Earth System Curator: Integration technology for Earth system models and data

V. Balaji
Princeton University and NOAA/GFDL

ESMF Community Meeting
Massachusetts Institute of Technology
Cambridge MA
21 July 2005

The routine use of Earth System models in research and operations

Operational forecasting model-based ***seasonal*** forecasts delivered to the public;

Decision support models routinely run for decision support on climate policy by governments, for energy strategy by industry and government, as input to pricing models by the insurance industry, etc.

Fundamental research the use of models to develop a predictive understanding of the earth system and to provide a sound underpinning for all applications above.

This will require a radical shift in the way we do modeling: from the current dependence on a nucleus of very specialized researchers to make it a more accessible general purpose toolkit. This requires ***an infrastructure for moving the building, running and analysis of models and model output data from the "heroic" mode to the routine mode***.

Toward an Earth System Model Environment

- Standards for model configuration
- Standards for model output data
- What are the difficulties currently faced in uniting diverse models and datasets (e.g in IPCC 2007)?
- First steps: model grid metadata standard.

What is the Earth System Curator?

- Future projections of climate are performed at many sites, and a key goal of current research is to reduce the uncertainty of these projections by understanding the differences in the output from different models.

- This ***comparative study of climate simulations*** (e.g IPCC) across many models has spawned efforts to build uniform access to output datasets from major climate models, as well as modeling frameworks that will promote uniform access to the models themselves.

The key element in the integration will be the Earth System Curator (ESC). ESC begins with a crucial insight: that the descriptors used for comprehensively specifying a model configuration are needed for a scientifically useful description of the model output data as well. Thus ***the same attributes may be used to specify a model as well as the model output dataset***: thus leading to a ***convergence of models and data***.

ESC is best considered a pilot project building prototype elements of a future ESME. The current project is to be funded by NSF and brings together NCAR, Princeton, MIT and GA Tech.

ESMF's metadata-laden data structures

Earth system models can broadly be described as composed of components in which physical quantities are integrated on a physical grid. In a framework like ESMF, these are described in terms of 5 layers of abstractions consisting of ***metadata-laden data structures***. These layers are:

`grid` describes the physical grid in a standard way, so that component-neutral regridding software can be used to transform quantities from one grid component to another, with no knowledge of those components themselves. We seek to inscribe the grid metadata within community standards and conventions, so that analysis tools cognizant of these conventions may take advantage of grid information.

`field` consists of the physical variable discretized on a **`grid`**, along with metadata describing the physical quantity itself. The field metadata in ESMF have been designed to resemble the CF convention, so that CF-compliant model output may be produced if desired.

`attribute` configuration attributes of a component: these are very generic, but are intended to contain all the physical input parameters used to configure a model.

ESMF's metadata-laden data structures

`state` is the instantaneous state of some set of **`field`**s within a model component. Typically these are used as part of "import" and "export" states that are exchanged between components; but they are often used to contain the entire model state as well.

`component` the top level entity of this design. Components are hierarchical: that is, they may be composed of other components. The top-level **`component`** is the application or model itself.

These software layers exist in the ESMF, and ESMF-compliant models in the near future will be using these abstractions, rich in metadata, to describe a wide range of models across the weather and climate community. Simply by using these abstractions and encoding them in model output, we are creating a layer of ***formal, structured, hierarchical metadata***. We call this the ***model metadata layer***, and it is the core of the Curator. The model metadata layer is what makes possible for either a fully-configured model configuration or a model dataset to be the result of a database query.

Comparative study of model metadata

There is a controversy within the community about the feasibility and advisability of treating components as interchangeable bits of code that can be slotted together at will. An understanding of component diversity will mark the limits of such an approach. We seek to answer two questions:

- Are components ostensibly labelled “atmosphere”, say, sufficiently similar that a single physical interface may be defined? Or, to put it another way, to what extent to two such components share a `state`?

- Do different modeling organizations see component granularity the same way? What incompatibilities are introduced if one model treats atmospheric chemistry say, as an indivisible entity within an atmosphere component, whereas another treats them as independent `component`s?

The ESMF component database

The Curator will contain tools to generate model metadata from the ESMF component database in two ways:

- a ***registry tool***, where the model developer enters the information that makes up the model metadata;
- or a ***source-scan tool***, which uses knowledge of ESMF data structures to extract this information automatically.

ESMF components are well-structured enough to make both approaches feasible. It is likely that the final tool will be a hybrid of both approaches, a machine pass followed by a human pass.

The tool will determine whether:

- it is ***technically feasible*** to use a component in an application: does it run on the target platform, and so on;
- it is ***physically feasible*** to use a component: are the range of available resolutions sufficient for the problem at hand; do the physical subcomponents match the problem under study, etc;
- it is ***compatible*** to use a component with other components in the application: do the available output fields match the required input field of the other components, etc.

Linking model and data frameworks

Community data frameworks are under development, at various institutions, informally linked by the Global Organization for Earth System Science Portals (GO-ESSP). For model output data to be scientifically useful, the researcher must have some knowledge of how the data was produced. Model data requires a ***model's eye view*** description of the data, another layer of metadata, which includes:

- Description of model components: e.g FMS atmosphere, land and sea ice coupled to MIT ocean.
- Description of grid configurations and resolutions.
- Choice of physics packages and input parameters.
- Model state and its fields.

