters are in units documented in the GGXF Conventions, the unit ID and ratio to SI standard unit as given in the Conventions should be used.
Recommendation 4 (clause 5.8.3)	Use arc-minutes or arc-seconds as the unit for parameters that are sexagesimal divisions of a degree.
Recommendation 5 (clause 5.8.5)	To assist implementations reading the GGXF file, data producers are discouraged from using the no data flag and instead to populate all nodes of a grid with estimated values, giving extrapolated values a high uncertainty.
Recommendation 6 (clause 5.8.5)	To cover an irregularly-shaped area and avoid a large number of nodes with no data, use multiple root grids.
Recommendation 7 (clause 5.8.5)	When interpolating grids values, implementations should raise an exception, return a no-data value, or otherwise alert users to calculations which require interpolating a no-data value.
Recommendation 8 (clause 5.8.6)	To minimise file size, coordinates of the grid nodes should not be included as parameters in grids in a GGXF binary file. However, producers creating GGXF binary files using a GGXF YAML text file with ggxf-csv external grid files are encouraged to include node coordinates in the ggxf-csv grid files to allow software to validate the row and column counts and the affine coefficients. These node coordinates are not required to be included in the resultant GGXF binary file.
Recommendation 9 (clause 5.8.9)	When all parameters given in the file header are used in the ggxfGroup, the *gridParameters* attribute should be omitted from the ggxfGroup header.
Recommendation 10 (clause 5.10.1)	The method that is recommended to be used for interpolation of the grid(s) in a ggxfGroup should be given in the ggxfGroup header.
Recommendation 11 (clause 5.10.2)	A GGXF file with content supporting coordinate transformation should indicate the nominal operation accuracy value applicable to the data in the file.
Recommendation 12 (clause 5.10.3)	The producer and distributer of the file, file date of issue, and if applicable a Digital Object Identifier (DOI) for the file content should be included in the GGXF file header metadata

Recommendation 13 (clause 5.10.4)	Metadata describing the terms under which the GGXF file is distributed (license information) should be included in the file header.
Recommendation 14 (clause 6.2.1)	Producers are encouraged to publish data using the GGXF binary format rather than the GGXF text format.
Recommendation 15 (clause 6.2.1)	For large grids producers are encouraged to have these grids in external files rather than directly in the YAML text file.
Recommendation 16 (clause 6.3.2)	When reading a GGXF binary file any attributes encountered that are not in the GGXF Conventions may be ignored.
Recommendation 17 (clause 6.3.5.1)	Consider defining different *parameterSet*s for parameters and their uncertainties, for example 'offset' and 'offsetUncertainty'. Applications not needing to utilise the uncertainty data can then skip the offsetUncertainty variable.
Recommendation 18 (clause 6.3.5.1)	In creating the name of a parameter set, avoid 'count' as this is may create a conflict with its use in the definition of structured data and in netCDF vector variable naming.
Recommendation 19 (clause 6.3.5.3)	It is recommended that the dimension defined in netCDF group for the number of parameters in a grid variable should be the parameterSet name appended with “Count”. For example, the dimension of the parameterSet "displacement" should be named “displacementCount".
Recommendation 20 (clause 6.3.5.5)	If a grid has missing values a producing application should determine a suitable *missing_value* for each grid data variable it creates. It should use this value in place of the noDataFlag in the variable. The *missing_value* attribute of a variable should be either larger than the largest actual packed data value or smaller than the smallest actual packed data value in the variable.
Recommendation 21 (clause 6.3.5.5)	A reading application encountering values matching the *missing_value* attribute in a variable should identify and treat the corresponding unpacked values as undefined.
Recommendation 22 (Annex B Table B.8)	To avoid problems with calendar arithmetic, use epoch rather than date.

Annex E
(informative)

Examples

These text examples are given following the YAML syntax requirements. Attributes that are character strings are given within double quotation marks when free text, but character strings that are values from the GGXF Conventions are unquoted.

This GGXF specification requires that GGXF YAML files include either a full grid or a reference to an external file containing the grid. However, when for illustration purposes the example includes an extract from the full grid, in the example the file extension is changed from ".yaml" to ".gxt"; then only the header records in the example are valid GGXF text file content.

Note that in YAML indentations are significant.

Because of their file size most GGXF files will have a binary encoding. Examples of GGXF binary files are given in ... *(link to Github)*

The features illustrated in these examples include but are not limited to:

Feature	**Example**						
	E.1	**E.2**	**E.3**	**E.4**	**E.5**	**E.6**	**E.7**
Deformation model					x		
Geoid model		x					
Latitude and longitude offsets	x		x				
Velocity grid				x			x
Deviations (deflections) of the vertical (gridded geodetic data not used directly in transformations)						x	
Accuracy as a gridded parameter			x				
Affine coefficients	x	x	x	x	x	x	
Bounding box	x	x	x	x	x	x	
Bounding polygon	x						
Citation of information source		x	x		x	x	
ggxfGroups: single ggxfGroup	x	x		x		x	
ggxfGroups: multiple ggxfGroups							
Grids: single grid		x				x	
Grids: multiple non-overlapping grids	x						
Grids: nested grids			x	x	x		
Grid priority							x
Grid data: in YAML file	x						
Grid data: in external file					x		
Grid data: extract (example, non-compliant)		x	x	x			
Parameters: single parameter at a node		x					
Parameters: multiple parameters at a node	x		x	x	x	x	
Parameters: variety across ggxfGroups					x		
Parameters: constant value					x		

E.1 GGXF file basics

This simple hypothetical example includes two butt-joining grids with the longitudinal spacing of the northern grid increased in comparison to that of the southern grid. The grid node indices *i,j* for perimeter nodes of each grid are shown in Figure E.1, with the interpolation CRS graticule coordinates also shown. The content applicability extent is shown as a green dashed polygon. The parameters in the grid (latitude and longitude offset) are contoured diagrammatically in Figure E.2 so that they may be compared with the row-major values in the grid array at the end of the example.

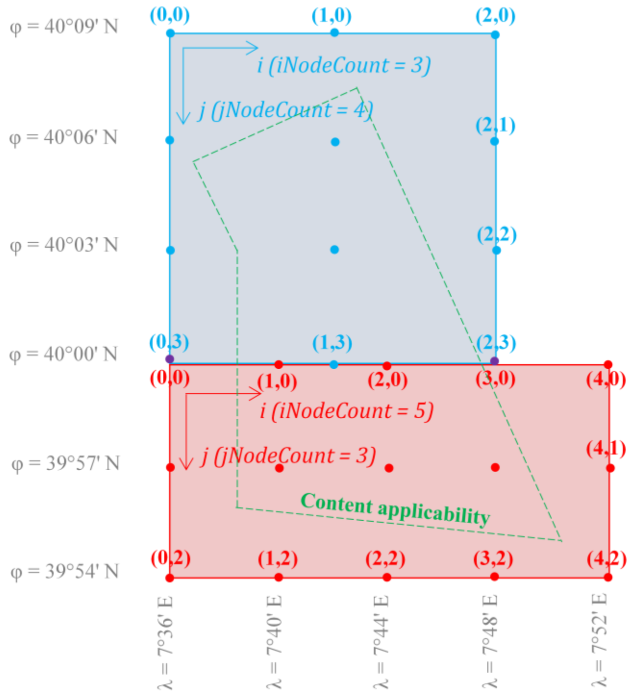


Figure E.1 — Butt-joining grid extents

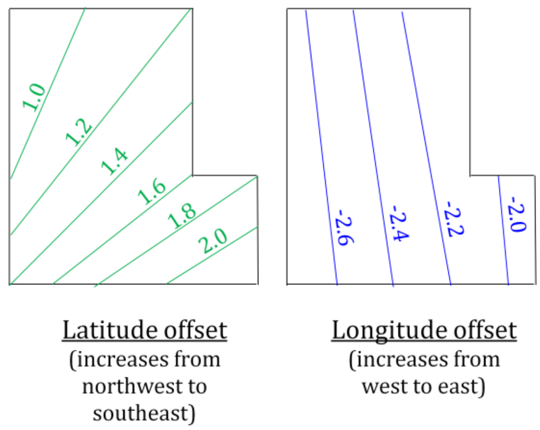


Figure E.2 — Gridded data

Commentary

In this example, the interpolation CRS and the source CRS are the same. Note that grid indices are zero-based, but in the grid header the NodeCount attributes are 1-based. So the maximum node index value = (nodeCount - 1)

From the interpolation CRS WKT definition, the CRS type is geographic and its units can be seen to be degrees. Then, from the affine coefficients in South grid header, because the rotation values in this affine transformation (the second and sixth coefficents) are zero, from the remaining coefficients and axis node count (which equals the number of intervals between nodes +1) the grid envelope lower corner can be determined to be at ((40.0 + ((3-1) * -0.05)), (7.6 + ((5-1) * 0.0))) = (39.9°, 7.6°) = 39°54'N, 7°36'E and the envelope upper corner to be at ((40.0 + ((3-1) * 0.0)), (7.6 + ((5-1) * 0.0666666666666666)) = (40.0°, 7.8666666666666666°) = 40°00'N 7°52'E. A similar calculation may be performed for the North grid.

