NetCDF Irregular Grids

Applied Science Associates
55 Village Square Drive
South Kingstown, RI 02879
401-789-6224

Document History

Reviewed by

Organization	Party	Date/Phase

Revision Record

Version	Date	Notes
0.0	11MAY2009	Draft document
0.1	12JUN2009	Draft revision
0.2	29JUN2009	Draft revision

Change Requests

Date	Party	Notes

Document Conventions

Internal notes, caveats are expressed in italics within square brackets.
Code samples and CDL output are expressed in small courier font.
Applied Science Associates, Inc. is referred-to in this document as 'ASA'.
Naval Oceanographic Office is referred-to in this document as 'NAVOCEANO'.

Contents

Abstract

Applied Science Associates, Inc (ASA), in conjunction with the Naval Oceanographic Office (NAVOCEANO), seeks to make environmental simulations generated from disparate models available to a broad range of users and client applications through the use of a standard format. This document summarizes the NetCDF format, significant conventions and ASA's implementation of those conventions in order to create a NetCDF document format standard which will accommodate the majority of model outputs. The focus is to add coherence and efficiency in creating documents based upon a standard and allowing ready consumption of standards-based data.

Introduction

This document describes the Network Common Data Form (NetCDF) format and the conventions to which the Irregular Grid format strives to adhere. It is intended to be used as a guide for providers of NetCDF data. NetCDF was developed at the Unidata Program of the University Corporation for Atmospheric Research (UCAR) (http://my.unidata.ucar.edu). NetCDF has been adopted for use in earth, ocean and atmospheric sciences in large part because it is a flexible, self-describing format capable of conveying large sets of array-oriented data. There are NetCDF libraries for software developers and modelers that provide a straightforward interface to that data.

The flexibility provided by NetCDF has allowed data providers and users to create NetCDF schemas to suit their own particular needs. While there are some NetCDF conventions to follow, such as the COARDS convention, sponsored by the Cooperative Ocean-Atmosphere Research Data Service, and the CF (Climate and Forecast Metadata) convention, these conventions still allow a great deal of flexibility for entities providing data. Data consumers may now depend on data that follows a particular convention; however they must continue to make allowances for a vast array of different schemas. ASA, having been in the position where new code must be created to accommodate the processing of data from new data sources, developed a least-common-denominator schema for the conveyance of ocean and atmospheric data.

It is our intention to define a standard NetCDF schema that adheres to CF conventions and will act as the base structure for communication of a standardized set of metocean model output variables. The overriding goal is to foster communication between data provider and data consumer such that data provided in this standard format will be guaranteed readable by applications which also adhere to the format. Additionally, by adhering to the CF convention,

this standard format will also be readable, for example, by any application that can read CF-or-COARDS-conformant NetCDF documents. Once base data has been written in compliance with the Irregular Grid standard, additional data may be written to the NetCDF document. However, it should be understood that any additional data written that does not follow Irregular Grid standard guidelines may not be guaranteed readable by clients that consume Irregular Grid standard documents.

Target Audience

This document has been written for technical and non-technical personnel as a guide for creating NetCDF documents that will allow both data producers and consumers to more-effectively communicate with each other by leveraging greater fore-knowledge of a given data set's schema. It will also serve to explain standards choices, and it will convey ASA's and NAVOCEANO's commitment to work within currently accepted guidelines while attempting to define workable standards that may be readily recognized within our community.

