The CF metadata standard

Jonathan Gregory
Centre for Global Atmospheric Modelling,
Department of Meteorology, University of Reading, UK
and Met Office Hadley Centre, Exeter, UK

6th November 2003

"Metadata" provides a description of data. Precise and adequate metadata is enormously important in any project where large quantities of data have to be handled. If the data comes from a variety of sources, it is essential to adopt some standard for the metadata so that common programs can be used for analysis and comparison. CF is a standard for "use metadata", whose aim is to distinguish quantities (physical description, units, prior processing, etc.) and to locate the data in space–time and as a function of other independent variables (coordinates). This is the kind of metadata that is used at the time the data is processed and displayed; it can be distinguished from "discovery metadata", which is used in catalogues for identifying datasets. CF provides only rather basic discovery metadata, such as ways to record where and how the file was produced.

CF [1] has been developed over the last few years as a community project mainly by Brian Eaton, Jonathan Gregory, Bob Drach, Karl Taylor and Steve Hankin, with suggestions and comments from many others. After two years of discussion and improvement on the WWW, it has reached the stage of its first mature release (CF-1.0). Meanwhile it has been adopted as the standard for several international projects such as AMIP, CMIP, ESMF and PRISM [2-5], and by various climate centres, including the Hadley Centre (Met Office, Exeter, UK), the National Center for Atmospheric Research (NCAR, Boulder, USA) and the Program for Climate Model Diagnosis and Intercomparison (PCMDI, Livermore, USA). The purpose of this article is to commend it for consideration by other projects.

CF is framed as a standard for data written in netCDF [6], but most of its ideas relate to metadata design in general, not specifically to netCDF. CF metadata could be contained in other formats, such as XML. The adoption of a common metadata standard makes conversion between different file formats a straightforward task of mapping and translating corresponding concepts.

As a file format for data exchange, netCDF has plenty to recommend it: it is portable, binary, easily translatable to and from an equivalent ASCII format, and supported by a lot of freely available software for processing and graphics, including the netCDF library itself, CDAT, Ferret and NCO [7-9]. A utility to check conformance of a netCDF file to the CF standard has been made available by the Hadley Centre with PRISM support. At PCMDI some f90 subroutines are being developed to facilitate the writing of CF-conforming netCDF data, with the aim of making it easier for those creating simulated and observational datasets to adopt the format.

CF is intended for use with climate and forecast (hence "CF") data, for atmosphere, surface and ocean. It was designed with model-generated data particularly in mind, but should be equally applicable to observational datasets. Indeed, if observed and simulated data are to be compared, it will be helpful if their metadata describes them in the same way. The COARDS [10] netCDF standard has a similar purpose and is widely

used. It has conventions for identifying coordinate axes (longitude, latitude, vertical and time), and for specifying units and missing data values. CF is backward-compatible with COARDS: applications which understand CF can also process COARDS datasets, and CF datasets will not break applications based on COARDS. To enable this, CF is a superset of COARDS: where COARDS is adequate, CF does not provide an alternative, while extensions to COARDS are all optional and provide new functionality. The motivation for developing CF was the need for these extra features, which include conventions for grid-cell boundaries, horizontal grids other than latitude–longitude, recording common statistical operations, standardised identification of physical quantities, non-spatiotemporal axes, climatological statistics and data compression.

The general principles in the design of CF are as follows. (1) Data should be self-describing. No external tables are needed to interpret the file. For instance, CF doesn't use numeric codes, like GRIB [11] does. (2) Conventions have been developed only for things we know we need. Instead of trying to foresee the future, we have added features as required and will continue to do this. (3) We wish to avoid being too onerous for data-writers and users of data, as this will make the standard unattractive. (4) The metadata should be readable by humans as well as easily parsed by programs. (5) Redundancy is minimised—a good general principle because it reduces the chance of inconsistency—and we try also to limit possibilities for making mistakes when writing data!

Although CF-1.0 has been frozen, several additions have already been worked out and will be included soon, in the next release. We hope that more software able to make use of CF features will become available; CDAT, for example, aims to support CF fully. CF so far has been developed by a rather small group. If it is successful and becomes widely used in the community, new arrangements may be needed for making decisions and implementing developments in the standard. If CF does not seem adequate for your needs, please describe them to us! Feedback is welcome on all aspects of CF via the CF email list.

Some features of CF

In the sections below we briefly describe some of the most important conventions introduced by CF. The comprehensive definition of CF can be found on the CF home page [1]. NetCDF files contain the data in variables, which can be single numbers, vectors, or multidimensional arrays. Variables can be of any data type, including character strings. Coordinates are also contained in variables. Both data and coordinates have "attributes" attached, of any data type. We use these concepts in the following discussion without further reference to netCDF.

Description of the data

CF requires all variables to have units unless they contain dimensionless numbers. Units are strings formatted according to the Unidata udunits [12] conventions, which support many possible units and varieties of syntax e.g. percent, metre, meter, meters, m, km, second, s, day, degC, K, Pa, mbar, W m-2, kg/m2/s, mm day-1, 1 (or any number). Some non-SI units, however, are not supported e.g. ppm (1e-6 can be used). If these limitations prove a serious problem, they will be addressed.