The grid data is shown in rows and columns for illustration, with both parameters included in one array. The latitude offsets (which increase from northwest to southeast) are in green and illustrate that the grid is populated from the 'top left' (northwest) node in a row-major order.

It is recommended that contact details for the producer and distributer of the file and file date of issue should be included as citation metadata in the file header. In this example such data is omitted; this is for brevity of the illustrated records.

E.1.1 YAML representation


```
ggxfVersion: "GGXF-1.0"
content: geographic2dOffsets
title: "Catalino Canyon transformation version 2022-06"
abstract: "Example transformation constructed for purposes of illustration."
filename: "Catalano_Canyon.yaml"

contentApplicabilityExtent:
  extentDescription: "Italy - Mediterranean Sea west of Sardinia - Catalano Canyon."
  boundingBox:
    southBoundLatitude: 39.9
    westBoundLongitude: 7.6
    northBoundLatitude: 40.15
    eastBoundLongitude: 7.87
  boundingPolygon: Polygon(( 40.09, 7.72, 40.12, 7.71, 39.92, 7.84, 39.93, 7.64,
                             40.05, 7.64, 40.09, 7.72 ))

interpolationCrsWkt: &crs |
    GEOGCRS["ED50",
      DATUM["European Datum 1950",
        ELLIPSOID["International 1924",6378388,297,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,2],
      AXIS["Geodetic latitude (Lat)",north],AXIS["Geodetic longitude (Lon)",east],
      ANGLEUNIT["degree",0.0174532925199433]]

sourceCrsWkt: *crs

targetCrsWkt: |
    GEOGCRS["ETRF2000",
      DATUM["European Terrestrial Reference Frame 2000",
        ELLIPSOID["GRS 1980",6378137,298.257222101,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,2],
      AXIS["Geodetic latitude (Lat)",north],
      AXIS["Geodetic longitude (Lon)",east],
      ANGLEUNIT["degree",0.0174532925199433]]

operationAccuracy: 2

parameters:
  - parameterName: latitudeOffset
    parameterSet: "offset"
    sourceCrsAxis: 0
    unitName: arc-second
    unitSiRatio: 4.84813681109536E-06
  - parameterName: longitudeOffset
    parameterSet: "offset"
    sourceCrsAxis: 1
    unitName: arc-second
    unitSiRatio: 4.84813681109536E-06

ggxfGroups:
    - ggxfGroupName: "Catalano_Canyon"
      interpolationMethod: bilinear

      grids:
        - gridName: "South"
          affineCoeffs: [ 40.0000000, 0.0, -0.05, 7.6000000, 0.0666666666666667, 0.0 ]
          iNodeCount: 5
          jNodeCount: 3
          data: [ 1.00, -2.70, 1.20, -2.50, 1.40, -2.30, 1.60, -2.10, 1.80, -1.90,
                  1.20, -2.74, 1.40, -2.52, 1.65, -2.31, 1.83, -2.13, 2.00, -1.92,
                  1.40, -2.78, 1.80, -2.53, 2.00, -2.33, 2.10, -2.14, 2.20, -1.93 ]

        - gridName: "North"
          affineCoeffs: [ 40.15, 0.0, -0.05, 7.6, 0.1, 0.0 ]
          iNodeCount: 3
          jNodeCount: 4
```

```
          data: [ 0.86, -2.62, 1.06, -2.34, 1.20, -2.04,
                  0.91, -2.64, 1.13, -2.36, 1.30, -2.06,
                  0.95, -2.66, 1.20, -2.38, 1.50, -2.08,
                  1.00, -2.70, 1.30, -2.40, 1.60, -2.10 ]
```

See E1.3 for an example of these grids being referenced externally through the *gridSource* attribute.

E.1.2 netCDF representation

The full GGXF netCDF file for this example is available on GitHub here. A CDL file with the GGXF netCDF header records for this example is available on GitHub here. These have been produced from the YAML data in E.1.1 above; that data is also on Github here.

E.1.3 External grid files

In the building of GGXF files, if a data producer chooses to use a GGXF YAML text file in conjunction with external grid files in the simple ggxf-csv text format supported by GGXF, the ggxf Group and its grid definitions in E.1.1 would be replaced by:

```
ggxfGroups:
    - ggxfGroupName: "Catalano_Canyon"
      interpolationMethod: bilinear
      grids:
        - gridName: "South"
          affineCoeffs: [ 40.0000000, 0.0, -0.05, 7.6000000, 0.0666666666666667, 0.0 ]
          iNodeCount: 5
          jNodeCount: 3
          dataSource:
            gridFilename: "Catalano_Canyon_South.csv"
            dataSourceType: ggxf-csv
            separator: comma

        - gridName: "North"
          affineCoeffs: [ 40.15, 0.0, -0.05, 7.6, 0.1, 0.0 ]
          iNodeCount: 3
          jNodeCount: 4
          dataSource:
            gridFilename: "Catalano_Canyon_North.txt"
            dataSourceType: ggxf-csv
            separator: space
```

Then the external grid files are:

Catalano_Canyon_South.csv in ggxf-csv format with comma as separator:

```
nodeLatitude,nodeLongitude,latitudeOffset,longitudeOffset
40.0000000,7.6000000,1.00,-2.70
40.0000000,7.6666667,1.20,-2.50
40.0000000,7.7333333,1.40,-2.30
40.0000000,7.8000000,1.60,-2.10
40.0000000,7.8666667,1.80,-1.90
39.9500000,7.6000000,1.20,-2.74
39.9500000,7.6666667,1.40,-2.52
39.9500000,7.7333333,1.65,-2.31
39.9500000,7.8000000,1.83,-2.13
39.9500000,7.8666667,2.00,-1.92
39.9000000,7.6000000,1.40,-2.78
39.9000000,7.6666667,1.80,-2.53
39.9000000,7.7333333,2.00,-2.33
39.9000000,7.8000000,2.10,-2.14
39.9000000,7.8666667,2.20,-1.93
```

Catalano_Canyon_North.txt in ggxf-csv format with space as separator:

```
nodeLatitude nodeLongitude latitudeOffset longitudeOffset
40.15 7.60 0.86 -2.62
40.15 7.70 1.06 -2.34
40.15 7.80 1.20 -2.04
40.10 7.60 0.91 -2.64
40.10 7.70 1.13 -2.36
40.10 7.80 1.30 -2.06
40.05 7.60 0.95 -2.66
40.05 7.70 1.20 -2.38
40.05 7.80 1.50 -2.08
40.00 7.60 1.00 -2.70
40.00 7.70 1.30 -2.40
40.00 7.80 1.60 -2.10
```

E.1.4 Parameter indexing and interpolation

In GGXF, parameter values are sequenced along each of the rows (count across *i* columns) then along each of the columns (count across *j* rows) (and in the 3-D case then along each of the planes (*k*)), as shown for the two-dimensional Catalano Canyon North and South grids in Figure E.3.

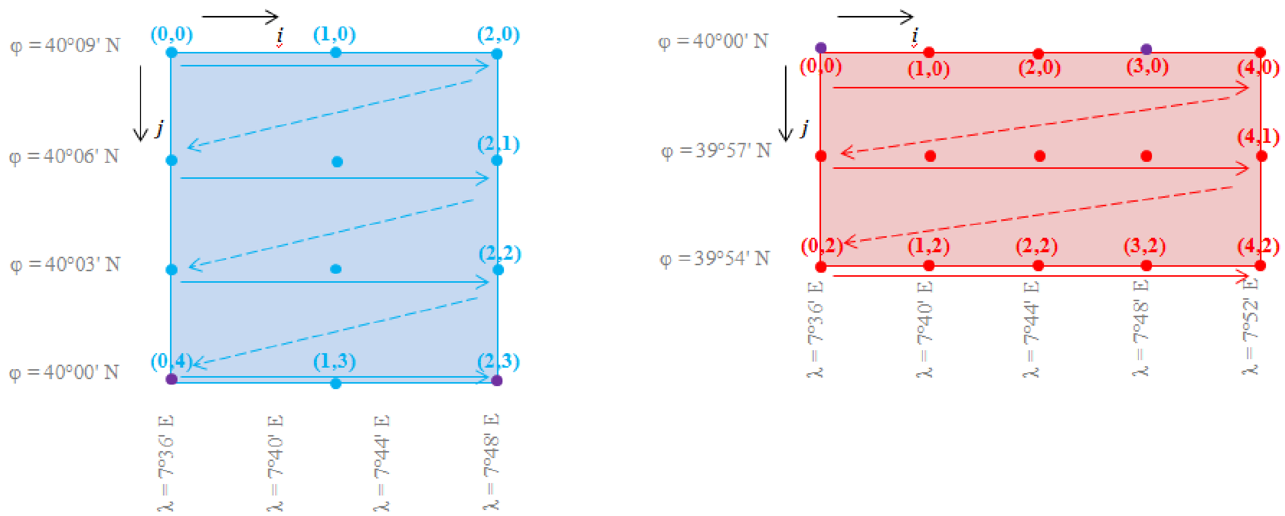


Figure E.3 — Array sequencing

The ggxfGroup header record declares that the interpolation method to be used for these Catalona Canyon grids is bilinear. Therefore parameter values from the surrounding four grid nodes are required for interpolation.