Requirements

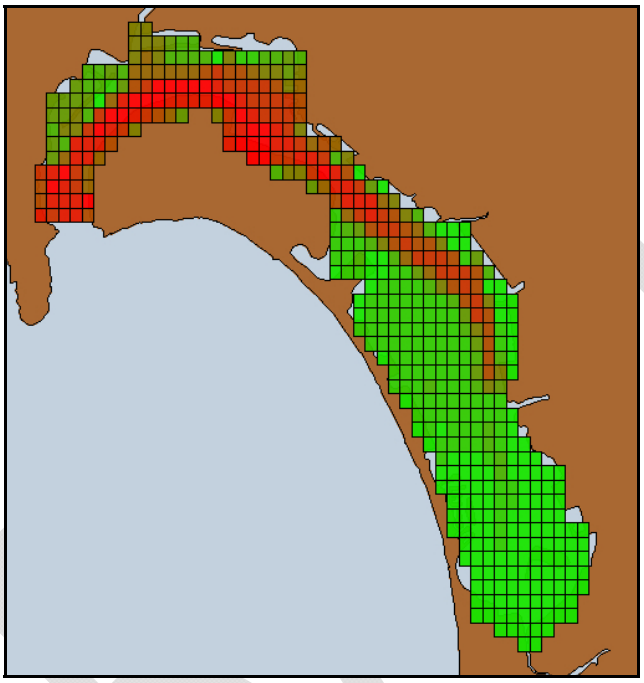
Data producers must be able to present model output in a standardized format that conforms to overall CF conventions and leverages CF standard naming conventions. A standard must be created such that client applications will need little to no additional programming in order to determine time and position parameters as presented within the NetCDF document. Data other than time and position will be presented in such a way within the document as to allow for discovery using standard names documented within the existing CF convention. Variables which are not accounted-for within the CF convention will not provide standard names. The standardized format will provide metadata (data about the data) through the use of accepted global attributes as defined in this document. Additional metadata will accompany the NetCDF document(s) in the form of an XML data specification document.

The purpose of the definition of a standard for irregular grids is not necessarily to provide a format that fits entirely within the bounds of the CF – or any other – convention. However, stressing CF compliance while insisting on adherence to this standard will advance ready acceptance of this standard beyond the services and applications created specifically to output and consume standard-formatted data.

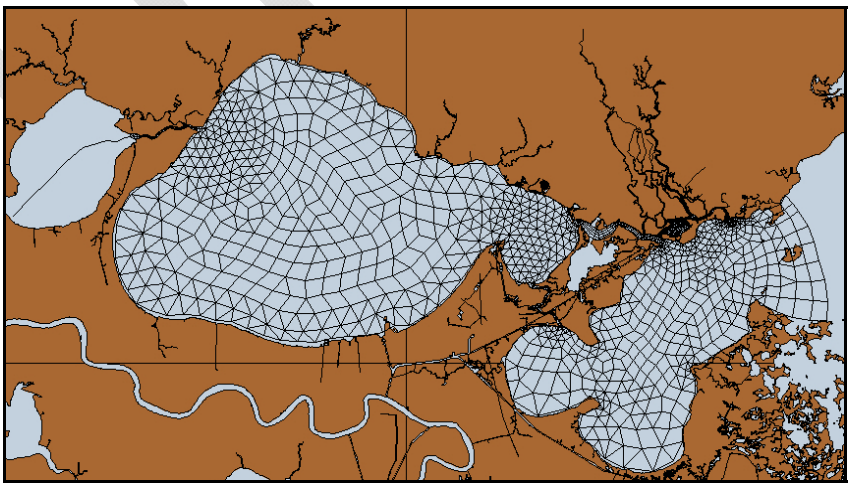
NetCDF Irregular Grid

NetCDF files are made-up of four parts (evident when an nc file is converted to a cdl file using a utility like ‘ncdump’, see sample outputs below): they are dimensions, variables, global attributes and data. ASA is adopting the CF convention in defining the following NetCDF Irregular Grid schema. Refer to Appendix A for abstracts of both the CF and COARDS conventions, and refer to the references section for links to further convention documentation.

The Irregular Grid standard allows the representation of multiple points that may or may not be aligned in a regular grid arrangement. In this context, a regular grid refers to a spatial representation of data defined by a rectangular matrix. While and irregular grid is fundamentally a collection of non-uniformly spaced points connected to form regions. Examples of each grid type are presented below.



Example of a Regular Grid: note cells representing land are masked but still maintained in the data structure.



Example of an Irregular Grid

Referred to above as a lowest-common-denominator schema, use of cells allows for representation of any arrangement of points in space – whether conforming to a structured grid, rotated grid, irregular grid or a simple collection of points. The use of cells rather than a strict grid representation allows the data provider to avoid mapping output data strictly to a gridded structure, land-masking data is not necessary for water data, the number of nearest neighbors may be variable, and underlying boundaries may be conveyed directly. In many cases, this will allow for smaller data files, and it allows for complex gridding and geographic overlay of multiple data sets. Multiple vertical levels may also be conveyed, and the nodes defining cell boundaries may be included.

