Using the LibCF/GRIDSPEC extensions to interpret data on mosaic grids with CDAT

David Kindig[1], Alex Pletzer[1], V. Balaji[2], Steve C Hankin[3], Edward J Hartnett[4],
Charles Doutriaux[5], Jeff Painter[5], Andrey Sobol[1], Matt Wrobel[1]

1 – Tech-X Corporation, 2-Geophysical Fluid Dynamics Laboratory, 3-Pacific Marine Environmental Laboratory, 4-Unidata/UCAR, 5-Lawrence Livermore National Laboratory

The Problem

- In models longitude-latitude grids create numerical stability issues related to singularities at the poles.
- **Figure 1** shows examples of logically rectangular grids which avoid the polar singularity problem.
  - Cubed-sphere (Atmosphere) (**1b,c**).
  - Tri-polar (Ocean and Ice) (**1d**).
  - Calhoun-Helzel-LeVeque (**1e**).
- These grids often involve a collection of structured grid tiles (e.g. 6 tiles for the cube sphere).
- We refer to these collections as ***mosaics*** for this poster (**1a**).
- Climate and Forecast Conventions[2] (CF) do not cover cases with data stored in separate files whether the files are time related or grid related.
- We do not address unstructured grids, which require new dynamics within the models.

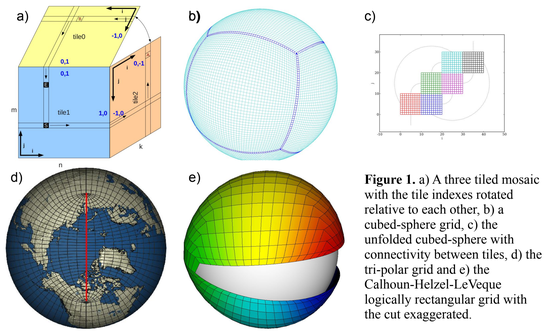


Figure 1. a) A three tiled mosaic with the tile indexes rotated relative to each other, b) a cubed-sphere grid, c) the unfolded cubed-sphere with connectivity between tiles, d) the tri-polar grid and e) the Calhoun-Helzel-LeVeque logically rectangular grid with the cut exaggerated.

GRIDSPEC

GRIDSPEC[1] was developed originally at the Geophysical Fluid Dynamics Laboratory (GFDL) by Balaji et al (http://www.gfdl.noaa.gov/~vb/gridstd/gridstdse2.html)

- Here we use the syntax to describe the connectivity between tiles from the original document. In order to accommodate CMIP5 requirements, time data aggregation was added.

We are proposing to extend the CF conventions to allow users to:

- Handle the needs of mosaics. A given variable will typically exist on multiple structured grid tiles (patches).
- Distribute data among multiple files, allowing grid information, time independent data, and data time-slices to be stored in separate files. One file (the host file) will provide a single entry point for consumers to access data scattered among multiple files.

The proposed additions to CF:

- Global attributes (**Table 1**)
  - To uniquely identify a data collection.
- To further identify a data collection we recommend the use of the CF global attributes **institution** and **source**.
- standard_name for variables
  - Add **gs_** to standard_names to allow GRIDSPEC to determine the type of file as seen in **Table 2**.
  - This allows variables to be identified as GRIDSPEC variables without reserving variable names.

A mosaic hierarchy will consist of (**Figure 2**):

- **Grid files** – block structured coordinate system for each tile.
- **Data files** – time dependent and time independent data for each tile.
- **Mosaic file** – connectivity information for the assembly of the grids.
- **Host file** – collection of file names making up the grid files, the data files and the mosaic file.

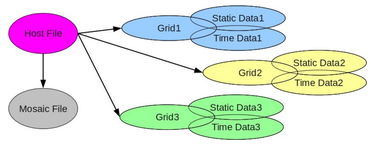


Figure 2. File hierarchy for GRIDSPEC. The Host File contains the path to all Grid and Data Files as well as the Mosaic File. The Data and Grid Files may be the same file.

Mosaics

The relationship between tiles:

- We call this relationship a map.
- Allows calculating any thing where navigating across tiles is necessary, e.g.
  - Zonal mean.
  - Laplacian of a field.
- GRIDSPEC supports contacts between tiles where the edges are parallel and there are at least two vertexes which correspond (**Figure 3**). The contact map contains the information needed to rebuild a mosaic or to translate from one tile to an adjacent tile.
- **Figure 4** shows examples of unsupported mosaics.