Assume that a coordinate transformation is required for source CRS location 39°58'N, 7°42'E. By applying the affine transformation to each grid in turn, this location is found to not fall in the North grid and to fall in the South grid at grid values *i*=1.5, *j*=0.7. Surrounding South grid nodes at *i*=1 and *i*=2 with *j*=0 and *j*=1 are required. The indices for these four nodes are indexed on [row(j),column(i),parameter]. From requirement req/yaml/gridData the location of the p^th parameter at node (i,j) is given by the formula

$$p + n_p \times (i + n_i \times j)$$

In the array

```
[ 1.00, -2.70, 1.20, -2.50, 1.40, -2.30, 1.60, -2.10, 1.80, -1.90,
  1.20, -2.74, 1.40, -2.52, 1.65, -2.31, 1.83, -2.13, 2.00, -1.92,
  1.40, -2.78, 1.80, -2.53, 2.00, -2.33, 2.10, -2.14, 2.20, -1.93 ]
```

(shown here by row for reader convenience) the indices for the first parameter (latitudeOffset) then are 3, 5, 13 and 15, and for the second parameter (latitudeOffset) are 4, 6, 14 and 16.

The parameter values at the four grid nodes surrounding the required location

i=1, j=0 i=2, j=0
i=1, j=1 i=2, j=1

which are read from the array are shown in figure E.4, with values for parameter 1 (latitudeOffset) in green and values for parameter 2 (latitudeOffse) in blue.

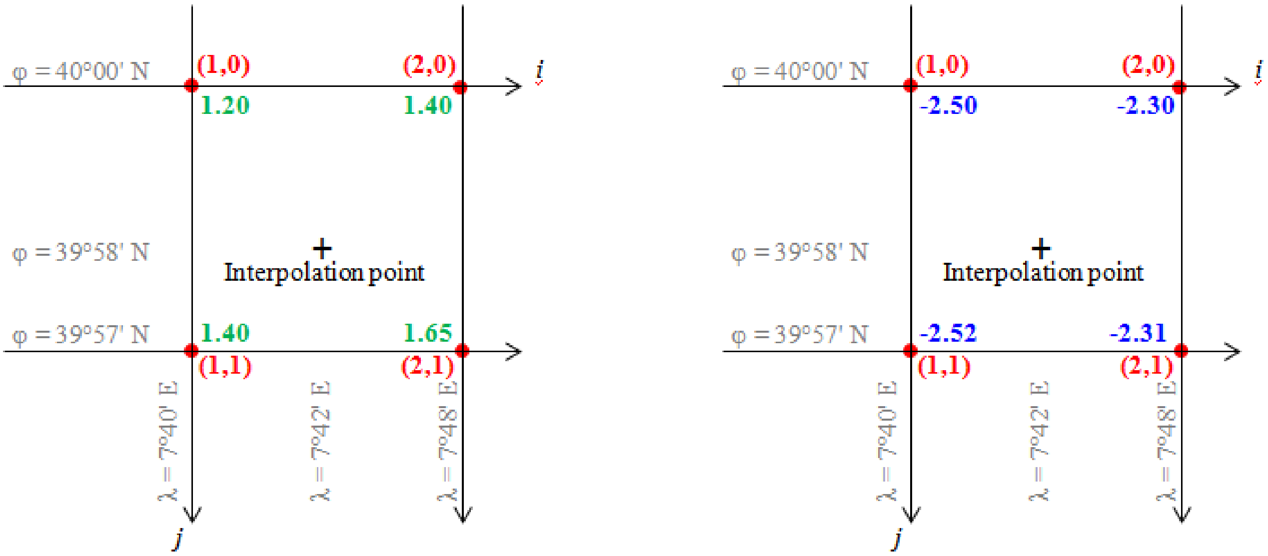


Figure E.4 — Parameter values to be interpolated

Bilinear interpolation for parameter 1 (latitudeOffset) gives δφ = 1.45000 arc-seconds. For parameter 2 (longitudeOffset), δλ = -2.40333 arc-seconds.

Then applying the interpolated offsets to the source CRS coordinates, coordinates transformed into the target CRS are:

39°58' + 1.45000"   = 39°58'01.450"N
7°42' + (-2.40333)" = 7°41'57.597"E

E.2 Geoid model

This example of a single grid uses extracts from the South Africa geoid 2010. The example illustrates:

- GGXF header records;
- extracts from the geoid model grid;
- use of the affine transformation to determine the grid cell in which a location falls;
- interpolation of the parameter value (hybrid geoid height) at the location.

The area covered by the grid is illustrated in Figure E.1 below. This may be described in the header through the contentBox records, but it is unnecessary to do so because it is described through the affine transformation and node counts in the grid header. The area of applicability of the grid is restricted to South Africa, as outlined in red. This is described in the header through the mandatory contentApplicabilityExtent bounding box (dashed green line) and textual description records. The location of a point of interest for which the geoid height parameter is interpolated in the grid is shown in green. Its interpolation is described in the commentary at the end of the example.