Dimensions

Dimensions are the array indices, globally used by the NetCDF document as parameters for all of the data variables. The Irregular Grid standard dimensions will be: time, ncell, level and numvert.

- time – unlimited count of time steps stored within the NetCDF document
- ncell – specific count of cells that will hold data, such as position and scalar data
- level – specific count cell planes along the vertical axis that will hold data
- numvert – maximum number of vertices about each cell, allows specification of cell dimensions even when using Irregular Grids. Note: specifying numvert as the maximum number of vertices will allow for Irregular Grids which contain cells with varying number of vertices – up to the maximum

Variables

Coordinate variable will be time.

- time – a function of the time dimension, will be saved as a number of seconds since the epoch date 01JAN1970

Auxiliary coordinate variables will be lat and lon. These position variables will be functions of the ncell dimension. Each ncell will be assigned coordinates lat and lon.

- lat – a function of ncell which contains the latitude value for a given ncell
- lon – a function of ncell which contains the longitude value for a given ncell

The auxiliary coordinate variables shall also contain attributes referencing boundary variables. The boundary variables will be functions of the ncell dimension as well as the numvert dimension. The boundary variables will allow modelers to specify cell boundaries within the NetCDF document. Boundary variables will also allow for algorithms that determine whether or not adjacent cells share an edge. Boundary variables compliment position variables, lat/lon. It is required that the boundary variables be the same type as the auxiliary coordinate variables which reference them and that there be one value per vertex for each ncell. The CF convention states

that boundary variables do not require any attributes. However the Irregular Grid standard specifies numvert as a maximum number of vertices, allowing data sets to contain cells with varying numbers of vertices up to the maximum represented by numvert. Therefore, the boundary variables (lat_bnds and lon_bnds) will include the _FillValue attribute, allowing consumers to derive the number of vertices per cell by inspection.

- lat_bnds – a function of ncell and numvert which contains the latitude values of all vertices for the given ncell
- lon_bnds – a function of ncell and numvert which contains the longitude values of all vertices for the given ncell

Additional variables will be functions of time, ncell and/or level, or any combination thereof. For example:

- depthBottom, if included, shall be stored as simply a function of ncell. The link from a depth measure to its geographic position may be made through a lookup of lat and lon for the given ncell.
- seaSurfaceHeight, if included, shall be stored as a function of time and ncell. A two-dimensional array may be stored within the NetCDF document representing surface elevation for each ncell for each time step.
- U component of current may be stored as a function of time, level and ncell. The resulting three-dimensional array will allow representation of the U component of water velocity, for example, at multiple positions, multiple levels for each time step.

Variables shall be designated with long names and short names that will make sense at a glance. Provide CF-standard names when available (as found in CF Standard Name Table, see references section). Reference the table below for accepted variable names, long names and standard names of common model output.

Variable Name	Long Name	CF - Standard Name
temperatureWater	Water temperature	sea_water_temperature
salinityWater	Salinity	sea_water_salinity
currentUComponent	U Component of current	eastward_sea_water_velocity
currentVComponent	V Component of current	northward_sea_water_velocity
currentWComponent	W component of current	upward_sea_water_velocity
currentSpeed	Current speed	sea_water_speed
currentDirection	Current direction	direction_of_sea_water_velocity
seaSurfaceElevation	Sea surface elevation	sea_surface_elevation
soundSpeedWater	Water sound speed	speed_of_sound_in_sea_water
conductivityWater	Water conductivity	sea_water_electrical_conductivity
densityWater	Water density	sea_water_density
depthBottom	Cell based bottom depth	depth
seaSurfaceHeight	Water elevation	sea_surface_height

windUComponent	U Component of wind	eastward_wind
windVComponent	V Component of wind	northward_wind

Additionally, the Irregular Grid standard at this time will recognize the following variables. The standard names specified in this table are not recognized as CF Standard Names. It is recommended that the standard names below not be included within Irregular Grid documents unless or until they are accepted as the CF Standard Names officially.