ESMF and PRISM are emerging standards that allow the development of the model metadata layer, based on the state data structures and its base classes. **Modeling framework data structures map directly on to community hierarchical metadata.** Observational data has an analogous data structure within ESMF as well: the ***location streams*** used in data assimilation.

Convergence of models and datasets

Given the existence of a model metadata layer, ***the same descriptor can be used as model input and model output.*** This means:

- the files that are used to configure, build and launch a model (written in, say, XML) contain the same physical information that must be written to the output dataset for a comprehensive description of how the data was generated.

- This information can also be stored in a relational database of model configurations and datasets: the Earth System Model Curator. Such a DB would allow experiment comparisons, high-level queries, experiment re-design, next-generation publication of scientific results.

Potential use scenarios

Climate scientists setup (assemble components, configure input parameters); comparisons (run configurations, results, with data); branch runs, ...

Impacts studies query models by pattern, couple biogeochemistry model either offline with dataset or online with model.

IPCC, MIPs descriptions of intercomparisons, setup new MIPs, archive MIP results.

Policymakers, industry and educators High-level access to swathes of model data.

Publication link datasets to publications; introduce interactive aspect to publication; annotation of data, certification and quality control.

Portability automatic best-practice configuration appropriate for platform.

Operations higher rate of technology transfer from research to operations.

Toward an Earth System Modeling Environment (ESME)

We seek to unite the data (GO-ESSP) and model (ESMF/PRISM) communities with climate scientists (IPCC, CMIP) to develop the model metadata layer, and the relational database of models and data that would be based on it.

Physical interfaces development of comprehensive physical interfaces for model components.

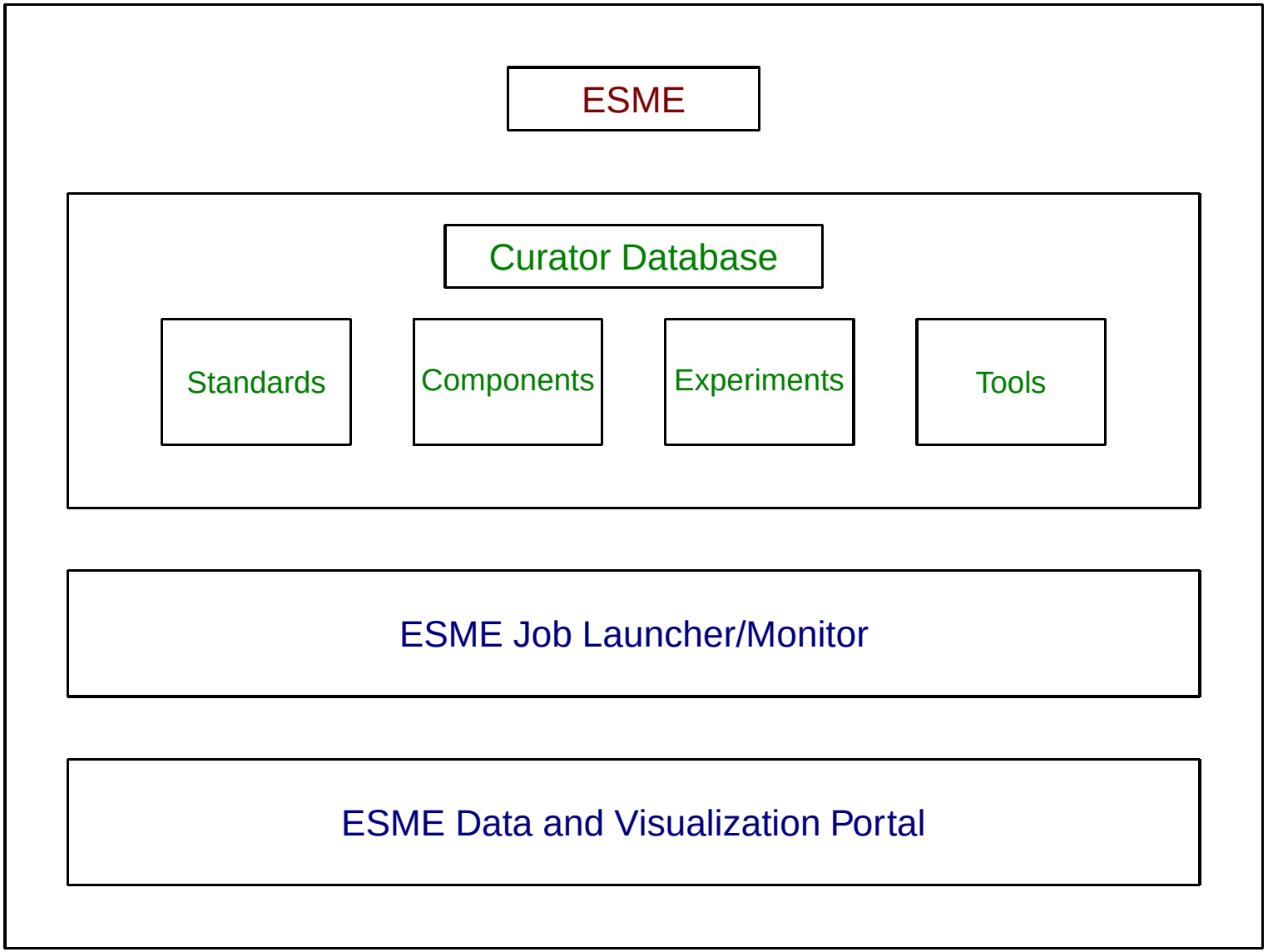
Hierarchical metadata development of a ***semantic web*** of model and data descriptors.