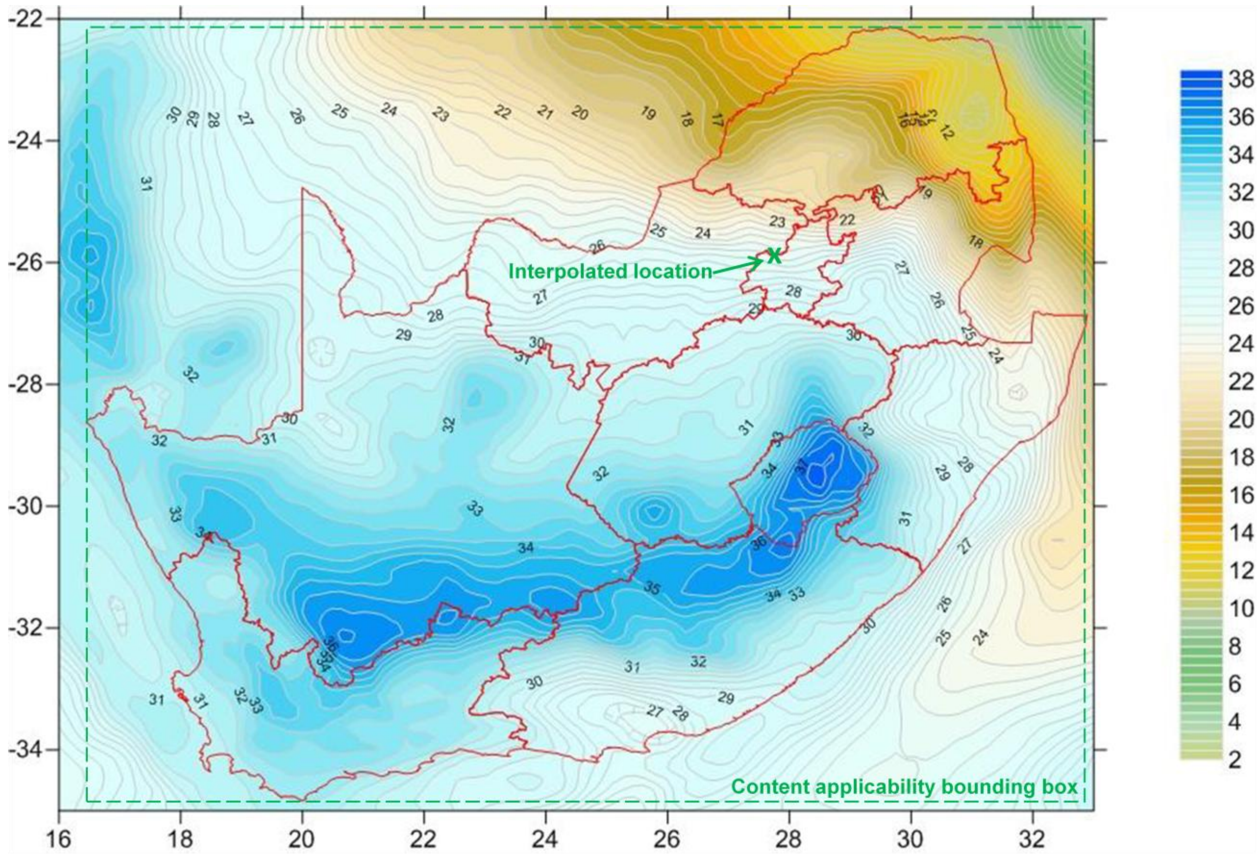


Figure E.1 — Geoid model grid extent and applicability

E.2.1 YAML representation


```yaml
ggxfVersion: "GGXF-1.0"
content: geoidModel
comment: "hybrid geoid"
title: "South_African_geoid_2010"
abstract: "Model for converting ellipsoidal heights determined using NGI's TrigNet to
orthometric heights on the South African Land Levelling Datum, to an accuracy of 10 cm
(design requirements). Accuracy is 7cm absolute and relative <2cm + GNSS related error."
filename: "SAGEOID2010.yaml"

contentApplicabilityExtent:
    boundingBox:
        southBoundLatitude:  -34.89
        westBoundLongitude:   16.45
        northBoundLatitude:  -22.13
        eastBoundLongitude:   32.96
    extentDescription: "South Africa - mainland onshore."

interpolationCrsWkt: |
    GEOGCRS["ITRF2005",
      DYNAMIC[FRAMEEPOCH[2000.0]],
        DATUM["International Terrestrial Reference Frame 2005",
          ELLIPSOID["GRS 1980",6378137,298.2572221,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,2],
      AXIS["Geodetic latitude (Lat)",north],
      AXIS["Geodetic longitude (Lon)",east],
      ANGLEUNIT["degree",0.0174532925199433]]

sourceCrsWkt: |
    GEOGCRS["ITRF2005",
      DYNAMIC[FRAMEEPOCH[2000.0]],
        TRF["International Terrestrial Reference Frame 2005",
          ELLIPSOID["GRS 1980",6378137,298.2572221,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,3],
      AXIS["Geodetic latitude (Lat)",north,
        ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Geodetic longitude (Lon)",east,
        ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Ellipsoidal height (h)",up,LENGTHUNIT["metre",1]]]

targetCrsWkt: |
    VERTCRS["VI LLD height",
      VDATUM["South Africa Land Levelling Datum"],
      CS[vertical,1],
      AXIS["Gravity-related height (H)",up],
      LENGTHUNIT["metre",1]]

operationAccuracy:      0.07

partyName:              "Chief Directorate: National Geospatial Information"
deliveryPoint:          "Private Bag X10"
city:                   "Mowbray"
postalCode:             "7705"
country:                "South Africa"
onlineResourceLinkage:  "ftp://ftp.trignet.co.za/South_African_Geoid"

parameters:
    - parameterName: geoidHeight
      sourceCrsAxis: 2
      unitName: metre
      unitSiRatio: 1.0

ggxfGroups:
    - ggxfGroupName: "SA geoid 2010"
      interpolationMethod: bilinear
      grids:
          - gridName: "SA geoid 2010"
```

```
            affineCoeffs: [-35.0, 0.0, 0.04166666666667, 16.0, 0.04166666666667, 0.0]
            iNodeCount: 409
            jNodeCount: 313
            data: [ 26.055, 26.121, ... 3.866,  3.826 ]
```

(shown as tabulation of rows and columns for illustration)

```
#  i=0     i=1         i=280         i=281       i=407    i=408

[ 30.136, 30.158, ... 13.591,      13.526, ...  3.866,  3.826,      # j=0    φ=-22°
  30.194, 30.198, ... 13.674,      13.615, ...  3.926,  3.891,      # j=1
    :       :           :             :           :       :
  33.958, 34.062, ... 25.452,      25.417, ... 16.143, 15.909,      # j=93
#                               X                                     φ=-25.9°, j=93.6
  33.888, 34.839, ... 25.640,      25.583, ... 16.309, 16.074,      # j=94
    :       :           :             :           :       :
  26.050, 26.114, ... 26.630,      26.609, ... 29.725, 29.768,      # j=311
  26.055, 26.121, ... 26.618,      26.599, ... 29.745, 29.789 ]     # j=312  φ=-35°

# λ=16°                      λ=27.7°                        λ=33°
#                           i=280.8
```

Commentary

As there is only one parameter, the attribute *parameterSet* is redundant and not used.

From the interpolation CRS WKT definition the CRS type is geographic and its units can be seen to be degrees. Then, as the rotation values in this affine transformation are zero, from the affine coefficients and axis count the grid envelope lower corner can be determined to be at ((-22.0 + (313 - 1) * -0.04166666667),(16 + (409 - 1) * 0.0)) = (35°S, 16°E) and the grid envelope upper corner to be at ((-22.0 + (313 - 1) * 0.0),(16.0 + (409 - 1) * 0.04166666667)) = (22°S, 33°E). In this single grid example, these derived values are the same as those that could be given through the optional metadata describing the extent of all grids in the file (*contentBox*).

To interpolate the grid for the geoid height at ITRF2005 position (25°54'S, 27°42'E) = (-25.9, 27.7), use the affine transformation to determine that the grid indices for this location are i=280.8, j=93.6. The grid nodes surrounding this location and the geoid height are shown in blue in the grid array above. Then, using the declared interpolation method (bilinear), the geoid height at the location is interpolated between these node values and evaluated as 25.526 metres.

Referring to the GGXF Conventions (Annex B.3) for the definition of content type *geoidModel* for the formula ($H_T = h_S - C$), for the point at 25°54'S, 27°42'E with an ITRF2005 ellipsoidal height (h_S) of 1450m, the SA LLD height (H_T) is (1450.000 - 25.526) = 1424.474 metres.

The ggxf-csv file with the full grid data for the full South Africa geoid 2010 is available on Github here. The GGXF YAML text file is available on Github here.

E.2.2 netCDF representation

The full South Africa geoid 2010 file in the GGXF netCDF binary file format is available on GitHub here. A CDL file with the GGXF netCDF header records for the South Africa geoid 2010 file is available on GitHub here.

Further examples of GGXF files for geoid models for Puerto Rico and the US Virgin Islands are available on Github here. In these one ggxf-csv file covers both Puerto Rico and the US Virgin Islands; there are separate GGXF files for the two areas as they have different vertical datums.

E.3 Geographic 2D offsets with accuracy

This example uses extracts from the NTv2 grid for Canada. Parameters at the grid nodes are latitude and longitude offsets and their accuracies. The complete NTv2_0 file consists of four butt-joining grids (east, west, north and Arctic), the first three of which have a total of 110 nested child grids. This example shows two of the parent grids (west and east) and three butt-joining child grids (Sarnia, Toronto and Windor) The grid node spacing is 5 x 5 arc-minutes for the parent grids and 30 x 30 arc-seconds for child grids. The child grids share common boundaries. The example assumes a study area bounded by 42°24'36"N, 81°45'18"W in the southwest, and 42°25'24"N, 81°44'42"W in the northeast. The study area requires grid data to be interpolated from all three child grids and their parent (see Figure E.3).