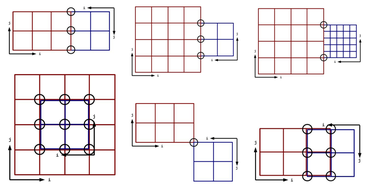


Figure 3. Supported time contacts. From right to left, top to bottom, the "C" contact_map for each contact is:
0:3,3:4::2:-1,2:3;
1:4,4:5::0:3,2:3;
1:4,4:5::0:6,5:6;
0:1,3:4::0:1,2:3;
0:3,2:4::2:-1,1:3;
1:4,1:4::0:3,2:-1;.

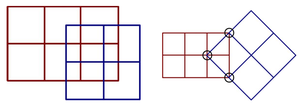


Figure 4. Unsupported Contacts

Global attribute	Describes	Example
collection_id	Unique identifier binding all files	"a54e5ee5-3232-4744-90e9-e3baddce48c3"
file_type	Type of data in file	"gs_host_file", "gs_mosaic_file", "gs_time_data_file"
tile_name	Tile name pointing to coordinates	"cubed_sphere_grid0"

Table 1. New global attributes

standard_name	File	Describes	example
gs_coordinate_names	Mosaic	array of coordinate names	"lon", "lat" , "elev"
gs_tile_contacts	Mosaic	Array of contact pairs	"top::front"
gs_contact_map	Mosaic	Array of connectivity pairs	"2:3,0:2::0:2,2:3"
gs_slice_format	Mosaic	Indexing order	"C" or "Fortran"
gs_mosaic_filename	Host	Mosaic filename	
gs_static_data_filenames	Host	Time independent data filenames	
gs_time_data_filenames	Host	Time dependent data filenames	
gs_tile_filenames	Host	Array of grid files containing grid data	
gs_tile_filenames	Both	Array of tile names	

Table 2. New standard_names

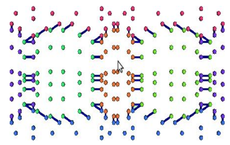
LibCF/GRIDSPEC API

- LibCF
- http://www.unidata.ucar.edu/software/libcf/
- MODaVE - GRIDSPEC wiki for more information
- https://ice.txcorp.com/trac/modave/wiki/CFProposalGridspec
- The API conforms to the LibCF conventions:
  - NetCDF3 compliant
  - C interface
- All GRIDSPEC types (Grids, Data, Mosaic and Host) have:
  - _def_ methods – Allows a user to define the type in memory
  - _put_ methods - Write the type to a file
  - _free_ methods – Free the type from memory
  - _get_ or _inq_ methods – get data or attributes from memory
- Data and Grids only need to receive the global attributes from Table 1 to become GRIDSPEC ready.
- Mosaic files must contain the following variables and attributes in order to be able rebuild the tiles with proper relationships.
  - Coordinate names
  - Tile names
  - **tile_contact** – Which tiles contact each other
  - **contact_map –** how do the indexes at the contact edges relate to each other
  - **slice_format** - "Fortran" or "C" ordering
- Host Files contain all information required to rebuild a data collection
  - Grid paths and file names
  - Data paths and file names
  - Mosaic path and filename

```
netcdf tst_three_tile_cubed_sphere_host {
dimensions:
        string = 256 ;
        nGrid = 3 ;
        nStatData = 1 ;
        nTimes = 1 ;
        nTimeData = 1 ;
variables:
        char mosaic_filename(string) ;
                mosaic_filename:standard_name = "gs_mosaic_filename" ;
        char tile_names(nGrid, string) ;
                tile_names:standard_name = "gs_tile_names" ;
        char tile_filenames(nGrid, string) ;
                tile_filenames:standard_nam

e = "gs_tile_filenames" ;
        char static_data_filenames(nStatData, nGrid, string) ;
                static_data_filenames:standard_name = "gs_static_data_filenames" ;
        char time_data_filenames(nTimes, nTimeData, nGrid, string) ;
                time_data_filenames:standard_name = "gs_time_data_filenames" ;

// global attributes:
                :collection_id = "7a782264-eb8f-408c-83b4-761229ec32b9" ;
                :file_type = "gs_host_file" ;
data:

 mosaic_filename = "tst_three_tile_cubed_sphere_mosaic.nc" ;

 tile_names =
  "tst_three_tile_cubed_sphere_grid0",
  "tst_three_tile_cubed_sphere_grid1",
  "tst_three_tile_cubed_sphere_grid2" ;

 tile_filenames =
  "tst_three_tile_cubed_sphere_grid0.nc",
  "tst_three_tile_cubed_sphere_grid1.nc",
  "tst_three_tile_cubed_sphere_grid2.nc" ;

 static_data_filenames =
  "tst_three_tile_cubed_sphere_stat_data0.nc",
  "tst_three_tile_cubed_sphere_stat_data1.nc",
  "tst_three_tile_cubed_sphere_stat_data2.nc";

 time_data_filenames =
  "tst_three_tile_cubed_sphere_time_data0.nc",
  "tst_three_tile_cubed_sphere_time_data1.nc",
  "tst_three_tile_cubed_sphere_time_data2.nc";
}
```