Relational database of model experiments and observational and model datasets.

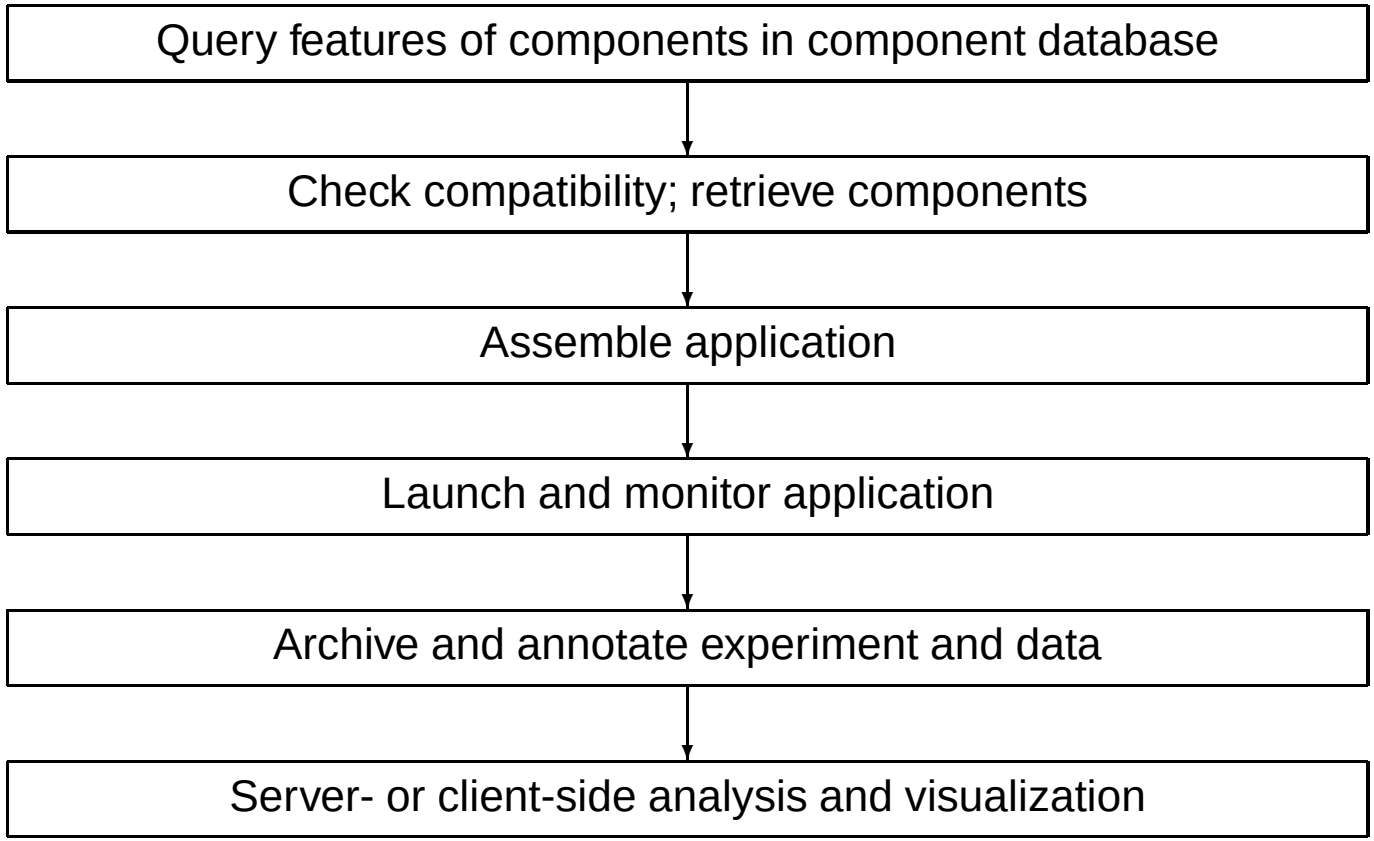
Data annotation certification by assigned authority, or *à la* Google. Links with scientific results and peer-reviewed literature.

Web portal interfaces to query operations, comparisons, client- and server-side data analysis.

Structure of the ESME



ESME workflow



Data consumers

Scientists perform sequences of computations (e.g ***"poleward heat transport", "length of growing season"***) on datasets. Typically this is scripted in some data analysis language, and ideally it should be possible to apply the script to diverse datasets.

The IPCC data archive at PCMDI has been a success for consumers without precedent, and will be cited in many groundbreaking works of climate research for many years to come.

Data producers

Observational and model output data in the climate-ocean-weather (COW) community is initially generated in native format, and any subsequent relative analyses requires considerable effort to systematise. Issues include moving and transient data sources, lossy data formats, curvilinear and other “exotic” coordinates.