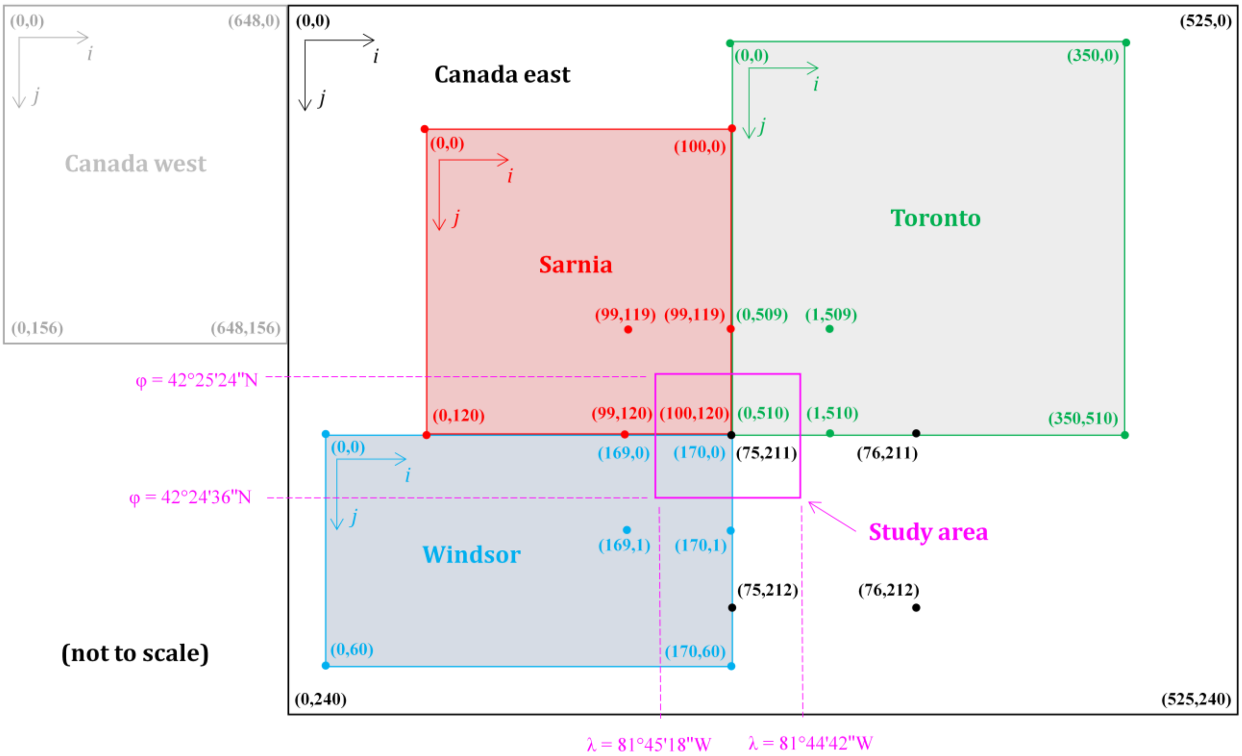


Figure E.3 — Extents for grids in this example

E.3.1 YAML representation


```
ggxfVersion: "GGXF-1.0"
content: geographic2dOffsets
title: "National Transformation v2_0"
abstract: "Transformation of geodetic latitude and longitude referenced to NAD27 to latitude and longitude referenced to NAD83(Original)."
filename: "27to83.yaml"

publicationDate: "1995-02"
partyName:              "Geodetic Survey Division, Natural Resources Canada"
onlineResourceLinkage: "https://webapp.geod.nrcan.gc.ca/geod/data-donnees/transformations.php"
licenseURL:               "https://open.canada.ca/en/open-government-licence-canada"
contentApplicabilityExtent:
  boundingBox:
    southBoundLatitude: 40.0
    westBoundLongitude: -141.0
    northBoundLatitude: 60.0
    eastBoundLongitude: -88.0
  extentDescription: "Canada south of 60°N"
```

```
interpolationCrsWkt: &crs |
     GEOGCRS["NAD27",
       DATUM["North American Datum 1927",
         ELLIPSOID["Clarke 1866",6378206.4,294.9786982,LENGTHUNIT["metre",1]]],
       CS[ellipsoidal,2],
       AXIS["Geodetic latitude (Lat)",north],
       AXIS["Geodetic longitude (Lon)",east],
       ANGLEUNIT["degree",0.0174532925199433]]

sourceCrsWkt: *crs

targetCrsWkt: |
     GEOGCRS["NAD83(Original)",
       DATUM["North American Datum 1983",
         ELLIPSOID["GRS 1980",6378137,298.2572221,LENGTHUNIT["metre",1]]],
       CS[ellipsoidal,2],
       AXIS["Geodetic latitude (Lat)",north],
       AXIS["Geodetic longitude (Lon)",east],
       ANGLEUNIT["degree",0.0174532925199433]]

operationAccuracy: 1.5

parameters:
  - parameterName: latitudeOffset
    parameterSet: "offset"
    sourceCrsAxis: 0
    unitName: arc-second
    unitSiRatio: 4.84813681109536E-06
  - parameterName: longitudeOffset
    parameterSet: "offset"
    sourceCrsAxis: 1
    unitName: arc-second
    unitSiRatio: 4.84813681109536E-06
  - parameterName: latitudeOffsetUncertainty
    parameterSet: "offsetUncertainty"
    sourceCrsAxis: 0
    unitName: metre
    unitSiRatio: 1.0
    uncertaintyMeasure: "2SE"
  - parameterName: longitudeOffsetUncertainty
    parameterSet: "offsetUncertainty"
    sourceCrsAxis: 1
    unitName: metre
    unitSiRatio: 1.0
    uncertaintyMeasure: "2SE"

checkPoints:
  - sourceCrsCoordinates:[ 43.06, -82.37 ]
    targetCrsCoordinates: [ 43.060037667, -82.369913139 ]
    parameterCheckValues:
      latitudeOffsetUncertainty: 0.0073
      longitudeOffsetUncertainty: 0.0037

ggxfGroups:
  - ggxfGroupName: "National Transformation v2_0"
    interpolationMethod: bilinear
    grids:
      - gridName: "CAwest"
        affineCoeffs: [ 60.0, 0.0, -0.0833333, -142.0, 0.0833333, 0.0 ]
        iNodeCount: 649
        jNodeCount: 157
        data:
# Grid 1 data extract (dLat, dLon, uncertaintyLat, uncertaintyLon)
# First row  i,j = (0,0) ... i,j = (648,0) (along 60°N from 142°W to 88°W)
# Last row   i,j = (0,156) ... i,j = (648,156) (along 47°N from 142°W to 88°W)
          [ -1.369, -7.569, 0.597, 1.118, ... 0.743,  0.420, 0.898, 1.314,
               :       :      :      :           :       :      :      :
```

```
          -2.294, -8.433, 0.624, 0.418, ... 0.176, -0.835, 0.012, 0.023 ]

      - gridName: "CAeast"
        affineCoeffs: [ 60.0, 0.0, -0.0833333, -88.0, 0.0833333, 0.0 ]
        iNodeCount: 529
        jNodeCount: 241
        data:
[ 0.743,  0.420, 0.898, 1.314,            ...                -0.190, 6.536, 0.140, 0.067,
    :      :      :      :      :      :      :      :          :      :      :      :
... 0.164, 0.416, 0.019, 0.034, 0.164, 0.417, 0.017, 0.033,... 0.201, 0.868, 0.014, 0.004,
... 0.165, 0.418, 0.025, 0.046, 0.164, 0.417, 0.018, 0.036,... 0.200, 0.865, 0.014, 0.004,
   0.381, -0.529, 0.028, 0.031,           ...                 2.705, 5.47,8 0.118, 0.136 ]

        childGrids:
          - gridName: "ONtronto"
            affineCoeffs: [ 46.6666667, 0.0, -0.0083333, -81.75, 0.1, 0.0083333 ]
            iNodeCount: 351
            jNodeCount: 511
            data:
[ 0.141, 0.353, 0.154, 0.116, 0.141, 0.354, 0.141, 0.111, ... 0.205, 0.824, 0.031, 0.045,
   :      :      :      :      :      :      :      :          :      :      :      :
  0.164, 0.416, 0.019, 0.034, 0.164, 0.417, 0.017, 0.033, ... 0.201, 0.868, 0.014, 0.004,
  0.165, 0.418, 0.025, 0.046, 0.164, 0.417, 0.018, 0.036, ... 0.200, 0.865, 0.014, 0.004 ]

          - gridName: "ONsarnia"
            affineCoeffs: [ 43.4166667, 0.0, -0.0083333, -82.5833333, 0.1, 0.0083333 ]
            iNodeCount: 101
            jNodeCount: 121
            data:
[ 0.136, 0.304, 0.006, 0.003, ... 0.152, 0.388, 0.014, 0.007, 0.151, 0.389, 0.014, 0.007,
   :      :      :      :           :      :      :      :      :      :      :      :
  0.146, 0.324, 0.028, 0.069, ... 0.164, 0.416, 0.019, 0.034, 0.164, 0.417, 0.017, 0.033,
  0.146, 0.326, 0.027, 0.074, ... 0.165, 0.418, 0.025, 0.046, 0.164, 0.417, 0.018, 0.036  ]

          - gridName: "ONwinsor"
            affineCoeffs: [ 42.4166667, 0.0, -0.0083333, -83.1666667, 0.1, 0.0083333 ]
            iNodeCount: 171
            jNodeCount: 61
            data:
[ 0.154, 0.269, 0.002, 0.001, ... 0.165, 0.418, 0.025, 0.046, 0.164, 0.417, 0.018, 0.036,
  0.154, 0.269, 0.002, 0.001, ... 0.166, 0.419, 0.027, 0.058, 0.164, 0.417, 0.018, 0.037,
   :      :      :      :           :      :      :      :      :      :      :      :
  0.158, 0.250, 0.033, 0.028, ... 0.157, 0.413, 0.020, 0.051, 0.157, 0.414, 0.020, 0.052 ]
```

Commentary

Firstly, note that the GGXF format requires longitudes to be positive eastwards but in the NTv2 format longitudes are positive westwards. Therefore the signs of the longitude offsets in a GGXF file are reversed from those found in an NTv2 file. The second parameter value in the Windsor GGXF grid above, the longitude offset at i=0,j=0 of 0.269 arc-seconds (highlighted in red) would be given in the NTv2_0.gsb file as -0.269 arc-seconds.

From the interpolation CRS WKT definition the CRS type is geographic and its units can be seen to be degrees. Then, as the rotation values in this affine transformation are zero, from the affine coefficients and axis node counts, should they be required the envelope lower and upper corner coordinates in the Interpolation CRS can be determined. For the Canada west grid these are at (47°N 142°W) and (60°N 88°W), for the Windsor grid the grid envelope lower corner is at ((42.4166667 + (61-1) * -0.0083333), (-83.1666667 + (171-1) * 0.0)) = (41°55'N, 83°10'W) and the grid envelope upper corner to be at ((42.4166667 + (61-1) * 0.0), (-83.1666667 + (171-1) * 0.0083333)) = (42°25'N, 81°45'W). (Note: in this text and the example the recurring decimal degree values have been rounded to seven decimal places: in the actual data these would be carried to greater precision).