Variable Name	**Long Name**	**Non-CF Standard Name**
horizontalEddyViscosity	Horizontal eddy viscosity at each level	horizontal_eddy_viscosity_at_each_level
filteredCurrentUComponent	Filtered U Component of current	filtered_eastward_sea_water_velocity
filteredCurrentVComponent	Filtered V Component of current	filtered_northward_sea_water_velocity
bottomStressAtUPoint	Bottom Stress at U-point	bottom_stress_at_u_point
bottomStressAtVPoint	Bottom Stress at V-point	bottom_stress_at_v_point
enstrophy	Enstrophy at each level	enstrophy_at_each_level
maximumTau	Maximum tau at each level	maximum_tau_at_each_level
vorticity	Vorticity at each level	vorticity_at_each_level

Name usage within NetCDF document variable declaration, example:

```
double currentUComponent(time, ncell)  //variable name
       currentUComponent:long_name = "U Component of current" ;
       currentUComponent:standard_name = "eastward_sea_water_velocity" ;
```

Guidance for use for the various names assigned an output variable are below.

Variable Attributes

Use of variable attributes will also conform to CF conventions, using standard attribute names and values as provided by the convention.

- (implied) variable name – variables holding model data of the types represented within the above table will use the variable names specified within the table. Applications coded to the standard presented within this document will require consistent variable naming in order to be able to access model data without human intervention or further development.
- long_name – the long name will be a more fully-descriptive name for the variable. Recommended long names are given in the table above.
- standard_name - use of standard names that follow the CF standard name table will allow NetCDF document consumers outside of those adhering to the standards presented within this document to more easily determine what data the variable represents. Example: the U component of currents variable would contain the standard_name attribute

'eastward_sea_water_velocity', a standard name provided by the CF standard names table. A complete listing of all valid CF standard names may be found at *http://cf-pcmdi.llnl.gov/*. Variables for which there are no CF standard names should not include the standard_name attribute.

- units - the units value must be consistent with what is found in the CF standard name table for the variable. If the variable does not exist within the CF standard name table then its units value must be consistent with variables with similar characteristics which are found within the CF standard name table.
- _FillValue – the fill value of value that represents missing or undefined data. The _FillValue type must be the same as the variable's declared type
- scale_factor – if present, the data within the variable is to be multiplied by this factor by the reading application

It is recommended that applications utilize the variable name directly and derive consistent descriptions from the variable name.

Global Attributes

Global attributes allow the modeler or data packager to convey useful, high-level metadata to the consumer. Standard CF convention attributes include the following:

- comment – miscellaneous information about the data
- Conventions – name of the conventions followed by the data set
- history – list of data modifications
- institution – producer of original data
- source – method of production of the original data
- title – short description of data set

In addition to the above attributes, data producers will add the following attributes to further describe the data set's metadata, origin and distribution characteristics:

- netcdf_class_description – short description of the NetCDF format in use
- lonWest - Minimum longitude for model coverage (decimal degrees)
- lonEast - Maximum longitude for model coverage (decimal degrees)
- latSouth - Minimum latitude for model coverage (decimal degrees)
- latNorth - Maximum latitude for model coverage (decimal degrees)
- startDTG - Start date/time of model output (data set) (yyyy-mm-dd hh:mm:ss UTC)
- endDTG - End date/time of model output (data set) (yyyy-mm-dd hh:mm:ss UTC)
- tauInterval - Temporal interval between model outputs within a single model output data set in seconds, minutes, hours or days. May also be set to "variable" for inconsistent taus. Units default to units defined for the time variable.
- classificationLevel – IC ISM defined – acceptable one letter abbreviations: U|R|C|S|T

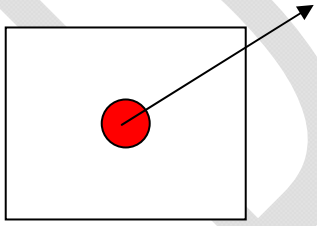
- classificationAuthority – IC ISM defined - used primarily at the resource level to specify the identity, by name or personal identifier, and position title of the original classification authority for the data
- distributionStatement – IC ISM defined - used at both the resource and the portion levels to identify the expansion or limitation on the DISTRIBUTION of the data
- downgradeDate – IC ISM defined – date classification was modified
- generatingModel – Model name referenced to a specific geographical implementation
- modelType – specific ocean model name
- gridTypeH - Type of horizontal grid of model output
- forecastType - Forecast type for model output (Hindcast|Forecast|Both)
- accreditation - Accreditation authority and/or process that provides a formal accreditation process that applies to the model (AMOP|OAML|AMOP-trans|OAML-trans|Other|None)
- operationalStatus - Semi-formal statement that applies to the implementation of the model type in a specific geographical region (Developmental|Pre-Operational|Operational)
- analysisDTG - The date/time of the Analysis which is typically time the forecast period starts (tau = 0) and the end of the hindcast period (if present) (yyyy-mm-dd hh:mm:ss UTC)

Graphical representation of Irregular Grid cells

The following images represent specific models which are covered by the Irregular Grid format.