Data managers

Data managers are the community within this ecosystem that facilitates the transformation of source dependent data to a neutral and readily consumable form. They develop the standards for describing data in a manner that permits these transformations, and develop tools to perform them.

The data ecosystem



We identify three communities: data ***producers***, ***managers*** and ***consumers***.

- Data is created in a manner most suited to producers (models, observations).
- It is delivered to consumers in a manner where data from different sources can be merged and coherently analysed.
- The manager niche in the ecosystem should take responsibility for mediating between these two communities. This is where CF, GO-ESSP play a role.

The key issue is to make it possible not only to display, but to construct a scientific study using, data from different sources, based on the datasets alone.

Standards play a role...

Model metadata: describing data source comprehensively, relatively easy for observations, harder for models but can asymptote toward completeness starting from current PCMDI standard

Physical fields: standard vocabulary for describing the relevant physical quantities (viz. CF `standard_name`). Variables can contain ***gridded*** or ***point*** (station, drifter) data.

Geospatial information: location information. This set of standards unites a much larger community (mobiles, GIS), in which our community has begun to play a role.

Grid metadata: interrelations between grids, between points and grids.

Grid metadata: what's included

The grid specification includes ***distances, areas, angles and volumes*** for decomposing thin spherical shells or cartesian space.

It could also contain specifications for ***exchange grids*** and ***masks***.

We apply thin-fluid scaling arguments to separate out the vertical. The vertical coordinate can be space- or mass-based.

LRGs and UPGs

LRG logically rectangular grid.

STG structured triangular grid.

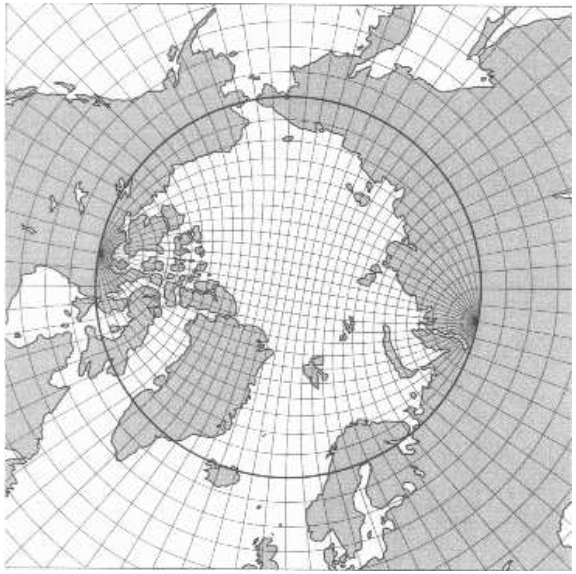
UTG unstructured triangular grid.

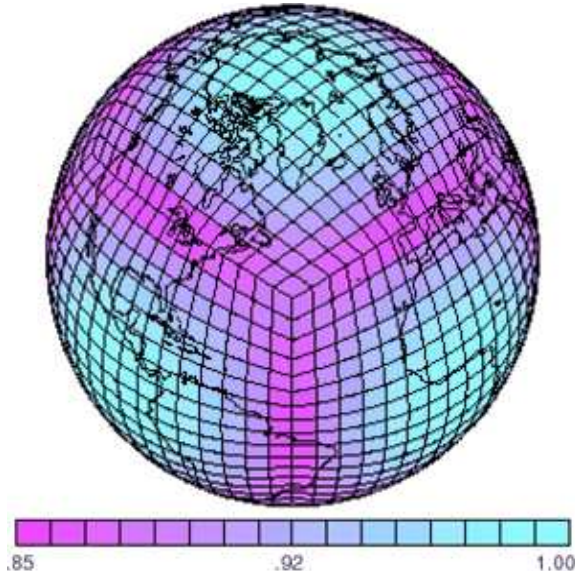
UPG unstructured polygonal grid.

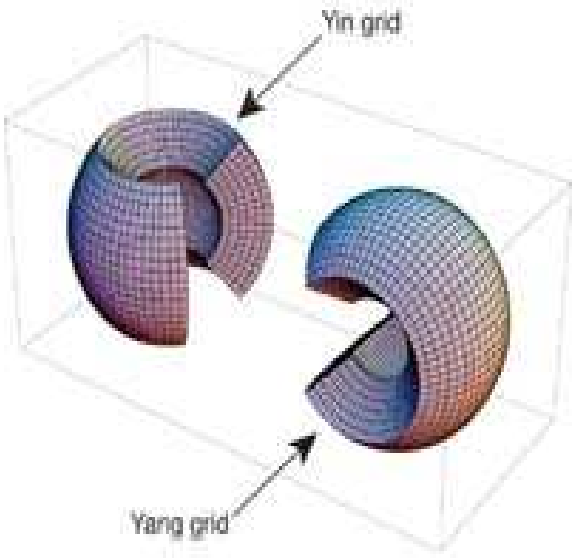
These allow us to describe all conceivable grids for the near- to mid-future.

An actual grid may be composed of a ***mosaic*** of LRGs or UPGs. In principle, you could even mix them (i.e define a grid with some LRG and some UPG tiles).