However, to identify the grid(s) in which a point of interest falls it is not necessary to convert the grid definitions to graticule coordinates. Instead, the graticule coordinates are converted to grid indices using the affine transformation coefficients. Doing this for the four corners of our study area in turn determines that:

- the grid index in the CAwest grid for the study area northwest corner is i=722.9, j=210.9. Both i and j values exceed the (nodeCount-1) values of the grid (648, 156) so the study area is outside of the west grid.
- all four corners of the study area fall within the CAeast grid.
- the study area southwest corner also falls in the Windsor grid at index 169.4, 0.8.
- the study area northwest corner also falls in the Sarnia grid at index 99.4, 119.2.
- the study area northeast corner also falls in the Toronto grid at index 0.6, 509.2.
- the study area southeast corner does not fall in any child grid. Offsets for the southeastern part of the study area have to be interpolated from the CAeast grid, in which the study area southeast corner is at index 75.1, 211.8.

For the study area southwest corner the four grid node indices for interpolation are:

i,j = 169,0 170,0
i,j = 169,1 170,1

The values for latitude and longitude offset at these grid nodes are shown in the grid extract above in blue and green respectively. Bi-linear interpolation for $\delta\varphi$ gives 0.1651" and for $\delta\lambda$ gives 0.4178". Then, referring to the GGXF Conventions (Annex B.3) for the definition of content type given in the file header (*geographic2dOffsets*), offsets are applied as additions. Using the formulas ($\varphi_T = \varphi_S + \delta\varphi$) and ($\lambda_T = \lambda_S + \delta\lambda$), the point at NAD27 latitude φ_S and longitude λ_S of 42°24'36"N, 81°45'18"W (west being negative) transforms to NAD83(Original) latitude φ_T and longitude λ_T of 42°24'36.165"N, 81°45'17.582"W. The accuracies can be interpolated in a similar fashion, with $\delta\varphi$ = 0.165" = 5.09m ± 0. 23m and $\delta\lambda$ = 0.4178" = 9.55m ± 0. 05m.

Note that the parameter values at common nodes on the child grid boundaries have identical values. For example Sarnia (99,120) and (100,120) are at the same location as Windsor (169,0) and (170,0) respectively and both have parameter values of $\delta\varphi$ = 0.165", $\delta\lambda$ = 0.417", N_{acc} = 0.025m, E_{acc} = 0.046m and $\delta\varphi$ = 0.164", $\delta\lambda$ = 0.417", N_{acc} = 0.025m, E_{acc} = 0.036m respectively. This facilitates continuity of interpolation across the boundary.

E.3.2 netCDF representation

The full NRCanada NTv2_0 file converted to the GGXF YAML file format and using external file referencing through the *dataSource* attribute is available on Github here.

The full NRCanada NTv2_0 file migrated to the GGXF netCDF file format is available on GitHub here).

A CDL file with only the netCDF header records is available on GitHub here.

E.4 Velocity grid

The hypothetical example here illustrates secular motion described through velocities. This hypothetical example uses data taken from the first ggxfGroup of example E.5 modelling secular deformation. (For consistency with other components the actual New Zealand deformation model published by LINZ from which extracts are used in E.5 describes secular motion as displacements to which a time function is applied).

E.4.1 YAML representation


```yaml
ggxfVersion: "GGXF-1.0"
content: velocityGrid
title: "NZ hypothetical velocity grid"
abstract: "Hypotherical example to illustrate secular motion described through velocities."
filename: "NZ hypothetical velocity grid.yaml"

version: "2011"
comment: |
    The parent grid describes secular deformation derived from NUVEL-1A rotation rates.
    The child grid describes secular deformation derived from the GNS model 2011 v4.

contentApplicabilityExtent:
  extentDescription: "New Zealand onshore and EEZ."
  boundingBox:
    southBoundLatitude: -55.95
    westBoundLongitude: 160.6
    northBoundLatitude: -25.88
    eastBoundLongitude: -171.2

interpolationCrsWkt: |
    GEOGCRS["NZGD2000",
      DATUM["New Zealand Geodetic Datum 2000",
        ELLIPSOID["GRS 1980",6378137,298.2572221,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,2],
      AXIS["Geodetic latitude (Lat)",north,ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Geodetic longitude (Lon)",east,ANGLEUNIT["degree",0.0174532925199433]],
    ID["EPSG",4167,URI["http://www.opengis.net/def/crs/epsg/0/4167"]]]

sourceCrsWkt: |
    GEOGCRS["NZGD2000",
      DATUM["New Zealand Geodetic Datum 2000",
        ELLIPSOID["GRS 1980",6378137,298.2572221,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,3],
      AXIS["Geodetic latitude (Lat)",north,ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Geodetic longitude (Lon)",east,ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Ellipsoidal height (h)",up,LENGTHUNIT["metre",1]],
    ID["EPSG",4959,URI["http://www.opengis.net/def/crs/epsg/0/4959"]]]

targetCrsWkt: |
    GEOGCRS["ITRF96",
      DYNAMIC[FRAMEEPOCH[1997.0]],
      TRF["International Terrestrial Reference Frame 1996",
        ELLIPSOID["GRS 1980",6378137,298.2572221,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,3],
      AXIS["Geodetic latitude (Lat)",north,ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Geodetic longitude (Lon)",east,ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Ellipsoidal height (h)",up,LENGTHUNIT["metre",1]]]
      ID["EPSG",7907,URI["http://www.opengis.net/def/crs/epsg/0/7907"]]]

operationAccuracy: 0.01
parameters:
      - parameterName: velocityEast
        parameterSet: "velocity"
        sourceCrsAxis: 1
```

```
        unitName: m/yr
        unitSiRatio: 3.16887651727315E-08
      - parameterName: velocityNorth
        parameterSet: "velocity"
        sourceCrsAxis: 0
        unitName: m/yr
        unitSiRatio: 3.16887651727315E-08

ggxfGroups:
  - ggxfGroupName: "national_velocity_model"
    interpolationMethod: bilinear
    grids:
      - gridName: "grid_nuvel1a_eez"
        affineCoeffs: [ -25.0, 0.0, -0.5, 158.0, 0.5, 0.0 ]
        iNodeCount: 73
        jNodeCount: 67
        data: [ 0.023004, 0.051255, ... -0.059495, 0.033768,
                  :          :              :          :
                -0.017214, 0.026107, ... -0.033665, 0.033706 ]
        childGrids:
            - gridName: "grid_igns2011_nz"
              affineCoeffs: [ -33.0, 0.0, -0.1, 165.5, 0.1, 0.0 ]
              iNodeCount: 141
              jNodeCount: 151
              data: [ 0.01239, 0.04615, ...  0.00702, 0.03462,
                          :         :             :         :
                       -0.02301, 0.03381, ... -0.03692, 0.03216 ]
```


Commentary

From the affine coefficients and axis node counts, for the grid derived from Nuvel 1 the envelope lower corner can be determined to be at (58°S 158°E) and the envelope upper corner to be at (25°S 194°E) = (25°S 166°W). For the IGNS model the envelope lower corner can be determined to be at (48°S 165°30'E) and the envelope upper corner to be at (33°S 179°30'E).

The example above is an extract from the New Zealand deformation model, in which it is included as one of the components. The full New Zealand deformation model is exemplified in E.5.

E.4.2 netCDF representation

A GGXF netCDF file for the velocity grids for Alaska (see E.7) is available on GitHub here.

The YAML file using ggxf-csv external grids for the Alaska velocity grids from which the GGXF netCDF file was produced is available on GitHub here.

A CDL file with only the netCDF header records for the Alaska velocity grids is available on GitHub here.

E.5 Deformation model

This example demonstrates the use of multiple ggxfGroups used to describe a deformation model. It uses selected extracts from the New Zealand NZGD2000 deformation model version 20180701. The full model consists of a national secular deformation model and 12 patches describing deformation from major earthquakes. It is represented in GGXF by 32 ggxfGroups with each ggxfGroup describing one of the model's components. Only parts of 2 of the 32 components in the model, illustrating secular, co-seismic and post-seismic elements, are included in this example.

