CF meeting
9-11 June 2020
Zoom session link

Discussion topic notes

"mesh variable" instead of "boundary variable" for contiguous grid cells (Ryan Abernathey)

https://github.com/cf-convention/discuss/issues/5

Most numerical atmospheric and ocean models use some form of staggered curvilinear grid discretization (e.g. Arakawa grids), in which variables are located at some specific point (e.g. cell center, cell face) with respect to a finite volume quadrilateral (grid cell). CF conventions currently recommend encoding the geometry of these cells via the “boundary variable” convention. Boundary variables are extremely general, accommodating arbitrary unstructured, non-contiguous cell geometry, requiring N boundary vertices to be specified for each grid point for N-dimensional geometry. For contiguous curvilinear coordinates, this representation is inefficient in terms of memory usage, due to the redundancy in the vertices of neighboring contiguous cells. For very large grids, this inefficient representation can become a bottleneck for analysis and visualization.

We propose to extend CF conventions with a new mechanism to represent such cell geometries which lies closer to how models represent their grids internally. We wish to incorporate the concept of a “mesh variable,” which describes the boundaries of contiguous grid cells in terms of cell vertices. The sgrid conventions provide an excellent template for such representation, so we will propose to incorporate some or all of SGRID into CF proper. However, as discussed in the github issue, several thorny technical issues would benefit from community discussion.

(Chris Barker): There is also the UGRID conventions (`(http://ugrid-conventions.github.io/ugrid-conventions/`) for representing unstructured grids. I suggest that from a CF perspective, SGRID and UGRID be handled similarly.

Notes from discussion:

- Should UGRID and SGRID use “cf_role” to define the “topology” variable?
    - UGRID uses: cf_role = "mesh_topology"
    - SGRID uses: cf_role = “grid_topology”
- Coordinate variables are optional: should they be?

Note from Ryan: recent events in the US have caused my plans to change. In response to the murders of George Floyd, Breonna Taylor, and countless other Black people, today there will be a general strike across STEM and academia (https://www.shutdownstem.com/). The idea is to pause business as usual and take action against racism. I support this issue strongly, and so I will not participate in the CF discussion today.

To make up for my absence, I have put my presentation online (https://speakerdeck.com/rabernat/cf-mesh-coordinates-proposal) and recorded a video of myself presenting it (https://vimeo.com/427709117). I know this is not the same as being there “in person,” but I hope it is enough. I will be happy to resume the discussion tomorrow via the original github issue.

David (post-discussion thoughts)
Another way to look at this would be to say that complex mesh coordinates should be 1D variables, eg meshlat(node), meshlon(node), and then reconstruct the actual 2D or 3D mesh from metadata attributes. This would accommodate any type of grid topology and delegate the problem of compression / decompression to software. This seems to be what UGRID is doing.

(Chris B): yes, UGRID does that. And in principle any grid can be represented is this way, as an unstructured grid. But for grids that are “logically” rectangular, it is nice to preserve that structure: i.e. use 2d coordinate variables, which is why SGRID does that.

Otherwise additional attributes could describe which variables play the role of mesh coordinates:

```
    “edge_coordinates”/”mesh_coordinates” as an additional attribute similar to “coordinates”

    c(lat, lon):
        mesh_coordinates: meshlat, meshlon

    meshlat(lat_node, lon_node):
        node_dimension: lat

    meshlon(lat_node, lon_node):
        node_dimension: lon
```

Jim Biard:

- SGrid has done a great job of covering a number of different use cases.
- Concern about the SGrid use of the cf_role attribute name in the “grid” variable.
- It seems that it might be good to specify a way to directly name the “edge” coordinate variables in the data variable attributes — either allow the edge coordinate name to be added to the “coordinates” attribute, or through another attribute such as “edge_coordinates” mentioned above.
    - (Chris) I think this is already in the SGRID (and UGRID) spec, as optional. Maybe it should not be optional?

- A lighter-weight solution such as described by David might be a good approach.

Chris Barker: Specific proposal:

- Have CF “bless” UGRID and SGRID standards.
- Specify that cf_role be used to specify the mesh variable.
-