Non-”standard” LRGs



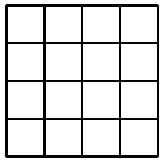




On the left is the ***tripolar grid*** (Murray 1996, Griffies et al 2004) used by MOM4 for GFDL’s current IPCC model CM2. In the middle is the ***cubed sphere*** (Rancic and Purser 1990) planned for the Finite-Volume atmosphere dynamical core for the next-generation GFDL models AM3 and CM3. On the right is another promising grid, the ***yin-yang grid*** (Kageyama et al 2004).

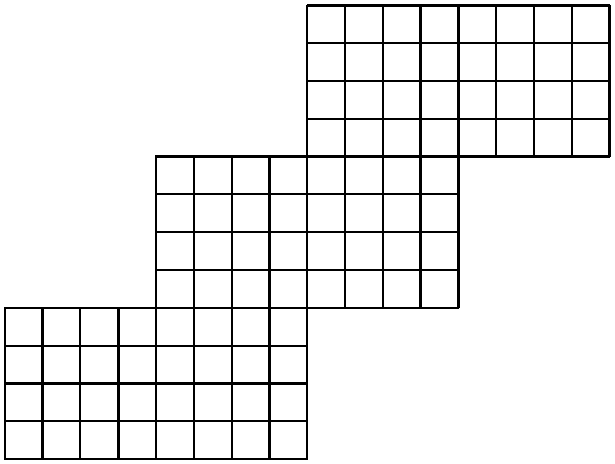
A key difference between these is that the tripolar grid is a single LRG.

What is a mosaic?

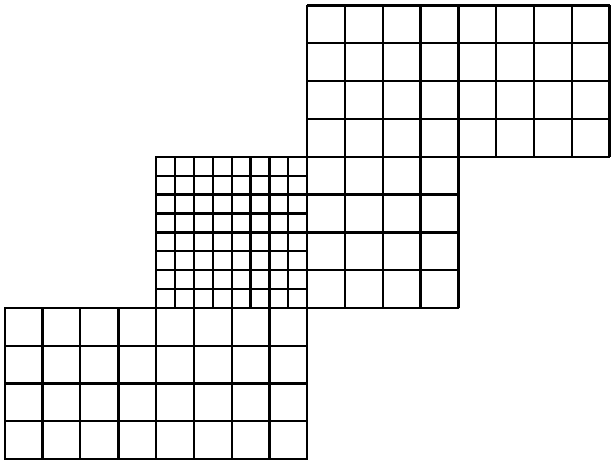


On the left is a basic 4×4 ***tile***; on the right are examples of grids composed of a mosaic of such tiles. The first is a ***continuous grid***, below is a ***refined grid***.

Most current software only supports what we call ***tiles*** here. The **`mosaic`** extension will allow the development of more complex grids for next-generation models.
First in our (GFDL's) sights is the ***cubic sphere***, primarily targeted at a next-generation finite-volume atmospheric dynamical core, but potentially others as well.
Further developments will include support for irregular tiling (e.g of the ocean surface following coastlines), and for refined, nested and adaptive grids.
Also, regular grids where an irregular decomposition is needed (e.g for a polar filter) can use mosaics to define different decompositions in different regions.

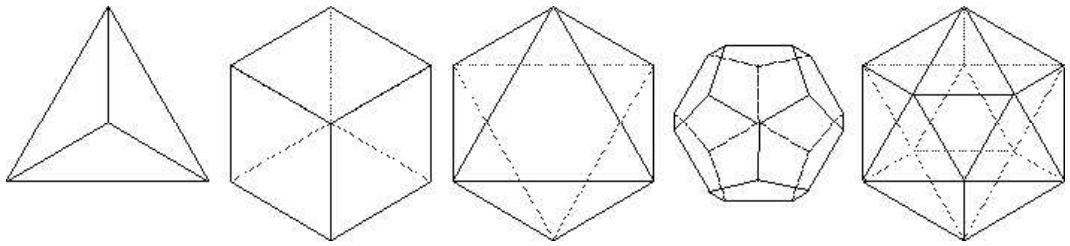


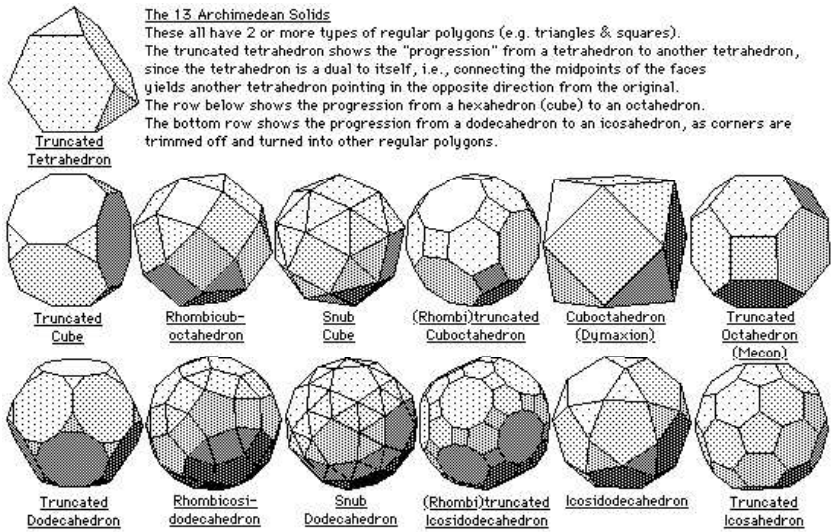
Regular grid mosaic.