E.5.1 YAML representation

The example is of a preparation text file including references to the external grids. The keywords for that external grid referencing are not part of this specification but specific to the file producer and are shown in grey italic text.

```
ggxfVersion: "GGXF-1.0"
content: deformationModel
title: "New Zealand Deformation Model"
abstract: "Defines the secular model (National Deformation Model) and patches for
significant deformation events since 2000."
filename: "nzgd2000-20180701-subset.yaml"

version: "20180701"
partyName: "Land Information New Zealand"
deliveryPoint: |
  Level 7, Radio New Zealand House
  155 The Terrace
  PO Box 5501
city: "Wellington"
postalCode: "6145"
electronicMailAddress: "customersupport@linz.govt.nz"
onlineResourceLinkage: "http://www.linz.govt.nz/nzgd2000"
publicationDate: "2018-07-01"
license: "Creative Commons Attribution 4.0 International"
contentApplicabilityExtent:
  extentDescription: "New Zealand onshore and EEZ."
  boundingBox:
    southBoundLatitude: -55.94
    westBoundLongitude: 160.62
    northBoundLatitude: -25.89
    eastBoundLongitude: -171.23
  boundingPolygon: Polygon ((-32.42 168.65 -34.98 168.10 -37.58 170.07 -40.60 167.30 -
   44.32 162.17 -51.17 160.62 -54.97 165.11 -55.94 168.78 -54.70 173.54 -53.26 174.64 -
   51.66 174.48 -53.04 178.46 -51.94 182.69 -50.45 183.84 -47.76 184.00 -46.81 187.31 -
   44.68 188.77 -42.93 188.53 -41.50 187.23 -40.26 182.07 -36.91 182.64 -34.59 180.22 -
   34.45 182.66 -33.12 184.50 -29.73 185.92 -27.47 185.34 -25.89 182.29 -27.22 179.01 -
   31.55 177.28 -34.32 179.37 -30.85 172.97 -30.88 171.22 -32.42 168.65))

interpolationCrsWkt: |
    GEOGCRS["NZGD2000",
      DATUM["New Zealand Geodetic Datum 2000",
        ELLIPSOID["GRS 1980",6378137,298.2572221,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,2],
      AXIS["Geodetic latitude (Lat)",north,ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Geodetic longitude (Lon)",east,ANGLEUNIT["degree",0.0174532925199433]],
    ID["EPSG",4167,URI["http://www.opengis.net/def/crs/epsg/0/4167"]]]

sourceCrsWkt: |
    GEOGCRS["NZGD2000",
      DATUM["New Zealand Geodetic Datum 2000",
        ELLIPSOID["GRS 1980",6378137,298.2572221,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,3],
      AXIS["Geodetic latitude (Lat)",north,ANGLEUNIT["degree",0.0174532925199433]],
```

```
      AXIS["Geodetic longitude (Lon)",east,ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Ellipsoidal height (h)",up,LENGTHUNIT["metre",1]],
    ID["EPSG",4959,URI["http://www.opengis.net/def/crs/epsg/0/4959"]]]

targetCrsWkt: |
    GEOGCRS["ITRF96",
      DYNAMIC[FRAMEEPOCH[1997.0]],
      TRF["International Terrestrial Reference Frame 1996",
        ELLIPSOID["GRS 1980",6378137,298.2572221,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,3],
      AXIS["Geodetic latitude (Lat)",north,ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Geodetic longitude (Lon)",east,ANGLEUNIT["degree",0.0174532925199433]],
      AXIS["Ellipsoidal height (h)",up,LENGTHUNIT["metre",1]]]
      ID["EPSG",7907,URI["http://www.opengis.net/def/crs/epsg/0/7907"]]]

operationAccuracy: 0.01

parameters:
      - parameterName: displacementEast
        parameterSet: "displacement"
        sourceCrsAxis: 1
        unitName: metre
        unitSiRatio: 1.0
      - parameterName: displacementNorth
        parameterSet: "displacement"
        sourceCrsAxis: 0
        unitName: metre
        unitSiRatio: 1.0
      - parameterName: displacementUp
        parameterSet: "displacement"
        sourceCrsAxis: 2
        unitName: metre
        unitSiRatio: 1.0
      - parameterName: displacementHorizontalUncertainty
        parameterSet: "displacementUncertainty"
        unitName: metre
        unitSiRatio: 1.0
        uncertaintyMeasure: 2CEP
      - parameterName: displacementUpUncertainty
        parameterSet: "displacementUncertainty"
        unitName: metre
        unitSiRatio: 1.0
        uncertaintyMeasure: 2SE

checkPoints:
  - sourceCrsCoordinates: [ -50.757000000, 165.271000000, 49.2000 ]
    sourceCoordinateEpoch: 2008.3
    targetCrsCoordinates: [ -50.756997865, 165.270996670, 49.2000 ]
    targetCoordinateEpoch: 2008.3

  - sourceCrsCoordinates: [ -50.757000000, 165.271000000, 49.2000 ]
    sourceCoordinateEpoch: 2018.3
    targetCrsCoordinates: [ -50.756995292, 165.270992658, 49.2000 ]
    targetCoordinateEpoch: 2018.3

ggxfGroups:
  - ggxfGroupName: "nz_linz_nzgd2000-ndm-grid02"
    comment: "Secular deformation model"
    interpolationMethod: bilinear
    gridParameters:
      - displacementEast
      - displacementNorth
    constantParameters:
      - parameterName: displacementHorizontalUncertainty
        parameterValue: 0.001
    timeFunctions:
      - functionType: velocity
        functionReferenceDate: "2000-01-01T00:00:00Z"
    grids:
```

```
    - gridName: "ndm_grid_nuvel1a_eez"
      comment: "Secular deformation model derived from NUVEL-1A rotation rates"
      affineCoeffs: [-25.0,0.0,-0.5,158.0,0.5,0.0]
      iNodeCount: 73
      jNodeCount: 67
      dataSource:
         dataSourceType: GDAL
         gdalSource: "GTIFF_DIR:1:nz_linz_nzgd2000-ndm-grid02.tif"
      childGrids:
        - gridName: "ndm_grid_igns2011_nz"
          comment: "Secular deformation model derived from GNS model 2011 V4"
          affineCoeffs: [-33.0,0.0,-0.1,165.5,0.1,0.0]
          iNodeCount: 141
          jNodeCount: 151
          dataSource:
             dataSourceType: GDAL
             gdalSource: "GTIFF_DIR:2:nz_linz_nzgd2000-ndm-grid02.tif"

  - ggxfGroupName: "nz_linz_nzgd2000-ds20090715-grid011"
    comment: "Dusky Sound (Fiordland) earthquake July 2009."
    interpolationMethod: bilinear
    gridParameters:
      - displacementEast
      - displacementNorth
      - displacementUp
    timeFunctions:
      - functionType: ramp
        startEpoch: 2009.536
        endEpoch: 2009.536
        functionReferenceEpoch: 2011.666
        scaleFactor: 1.05
      - functionType: ramp
        startEpoch: 2009.536
        endEpoch: 2011.666
        functionReferenceEpoch: 2011.666
        scaleFactor: 0.29
    grids:
      - gridName: "patch_ds_20090715_grid_ds_P0_L1"
        affineCoeffs: [-50.125,0.0,-0.125,165.4,0.15,0.0]
        iNodeCount: 11
        jNodeCount: 11
        dataSource:
           dataSourceType: GDAL
           gdalSource: "GTIFF_DIR:1:nz_linz_nzgd2000-ds20090715-grid011.tif"
```


E.5.2 netCDF representation

The YAML above uses a small subset of the New Zealand deformation model. The full deformation model file converted to the GGXF YAML file format and using external file referencing through the *dataSource* attribute is available on Github here.

The full New Zealand deformation model file migrated to the GGXF netCDF file format is available on GitHub here).

A CDL file with only the netCDF header records is available on GitHub here.

E.6 Gridded geodetic data not used in coordinate transformation software

This example illustrates gridded geodetic data that is not used directly in coordinate transformation software. The example uses deviation of the vertical (sometimes referred to as deflections of the normal) data for Puerto Rico held in a single grid.

Because the data is not used for coordinate transformations or point motion operations, there is no source CRS or target CRS and sourceCrsAxis is not given. Description of the interpolation CRS to which the data is referenced is still required.

E.6.1 YAML representation


```yaml
ggxfVersion: "GGXF-1.0"
content:     deviationsOfTheVertical
title:       "PRVI DOV 2018"
abstract:    "PRVI hybrid deflection model. Deflections are at the Earth's surface"
filename:    d2018prvi.yaml

contentApplicabilityExtent:
    extentDescription: "US Puerto Rico and Virgin Islands - onshore."
    boundingBox:
        southBoundLatitude:  17.67
        westBoundLongitude:  -65.09
        northBoundLatitude:  18.42
        eastBoundLongitude:  -64.6

partyName:             "National Geodetic Survey, National Oceanic and Atmospheric Administration"
deliveryPoint:         "1315 East West Hwy"
city:                  "Silver Spring"
postalCode:            "20910"
country:               "United States of America"
onlineResourceLinkage: "https://geodesy.noaa.gov/PC_PROD/GEOID18/Format_ascii/g2018p0.asc.zip"

interpolationCrsWkt: |
    GEOGCRS["NAD83 (2011)",
      DATUM["North American Datum 1983 (2011) epoch 2010.00",
          ELLIPSOID["GRS 1980",6378137.0,298.2572221,LENGTHUNIT["metre",1]]],
      CS[ellipsoidal,2],
      AXIS["Geodetic latitude (Lat)",north],
      AXIS["Geodetic longitude (Lon)",east],
      ANGLEUNIT["degree",0.0174532925199433]]

parameters:
      parameterName: deviationEast
      unitName: arc-second
      unitSiRatio: 4.84813681109536E-06
      parameterName: deviationNorth
      unitName: arc-second
      unitSiRatio: 4.84813681109536E-06

ggxfGroups:
    - ggxfGroupName: "Puerto Rico Virgin Islands DEFLEC18"
      interpolationMethod: biquadratic
      grids:
          - gridName: "Puerto Rico Virgin Islands DEFLEC18"
            affineCoeffs: [15.0, 0.0, 0.01666666667, -69.0, 0.01666666667, 0.0]
            iNodeCount:  301
            jNodeCount: 361
            data: []
```

E.6.2 netCDF representation

The YAML above omits the grid data. The full YAML file including the grid data is available on Github here.