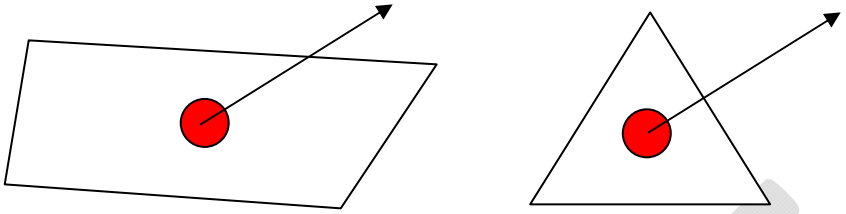
HydroMap 2D & 3D

- Rectangular cells
- Can be nested (non-uniform cell area)
- Scalar and vector data at cell centroid



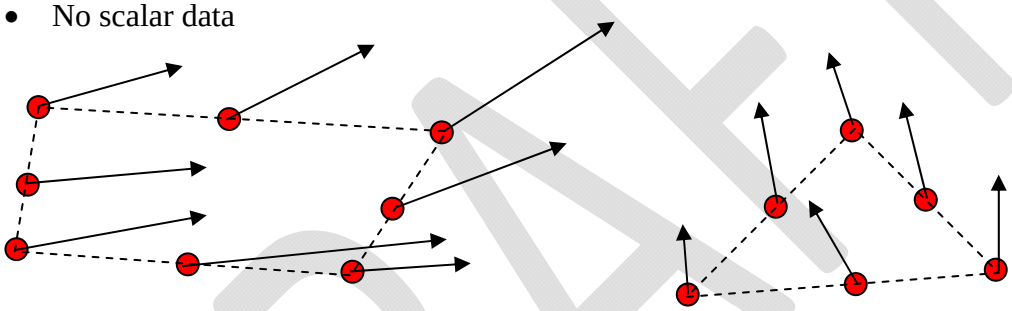
RMA2 – “All”

- Quadrilateral or triangular cells
- Scalar and vector data at cell centroid
- Vector is the average of node and mid-edge vectors (see RMA2 –“Vectors Only”)



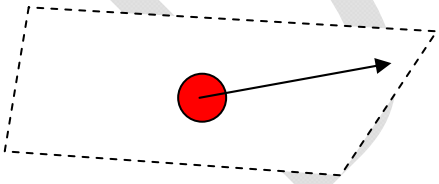
RMA2 – “Vectors Only”

- Quadrilateral or triangular cells
- Vector data at nodes and mid-edges
- No scalar data



Delft 2D & 3D

- Quadrilateral cells
- Can be boundary fitted
- Scalar and vector data at cell centroid



Note:

The Irregular Grid format is not limited to the model files shown above. Other models such as NCOM, HYCOM and ROMS could also be adapted. The nature of the Irregular Grid schema is such that most data sets, regardless of grid type, may be saved in this format.