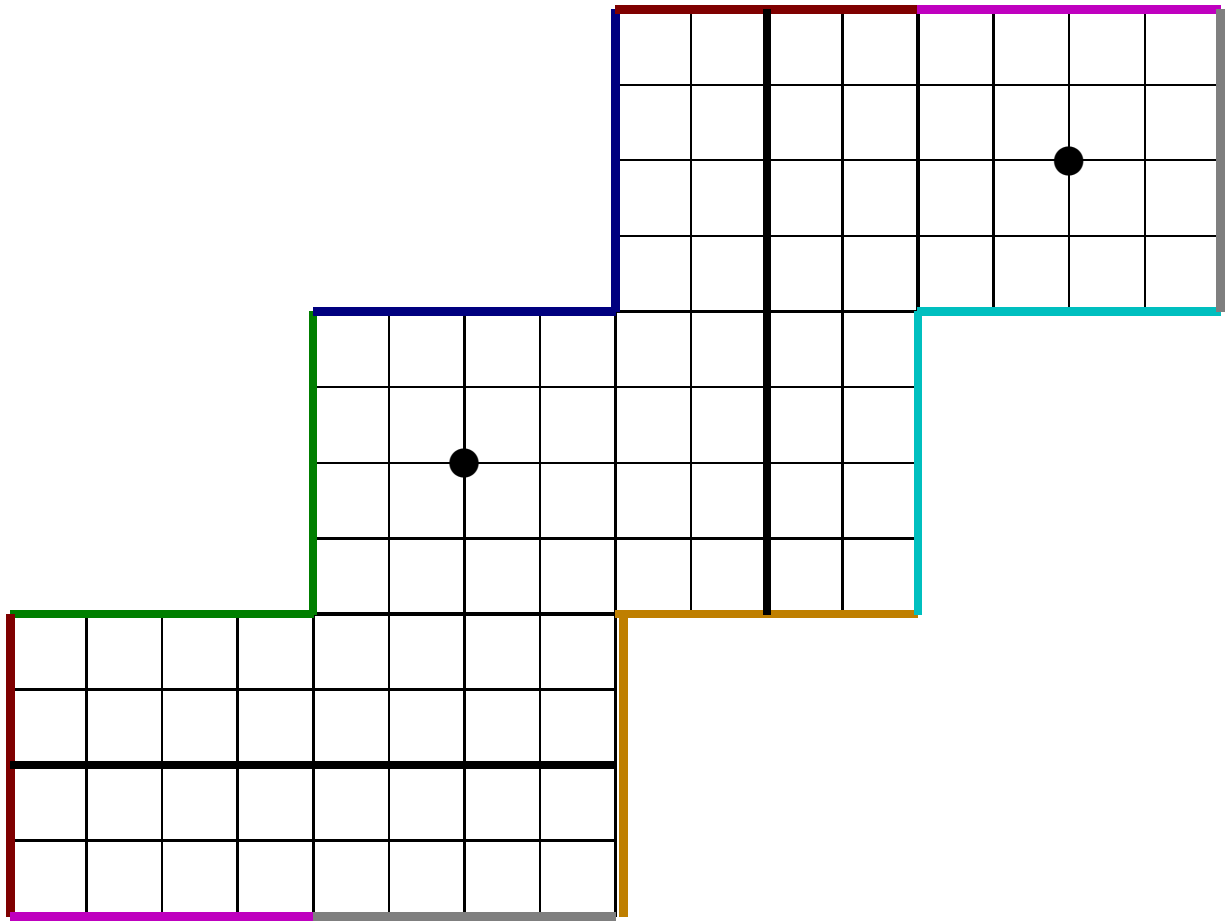
Refined grid mosaic.