The full file for PRVI deviations of the vertical in the GGXF netCDF file format is available on GitHub here.

A CDL file with only the netCDF header records is available on GitHub here.

E.7 Grid priority

The gridPriority attribute is used to identify which of intersecting sibling grids should take priority for use in interpolation of nested grids (refer to 5.6). The example below uses velocity grids from the US NGS HDTP application to illustrate the application of grid priority in GGXF. HDTP includes ten velocity grids covering the coterminus US (CONUS) and Alaska. Their spatial extents are shown in Figure E.4. HDTP prioritises their use through ranking with 1 being the highest rank and 10 the lowest rank.

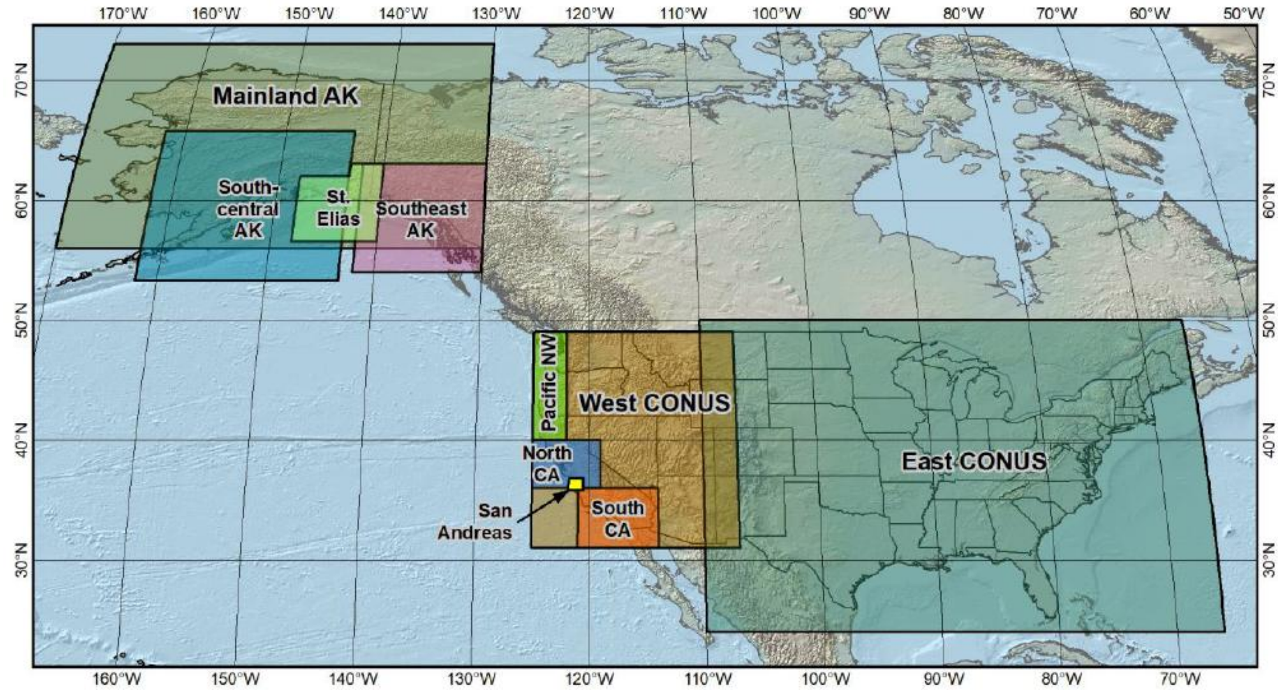


Figure E.4 — Spatial view of HDTP grids

In GGXF these grids may be represented in separate ggxfGroups for CONUS and Alaska with each grid being a root grid in their respective region. Being root grids they are siblings and because they all intersect with at least one sibling in their ggxfGroup, all require a gridPriority (in GGXF higher ranked grids have higher priority). The mapping from HDTP to GGXF structure is shown in Figure E.5.

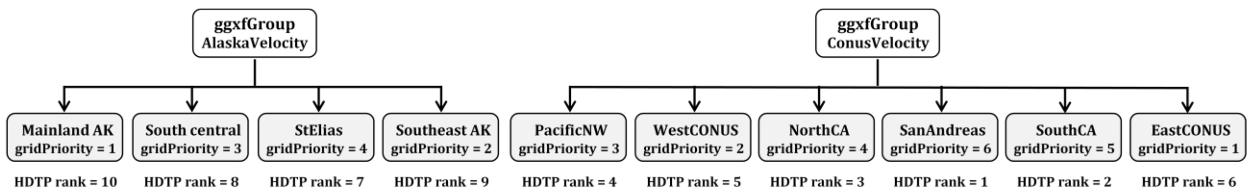


Figure E.5 — Structural view of GGXF grids

The HDTP grids for Alaska converted to the GGXF YAML file format and using external file referencing through the *dataSource* attribute is available on Github here.

The full set of HDTP grids for Alaska in the GGXF netCDF file format is available on GitHub here.

Bibliography

[1] OGC Abstract Specification Topic xx, *Deformation model Functional model*, https://github.com/opengeospatial/CRS-Deformation-Models/tree/master/products/specification (accessed 2022-05-10).

[2] OGC Abstract Specification Topic 2, *Referencing by coordinates*, http://docs.opengeospatial.org/as/18-005r5/18-005r5.html (accessed 2020-05-26).

[3] EPSG Geodetic Parameter Dataset, https://epsg.org (accessed 2020-05-26).

[4] ISO Geodetic Registry, https://geodetic.isotc211.org (accessed 2020-05-26).

[5] IERS Conventions (2010), https://www.iers.org/IERS/EN/Publications/TechnicalNotes/tn36.html/ (accessed 2021-05-26).

[6] Leenhouts, P. P. (1985). *On the computation of bi-normal radial error*. NAVIGATION, Journal of the Institute of Navigation, 32(1), 16-28.

[7] Langley, R. B. (1991). *The mathematics of GPS*. GPS World, 2(7), 45-50.

[8] OGC Implementation Specification 06-103r4, *Simple Features Access — Part 1: Common Architecture*, https://portal.ogc.org/files/?artifact_id=25355 (accessed 2021-05-26)

[9] Unicode Standard Annex #31, https://unicode.org/reports/tr31/#Default_Identifier_Syntax, (accessed 2021-05-26)

[10] YAML specification, https://yaml.org/spec/1.2/ (accessed 2021-05-26)

[11] netCDF User's Guide http://www.unidata.ucar.edu/software/netcdf/docs/user_guide.html (accessed 2022-05-26)

References for Table B.9 (Intepolation method identifier)

[12] Arata, Louis K. "Tricubic interpolation." *Graphics Gems V* (1995): 107-110.

[13] Haynes, Andrew L., and Clare E. Parnell. "A trilinear method for finding null points in a three-dimensional vector space." *Physics of Plasmas* 14, no. 8 (2007): 082107.

[14] Lekien, Francois, and J. Marsden. "Tricubic interpolation in three dimensions." *International Journal for Numerical Methods in Engineering* 63, no. 3 (2005): 455-471.

[15] Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[16] Russell, William S. "Polynomial interpolation schemes for internal derivative distributions on structured grids." *Applied Numerical Mathematics* 17, no. 2 (1995): 129-171.

[17] Shi, Wen Zhong, Q. Q. Li, and C. Q. Zhu. "Estimating the propagation error of DEM from higher-order interpolation algorithms." *International Journal of Remote Sensing* 26, no. 14 (2005): 3069-3084.

[18] Smith, D. *Biquadratic Interpolation*. NOAA Technical Memorandum NOS NGS 84, 2022. https://geodesy.noaa.gov/library/pdfs/NOAA_TM_NOS_NGS_0084.pdf (accessed 2022-11-26).

[19] Wagner, Rick. "Multi-linear interpolation." *Beach Cities Robotics* (2008).

[20] Young, David M., and Robert Todd Gregory. *A survey of numerical mathematics*. Vol. 1. Dover, 1988.