Sample: Irregular Grid NetCDF header

```
netcdf delft2D_sample {
dimensions:
      time = UNLIMITED ; // (49 currently)
      ncell = 4497 ;
      level = 1 ;
      numVert = 4 ;
variables:
      double time(time) ;
            time:long_name = "Time" ;
            time:units = "seconds since 1970-01-01" ;
      double levelDepth(level, ncell) ;
            levelDepth:long_name = "level" ;
            levelDepth:positive = "down" ;
            levelDepth:units = "m" ;
      double lat(ncell) ;
            lat:long_name = "Latitude" ;
            lat:units = "degrees_north" ;
            lat:bounds = "lat_bnds" ;
      double lon(ncell) ;
            lon:long_name = "Longitude" ;
            lon:units = "degrees_east" ;
            lon:bounds = "lon_bnds" ;
      double lat_bnds(ncell, numVert) ;
            lat_bnds:_FillValue = -9999. ;
      double lon_bnds(ncell, numVert) ;
            lon_bnds:_FillValue = -9999. ;
      float currentUComponent(time, level, ncell) ;
            currentUComponent:long_name = "U component of current" ;
            currentUComponent:units = "m/s" ;
            currentUComponent:standard_name = "eastward_sea_water_velocity" ;
            currentUComponent:_FillValue = -9999.f ;
      float currentVComponent(time, level, ncell) ;
            currentVComponent:long_name = "V component of current" ;
            currentVComponent:units = "m/s" ;
            currentVComponent:standard_name = "northward_sea_water_velocity"
;
            currentVComponent:_FillValue = -9999.f ;
      float horizontalEddyViscosity(time, level, ncell) ;
            horizontalEddyViscosity:long_name = "Horizontal eddy viscosity at
each level" ;
            horizontalEddyViscosity:units = "m2/s" ;
            horizontalEddyViscosity:_FillValue = -9999.f ;
      float filteredCurrentUComponent(time, ncell) ;
            filteredCurrentUComponent:long_name = "Filtered U Component of
current" ;
            filteredCurrentUComponent:units = "m/s" ;
            filteredCurrentUComponent:_FillValue = -9999.f ;
      float bottomStressAtVPoint(time, ncell) ;
            bottomStressAtVPoint:long_name = "Bottom Stress at V-point" ;
            bottomStressAtVPoint:units = "N/m2" ;
            bottomStressAtVPoint:_FillValue = -9999.f ;
      float enstrophy(time, level, ncell) ;
```

```
            enstrophy:long_name = "Enstrophy at each level" ;
            enstrophy:units = "1/s2" ;
            enstrophy:_FillValue = -9999.f ;
      float maximumTau(time, ncell) ;
            maximumTau:long_name = "Maximum tau at each level" ;
            maximumTau:units = "N/m2" ;
            maximumTau:_FillValue = -9999.f ;
      float filteredCurrentVComponent(time, ncell) ;
            filteredCurrentVComponent:long_name = "Filtered V Component of
current" ;
            filteredCurrentVComponent:units = "m/s" ;
            filteredCurrentVComponent:_FillValue = -9999.f ;
      float seaSurfaceHeight(time, ncell) ;
            seaSurfaceHeight:long_name = "Sea surface elevation" ;
            seaSurfaceHeight:units = "m" ;
            seaSurfaceHeight:standard_name = "sea_surface_elevation" ;
            seaSurfaceHeight:_FillValue = -9999.f ;
      float vorticity(time, level, ncell) ;
            vorticity:long_name = "Vorticity at each level" ;
            vorticity:units = "1/s" ;
            vorticity:_FillValue = -9999.f ;
      float bottomStressAtUPoint(time, ncell) ;
            bottomStressAtUPoint:long_name = "Bottom Stress at U-point" ;
            bottomStressAtUPoint:units = "N/m2" ;
            bottomStressAtUPoint:_FillValue = -9999.f ;

// global attributes:
            :title = "Testing Delft Conversion" ;
            :reference = "Created: 2009-06-12 15:17:58 UTC" ;
            :netcdf_class_description = "Unstructured Grid" ;
            :Conventions = "CF-1.1" ;
            :lonWest = "129.369369506835" ;
            :lonEast = "129.543395996093" ;
            :latNorth = "36.1083259582519" ;
            :latSouth = "35.9918975830078" ;
            :startDTG = "2008-03-08 00:00:00 UTC" ;
            :endDTG = "2008-03-10 00:00:00 UTC" ;
            :tauInterval = "3600" ;
            :classificationLevel = "Unclassified" ;
            :classificationAuthority = "not applicable" ;
            :distributionStatement = "not applicable" ;
            :downgradeDate = "not applicable" ;
            :generatingModel = "not applicable" ;
            :modelType = "not applicable" ;
            :gridTypeH = "not applicable" ;
            :forecastType = "not applicable" ;
            :accreditation = "not applicable" ;
            :operationalStatus = "not applicable" ;
            :analysisDTG = "not applicable" ;
            :comments = "not applicable" ;
            :source = "not applicable" ;
            :institution = "not applicable" ;
            :history = "not applicable" ;
}
```