UPG mosaics are likely to become more common



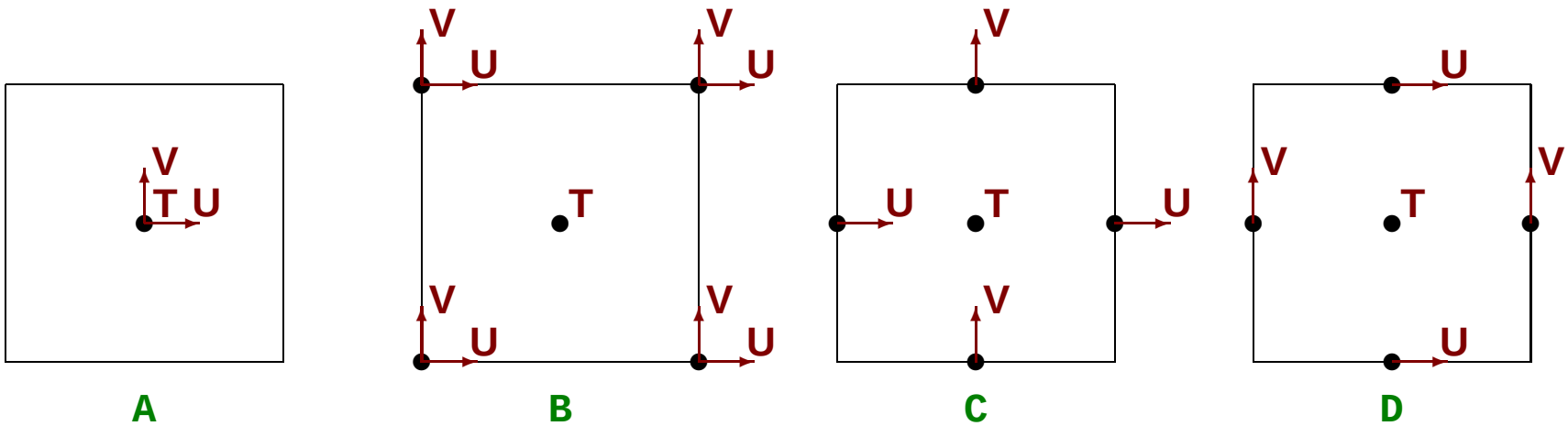


Cubed sphere



Mosaic topology for the cubed sphere. Note that boundaries may change orientation: the point just to the “west” of **(5,6)** is in fact **(3,4)**; and furthermore vector quantities transiting the boundary at that point will undergo rotation.

Supergrids

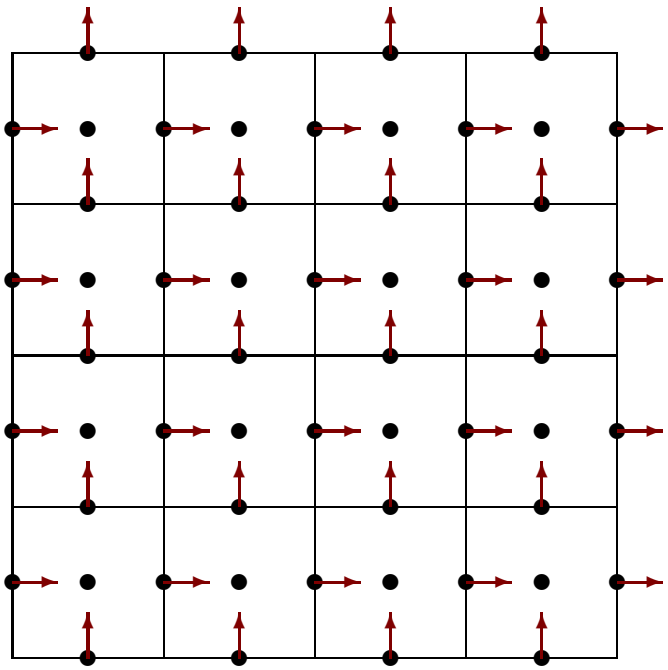


Algorithms place quantities at different locations within a grid cell (“staggering”). This has led to considerable confusion in terminology and design: are the velocity and mass grids to be constructed independently, or as aspects (“subgrids”) of a single grid? How do we encode the relationships between the subgrids, which are necessarily fixed and algorithmically essential?

In this specification, we dispense with subgrids, and instead invert the specification: we define a **supergrid**. The ***supergrid*** is an object potentially of higher refinement than the grid that an algorithm will use; but every such grid needed by an application is a subset of the supergrid.

LRG supergrids are themselves LRGs while a UPG supergrid can always be described by a unstructured triangular grid (UTG).

Specifying a single tile: C-grid LRG



The supergrid is defined as 9×9:

```
gridspec_version = 0.1
nx = 9
ny = 9
geographic_latitude(nx,ny)
geographic_longitude(nx,ny)
grid_east_angle(nx,ny)
grid_north_angle(nx,ny)
dx(nx-1,ny)
dy(nx,ny-1)
area(nx-1,ny-1)
intend_x_refinement = 2
intend_y_refinement = 2
```

(1)

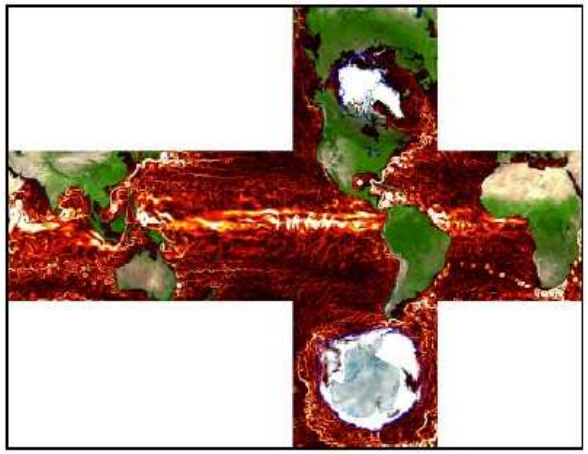
Optional keywords **uniform**, **orthogonal** are used to compress the spec.

```
gridspec = "gridspec.nc"
   checksum = "..."
nx_u = 1:nx:intend_x_refinement
ny_u = 2:ny:intend_y_refinement
grid_eastward_velocity(nx_u,ny_u)
```

(2)

From tiles to mosaics

If each tile is written out separately, current software is already capable of displaying results:



Any computation that crosses a tile boundary involves the specification of ***contact regions*** between tiles. Contact regions cannot necessarily be deduced from geospatial information.