Example 1. Host File CDL example.

Example 2. Connectivity between tiles of a cubed-sphere.



References:

1) Gridspec: A standard for the description of grids used in Earth System models, http://www.gfdl.noaa.gov/~vb/gridstd/gridstdse2.html
2) NetCDF Climate and Forecast Metadata Convention, http://cf-pcmdi.llnl.gov/
3) Climate Data Analysis Tool http://www2-pcmdi.llnl.gov
4) Climate Model Intercomparison Project-5 http://cmip-pcmdi.llnl.gov

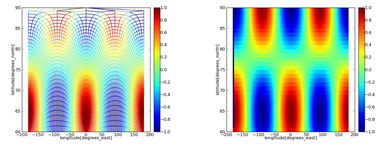
Interpolation

- Linear interpolation using nearest neighbors only
  - No over-shooting
  - Straightforward to parallelize
- Pseudo-Newton search of position in index space
- Only one iteration required for uniform, rectilinear grids
- Line search to improve convergence
- Use previous index location as initial guess when regridding from structured to structured grid

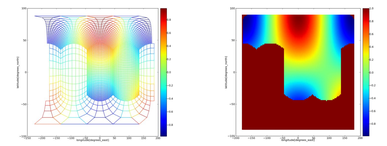
"Snake" iterator to navigate coordinate data, hopping to nearest from one vertex to neighbor vertex

- Below are two examples of interpolation using mock data. Plots obtained using matplotlib and CDAT.

Example 3. Bi-polar cap defined for lat >= 60 deg north. There are two poles at lat = 60 deg N, lon = 0, +/- 180 deg and there is a coordinate cut at -/+ 180 deg.



Example 4. Cubed Sphere to lon-lat grid. Three tiles are shown along with some target points not are not in original grid. The Lon-lat grid covers entire globe.



CDAT

f.(method)	Note
=cdms2.open	(uri, mode="r", template=None, dods=1) uri=host_file, if valid GRIDSPEC file, it will contain the global attributes from Table 1, and depending on the file, standard_names from Table 2.
listvariables()	Opening a host file will have to return a list of variables by searching the first file in each file_name type. Assumes variable homogeneity between files.
listdimensions()	Will have to search all files for dimensions incase there are different resolutions represented.
["var1]	Returns a whole variable with nTime, nx, ny, nGrid dimensions.
["var1"].shape	e.g. (6,10,20,6)
Axes	Because of structuredness of grids, could return a range.
getGrid(id)	Return the requested grid based on an index.
tile_contacts()	Return a list of tile_contacts.
contact_map()	Return a list of the contact maps.
Nccf_get_grid_ranges	Return an index on an adjacent tile.

Table 3. A set of CDAT API methods and objects.

Future Directions

- Better integration between LibCF/GRIDSPEC and the remainder of LibCF needed
- Will need support for virtual files (in memory data access) in NetCDF
- Proposing GRIDSPEC extensions to CF
  - Multi-file aggregation
  - Mosaic file connectivity
  - Support for staggered data
- Need to support face and edge centered data (Arakawa C/D)
- Supergrids?
- Become an integral part of the NetCDF library

© 2009 Tech-X Corporation

AGU Fall Meeting 2010
Dec 13 – 17, 2010

Work funded by MoDAVE: DOE/SBIR
DE-FG02-08ER85153

U.S. DEPARTMENT OF ENERGY
Office of Scienc