The physical quantity contained in a data variable can be described by the long_name string attribute, as in COARDS. However, this could be anything, so to provide a systematic identification CF introduces a standard_name string attribute whose value is one from the CF "standard name table". Standard names are intended to be self-explanatory. As CF is applicable to many areas of geoscience, the names have to be more informative than would suffice for any one area. For instance, there is no name for plain "potential temperature", since we have to distinguish air_potential_temperature and sea_water_potential_temperature. The names are precise enough to imply particular physical dimensions (mass–length–time, expressed as a "canonical unit"); for example large_scale_rainfall_amount (canonical unit kg m-2), large_scale_rainfall_flux (kg m-2 s-1) and large_scale_rainfall_rate (m s-1) are all different, although they might all be vaguely referred to as "large-

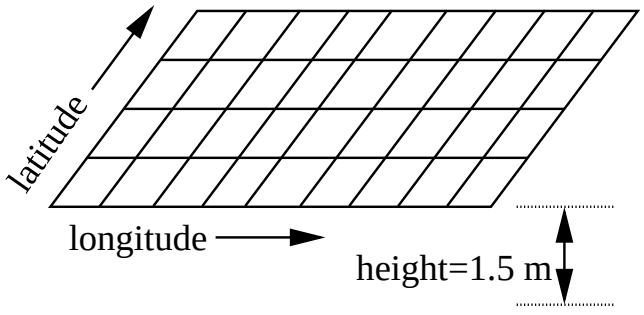
scale rain".

The standard name table currently has a few hundred entries and we expect it to grow in response to proposals, which are often made and are welcomed. The main effort in doing this is the thinking required about whether quantities are really the same and how they should best be described. In this task the expertise of specialists in the relevant areas is really vital.

Dimensions and coordinates

Dimensions establish the index space of data variables e.g. temperature(lat,lon) could have dimensions lat=73 and lon=96. Coordinates are the independent variables on which the data depend e.g. temperature could be a function of latitude and longitude, being 252.2 K at 0°E and 25.0°S. Dimensions and coordinates are closely associated, but aren't identical. You can have a dimension without coordinates, such as that distinguishing the various points or stations whose timeseries share a time axis; such an axis just has arbitrary indices rather than a continuous coordinate variable (the "point" dimension of Figure 2). Conversely, you can have a coordinate without a dimension (or a dimension of size unity), such as the height in this case:

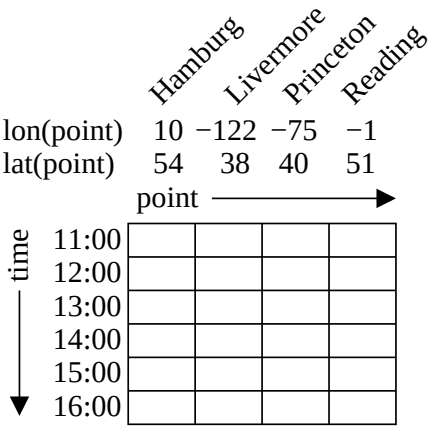
Figure 1



In COARDS there is a single monotonic coordinate variable for each dimension. This is sufficient for a basic description for a variable with dimensions such as (time,sigma,lat,lon). For dimensionless vertical coordinates like sigma (the fraction of the surface pressure), CF provides a way to record how dimensional coordinates (of pressure in this case) can be computed from the dimensionless values.

CF also introduces other kinds of variable containing coordinate information: (1) Scalar coordinate variables for single-valued coordinates, like the 1.5 m height in Figure 1. This is simply for convenience—it requires less machinery than a dimension of size unity. Single-valued coordinates are often omitted from metadata, but they are very useful—for example, the time information for a field applying to a single time (like 12:15 on 6th November 2003) can be recorded as a scalar coordinate variable. (2) Auxiliary coordinate variables, which can have any subset of the dimensions of the data variable, and are not necessarily monotonic. These provide "alternative" sets of coordinates for dimensions. For instance, we might like to label the vertical axis with model level number as well as sigma coordinate, or to provide location information and station names for the points in a timeseries:

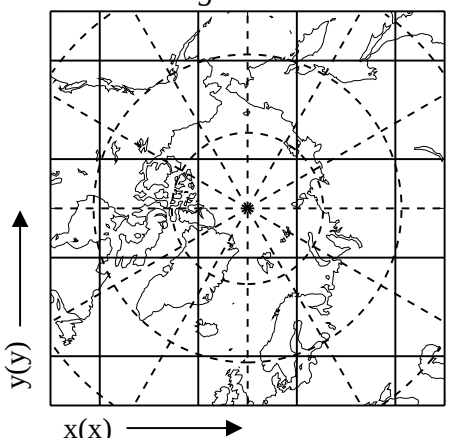
Figure 2



A very important use of auxiliary coordinates is to supply latitude and longitude for each 2D point in the situation where the x- and y-axes of the grid are not latitude and longitude e.g. if they refer to a rotated north pole, or are based on a map projection as in Figure 3 (lat-lon lines dashed, grid-lines solid). The latitude and longitude variables in this case have dimensions (y,x). CF also provides a means to describe how the grid was constructed (on a polar stereographic projection, in this case). If it can use this information, an application can work out the grid-points in latitude and longitude, and other information about the cells. However, CF requires the

latitude and longitude of the points to be recorded explicitly so that the data can be interpreted by generic applications that don't know how the projection works.