References

Brian Eaton, et al. 2008. *NetCDF Climate and Forecast(CF) Metadata Conventions Version 1.1*

Various, 2003-2008. NetCDF CF Convention online documentation - *http://cf-pcmdi.llnl.gov/*

Unidata, 2009. Unidata portal - *http://www.unidata.ucar.edu/*

Appendix A Conventions

I. COARDS

Cooperative Ocean-Atmospheric Research Data Service (COARDS) conventions have been registered with Unidata. Documentation and samples may be found at ftp://ftp.unidata.ucar.edu/pub/netcdf/Conventions/COARDS. Standard summary is as follows:

- NetCDF filename extension is ".nc"
- Global attributes are to include the "Conventions" attribute, declared as: :Conventions = "COARDS"
- Variables
    - Variable attributes: The following attributes are standardized and significant. Variables may be declared with any number of attributes. For the most part, any attributes not listed would be ignored by ASA applications.
        - long_name – descriptive name or title for a given variable
        - scale_factor – data multiplier
        - add_offset – data offset adjustment. scale_factor and add_offset may be used in conjunction in order to facilitate data compression.
        - _FillValue – if declared as the same data type as the variable, the value offered within this attribute will be the variable's default value.
        - missing_value – used to represent missing data in a significant manner unlike the _FillValue, which represents a data element that has not been written-to.
        - units – the variable's unit notation, conforming when possible to the Unidata udunits package, described at: http://my.unidata.ucar.edu/content/software/udunits
        - positive – the positive direction for a variable measured on a vertical axis, either "up" or "down."
        - scale_factor and add_offset attributes may be used to define how data is to be modified after reading from a variable containing packed data.
    - Variable names should begin with a letter. Variable names may be composed of letters, digits and underscores, and must be case-insensitive.
    - Variables should be defined based on a maximum of 4 space and time or index and time dimensions. Should additional dimensions be necessary, they should precede the space/time dimensions in variable declarations.
    - Order of dimensions within variable declarations should be as follows: time, height/depth/pressure, latitude, longitude.
    - It is recommended that variable data type "char" not be used.
- Coordinate Variables

- Coordinate variables are defined as 1-dimensional variables named exactly as their dimensions are named. Coordinate variable values should consistently increase or consistently decrease and must not have any missing data. Coordinate variable values need not be evenly spaced.
- Coordinate variable attributes
    - _FillValue and missing_value – not allowed for coordinate variables
    - Units – required if the coordinate variable represents time or space values. COORDS does not support the unit "degrees." Description of latitude or longitude coordinate variable is recommended as "degrees_north" or "degrees_east"
    - Time or Date variable: must include the units attribute. The units attribute will indicate increments from a reference date/time.

II. CF 1.1

The Climate and Forecast (CF) convention is described as "generalizing and extending" the COARDS convention. The CF convention is designed to be backward compatible with COARDS – Any document conforming to COARDS also conforms to CF. CF does relax some of the COARDS constraints in order to make CF NetCDF documents more efficient and/or more descriptive for CF models, however CF-compliant documents should still be readable by applications that can read COARDS documents.

CF conformance requirements are listed at *http://cf-pcmdi.llnl.gov/*. Summary CF requirements are as follows:

- NetCDF filename extension is ".nc"
- Global attributes
    - The "Conventions" attribute, declared as :Conventions = "CF-1.0"
- Dimension, variable and attribute names must begin with a letter and may only be composed of letters, numbers and the underscore. When case is ignored, no two variable names should be identical.
- Dimensions
    - It is recommended, but not required, that dimensions be declared in the following order: time, vertical dimension, latitude dimension, longitude dimension, followed by any additional declarations required by the model.
    - In variable declarations, any additional, non-time/space dimensions that are required by the variable should be added to the left of the time/space dimensions.
    - Any dimension declared as UNLIMITED must be the left-most dimension regardless of what other dimensions might be present.
    - Dimensions representing cell vertices in bounds variables should appear to the right of the time/space dimensions.