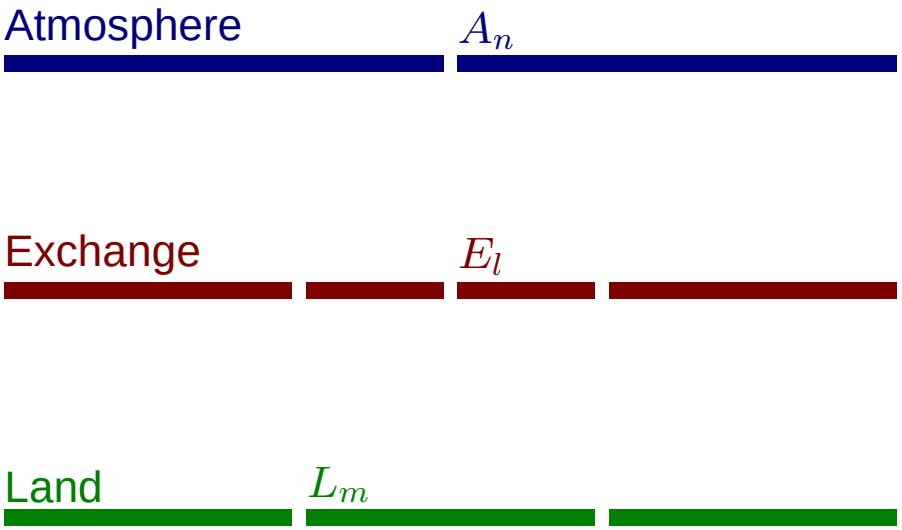
```
mosaic_version = 0.2
mosaic "atmos"
   grid_type "cubed_sphere"
   grid_mapping
   tile "face1"
   ...
mosaic "ocean"
   grid_type "tripolar_grid"
   grid_mapping
   tile "tile"
contact_region "atmos:face1" "ocean:tile"
   ncells
   parent(ncells,2)
   frac_area(ncells,2)
   mask
```

(3)

Regridding is a key operation that requires contact region information. ***Conservative*** regridding between multiple ***components*** or ***nests*** requires ***exchange*** and ***mask*** information.

Definition of an exchange grid

- A ***grid*** is defined as a set of ***cells*** created by ***edges*** joining pairs of ***vertices*** defined in a discretization.
- An ***exchange grid*** is the set of cells defined by the union of all the vertices of the two parent grids, and a ***fractional area*** with respect to the parent grid cell.



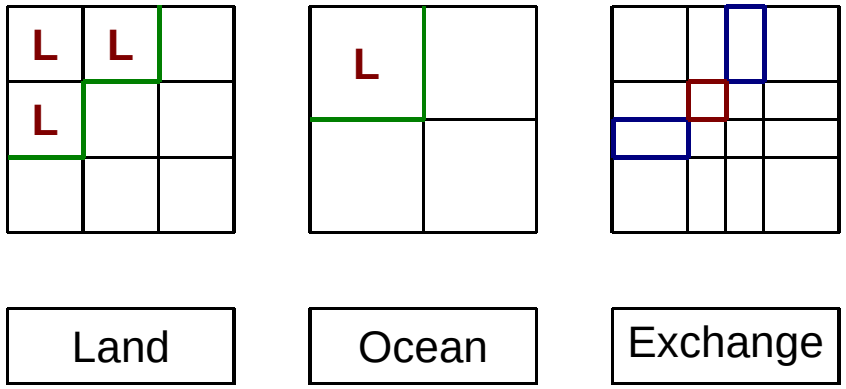
- Exchange: interpolate from source grid using one set of fractional areas; then average onto the target grid using the other set of fractional areas.

- Consistent moment-conserving interpolation and averaging functions of the fractional area may be employed.

Masks

Complementary components: in Earth system models, a typical example is that of an ocean and land surface that together tile the area under the atmosphere.

Land-sea mask as discretized on the two grids, with the cells marked **L** belonging to the land. Certain exchange grid cells have ambiguous status: the two blue cells are claimed by both land and ocean, while the orphan red cell is claimed by neither.



Therefore the mask defining the boundary between complementary grids can only be accurately defined on the exchange grid.

In the FMS exchange grid, by convention (and because it is easier) we generally modify the land grid as needed. We add cells to the land grid until there are no orphan “red” cells left on the exchange grid, then get rid of the “blue” cells by ***clipping*** the fractional areas on the land side.

Extensions to current grid specification in CF

- The specification of a tile is consistent with the current CF gridspec, but extends it by defining the supergrid and staggering.

- The definition of a mosaic is new. ***The mosaic specification can help widen the parallel I/O and filesize bottlenecks***.

- The grid specification is maintained separately from the dataset, which links to it. Integrity of linkages between files is maintained by adding a **`checksum`** attribute to each linked file.

- If the gridspec file is standardized, it can be used for model input as well as output. For coupled or nested models, this file may also contain the necessary data to relate component grid mosaics.

What's needed next

- prototype and test across more than one institution;
- CF to agree to an extended standard for gridded datasets;
- PRISM/ESMF to agree to produce compliant data;
- Tools to become capable of applying standard and bespoke regridding techniques.
- We will propose a draft standard, a compliant `gridspec.nc` file, and sample datafiles consistent with the gridspec.