Figure 3



Following udunits and COARDS, time in CF (year, month, day, hour, minute, second) is encoded with units "*time_unit* since *reference_time*". The encoding depends on the calendar, which defines the permitted values of (year, month, day). For instance, 31st August 2003 is not a valid date in the "360-day" calendar, which has twelve 30-day months, although of course it is valid in the real-world "standard" calendar. In the standard calendar, 15:00 on 29th February 2000 is 36583.625 days since 0:00 on 1st January 1900, but it is 36058.625 days in the 360-day calendar.

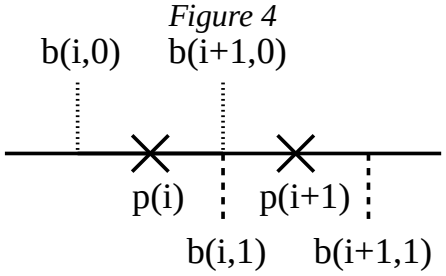
COARDS allows only the standard calendar, but many models use others. CF recognises a wide selection, including calendars for palaeoclimate simulations and perpetual seasons. Unfortunately udunits supports only the standard calendar. There are conversion routines for other calendars in CDAT. We hope to make appropriate software more readily available.

Bounds and cells

It is often necessary to know the extent of a cell as well as the grid-point location, e.g. to calculate the area of a longitude–latitude box or the thickness of a vertical layer. If grid-points are regularly spaced, their boundaries can be deduced assuming each point is in the middle of its cell. For irregularly spaced grid-points, and for single-valued coordinate variables, this can't be done. CF provides a way to attach boundary variables to any variable containing coordinate data. A boundary variable has an extra trailing dimension to index the vertices of the cells.

The simplest case is for a 1D coordinate variable (Figure 4):

Figure 4



These might be successive time-intervals, for example, with grid-points p of noon on 6th and 7th of November, and bounds b at midnight on 6th, 7th and 8th. The bounds for a 1D coordinate variable of dimension n could usually be stored in a vector of dimension (n+1). CF uses (n,2) instead as shown for technical reasons (it is convenient for use with the netCDF unlimited dimension) and because it allows the possibility of non-contiguous and overlapping cells. In fact the bounds can be used to test contiguousness. In the figure, cell i and cell i+1 are contiguous if b(i+1,0)=b(i,1). For multidimensional auxiliary coordinate variables, such as the 2D latitude and longitude variables illustrated above, we have to supply the coordinates of each vertex of the polygon and contiguousness can similarly be tested by coincidence of vertices.

CF defines a cell_methods attribute, which indicates how variation within the cells is represented. By default, it is assumed that intensive quantities apply at grid-points e.g. temperature values apply at the spatial points and instants of time specified by their coordinates, while extensive quantities are a sum over the grid-cell e.g. a precipitation amount is an accumulation in time. The method may be different for each axis e.g. precipitation amount (kg m-2) is intensive in space. The non-default methods are operations such as mean, maximum, minimum and standard deviation. A zonal-mean variable, for instance,

has a cell_methods attribute that specifies it is a mean over longitude. A timeseries of daily maximum values has a cell_methods indicating that the values are maxima within their cells in time. The operations recorded by cell_methods might affect more than one axis at once, for example the maximum of the ocean meridional overturning streamfunction within a depth–latitude cell.

A further use of cell_methods is to characterise climatological statistics. An interval of climatological time represents a set of subintervals which are not contiguous. There are three kinds: (1) Corresponding portions of the annual cycle in a set of years e.g. average January temperatures in the climatology of 1961–1991. (2) Corresponding portions of a range of days e.g. the average diurnal cycle in April 1997. (3) Both concepts at once. CF uses the cell_methods to indicate the interpretation of the time bounds. For example, the average winter-minimum temperature for 1961–1991 has bounds of 1st December 1961 (beginning of the first winter) and 1st March 1991 (end of the last), and the cell_methods indicates the values are a minimum with years, and a mean over years.

References

1 http://www.cgd.ucar.edu/cms/eaton/cf-metadata/
2 http://www-pcmdi.llnl.gov/amip/
3 http://www-pcmdi.llnl.gov/cmip/
4 http://www.esmf.ucar.edu/
5 http://www.prism.enes.org/
6 http://www.unidata.ucar.edu/packages/netcdf/
7 http://esg.llnl.gov/cdat/
8 http://ferret.pmel.noaa.gov/Ferret/
9 http://nco.sourceforge.net/
10 http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html
11 http://dss.ucar.edu/docs/formats/grib/gribdoc
12 http://www.unidata.ucar.edu/packages/udunits/