      [some contradictions here and below regarding dimension ordering in variable

declarations – basically, coordinate and auxiliary coordinate variables should be declared as described]

- Variables
  - Variable attributes: The following attributes are standardized and significant. As with COOARDS, variables may be declared with any number of attributes. Attributes beyond those described below will most-likely be ignored by ASA applications.
    - long_name – descriptive name or title. Recommended but not required
    - standard_name – descriptive name or title that follows guidelines in the CF standard name table (*http://cf-pcmdi.llnl.gov/*). Recommended but not required. As CF standard names do not currently completely fulfill the needs of current modeling, ASA suggests "standard-looking" names be used for standard_names when reasonable equivalents are not available in the standard name table. CF still asserts that the only legal values for standard_name are explicitly contained within the standard name table.

      All variables, except boundary variables, should have either long_name or standard_name attributes. When ASA applications render NetCDF data, they look first to long_name, then to standard_name, finally to the variable name for the variable description to tag rendered data with.
    - _FillValue – if declared, must be of the same type as its associated variable, and it must not be within the valid_range, if present, declared for the variable.
    - missing_value – must be of the same type as its associated variable. ASA applications use missing_value to exclude points with matching data from maps and charts. CF convention states that missing_value should no longer be used in favor of _FillValue, however ASA applications currently work with missing_value. CF recommends where missing_value is still required, that _FillValue be used as well, and that _FillValue contain the same value as missing_value.
    - units – the variable's unit notation
      - required for all variables that represent dimension (except for boundary variables), conforming to the udunits package (*http://cf-pcmdi.llnl.gov/*)
      - units representing level, layer and sigma level are exempt to conforming to udunits.
      - Units for variables that have standard_name specified must be consistent with the units represented for that standard_name in the standard name table.

- CF states that units level, layer and sigma_level are deprecated.

  CF conventions are relatively manic on the subject of units. There are detailed descriptions of what to use where in the CF documentation. There are particularly detailed instructions on the handling of the units attribute for dimensionless vertical variables (such as sigma in POM) in Appendix C of the CF convention. For ASA application purposes, units are required to provide context and descriptive quantification.
  - positive – the positive direction for a vertical variable. The only legal values are "up" and "down"
  - scale_factor and add_offset attributes may be used to define how data is to be modified after reading from a variable containing packed data.
- Variable names should begin with a letter. Variable names may be composed of letters, digits and underscores, and must be case-insensitive.
- Variables should be defined based on a maximum of 4 space and time or index and time dimensions. Should additional dimensions be necessary, they should precede the time/space dimensions in variable declarations, except in the case of boundary variables, where the number of vertices would follow the time/space dimensions.
- Order of dimensions within variable declarations should be as stated in the Dimensions section above.

- Coordinate Variables
  - Coordinate variable values should consistently increase or consistently decrease (they must be "monotonic"), and must not have any missing data.
  - Coordinate variable attributes
  - _FillValue and missing_value – not allowed for coordinate variables
  - units
  - required if the coordinate variable represents time or space values.
  - Time variable: must include the units attribute. The units attribute must indicate increments from a reference date/time.
  - For multi-dimensional coordinate variables, the variable name should not match the name of any of its dimensions. For example, longitude in a rectangular grid should be declared lon(y,x) with x and y declared as row/column dimensions. Longitude should not be declared lon(lat, lon)… Lat and lon should not be declared as row/column dimensions for rectangular grids. They may still be declared as dimensions for single-dimensional variables, "lon(lon)" for example.

- Cell Boundary Variables

- A coordinate variable representing cell position must have a "bounds" attribute whose value is another variable name (the boundary variable).
- The boundary variable must have the same dimensions as its associated cell position variable. In addition, the boundary variable will have a dimension to the right of these dimensions which will signify the amount of vertices in the cell.
- Boundary variables are not required to have any attributes. However, if any attributes are defined, they must match those of its associated cell position variable.
- Points specified by the cell position variable should lie within or on points specified by boundary variables.

Appendix B Glossary of Terms

Coordinate variable - A one-dimensional variable defined with a dimension of the same name. Missing values are not permitted for coordinate variables. (i.e., float lon(lon), where lon has been defined as a dimension of n size).

Auxiliary coordinate variables - Variables containing coordinate data defined with an indexing dimension not of the same name. May be multi-dimensional. (i.e., float lon(ncell), where station would be an index of n stations).

IC ISM – Intelligence Community Information Security Marking

AMOP - Administrative Model Oversight Panel (CNMOC)

OAML - Oceanographic & Atmospheric Model Library (CNMOC)