The FMS coupler architecture

V. Balaji

Ocean Model Couping Workshop

Princeton NJ

7 February 2007

Coupler

Used for data exchange between models. Key features include:

Conservation: required for long runs.

Resolution: no constraints on component model timesteps and spatial grid. Supports both explicit and implicit timestepping.

Exchange grid: union of component model grids, where detailed flux computations are performed (Monin-Obukhov, tridiagonal solver for implicit diffusion, ...)

Fully parallel: Calls are entirely processor-local: exchange software will perform all inter-processor communication.

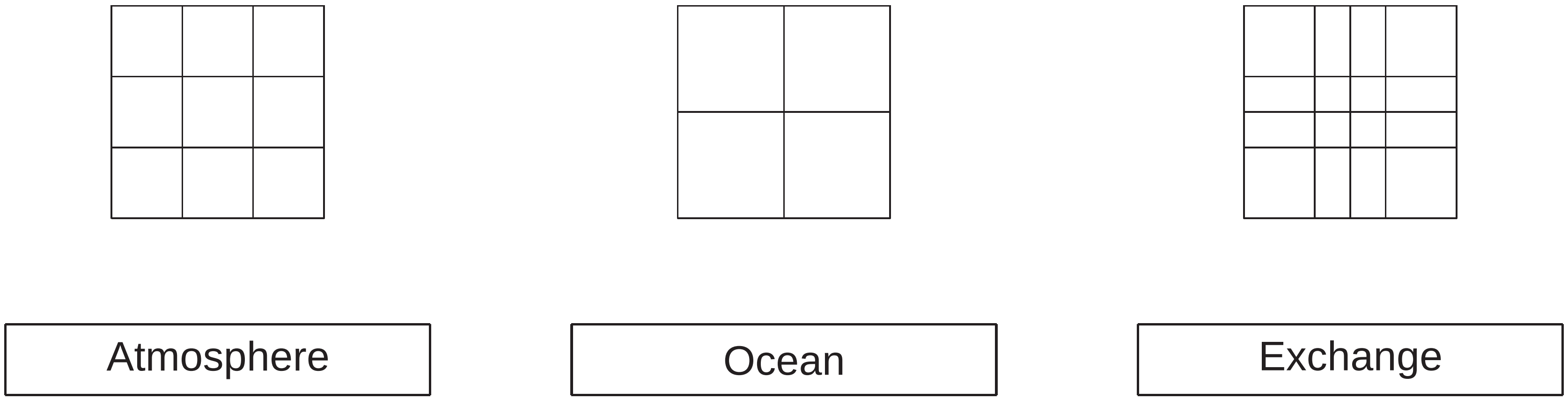
Modular design: uniform interface to main calling program.

No brokering: each experiment must explicitly set up field pairs.

Single executable:

Highly efficient: currently able to couple atmos/ocean at 3h intervals, atmos/land/ice implicitly at each timestep at current dec/cen resolutions.

Exchange grid



- Each cell on exchange grid ”belongs” to one cell on each parent grid;
- Conservative interpolation up to second order;
- All calls exchange local data; data-sharing among processors is internal to the exchange software, and non-blocking.
- Physically identical grids (e.g ocean and sea ice) exchange data without interpolation.

Features of the FMS coupler

- Encapsulated boundary state and boundary fluxes.

- Single location for initialization and linking of boundary fields.

- Use of field manager to organize operations on individual fields and field bundles.

- Support for serial and concurrent coupling within single executable.

- Implicit coupling between land-ocean surface and atmosphere on atmospheric timestep; explicit coupling between ocean surface and ocean on coupling timestep.

coupler_main slow loop

```
do nc = 1, num_cpld_calls
   call generate_sfc_xgrid( Land, Ice )
   call flux_ocean_to_ice( Ocean, Ice, Ocean_ice_flux )
   call update_ice_model_slow_up( Ocean_ice_flux, Ice )
!fast loop
   call update_land_model_slow(Land)
   call flux_land_to_ice( Land, Ice, Land_ice_flux )
   call update_ice_model_slow_dn( Atmos_ice_flux, Land_ice_flux, Ice )
   call flux_ice_to_ocean( Ice, Ice_ocean_flux )
   call update_ocean_model( Ice_ocean_flux, Ocean )
enddo
```

`coupler_main` fast loop

```
do na = 1, num_atmos_calls
   Time = Time + Time_step_atmos
   call sfc_boundary_layer( Atm, Land, Ice, &
                            Land_ice_atmos_flux )
   call update_atmos_model_down( Land_ice_atmos_flux, Atm )
   call flux_down_from_atmos( Time, Atm, Land, Ice, &
                              Land_ice_atmos_flux, &
                              Atmos_land_flux, Atmos_ice_flux )
   call update_land_model_fast( Atmos_land_flux, Land )
   call update_ice_model_fast( Atmos_ice_flux, Ice )
   call flux_up_to_atmos( Time, Land, Ice, Land_ice_atmos_flux )
   call update_atmos_model_up( Land_ice_atmos_flux, Atm )
enddo
```

Example: ocean boundary

```
type ocean_boundary_data_type
   type(domain2D) :: Domain
   real, pointer, dimension(:,:) :: t_surf, s_surf, sea_lev, &
        frazil, u_surf, v_surf
   logical, pointer, dimension(:,:) :: mask
   type (time_type)                  :: Time, Time_step
end type ocean_boundary_data_type

type, public :: ice_ocean_boundary_type
   real, dimension(:,:), pointer :: u_flux, v_flux, t_flux, q_flux
   real, dimension(:,:), pointer :: salt_flux, lw_flux, sw_flux, lprec, fpr
   real, dimension(:,:), pointer :: runoff, calving
   real, dimension(:,:), pointer :: p
   real, dimension(:,:,:), pointer :: data
   integer :: xtype             !REGRID, REDIST or DIRECT
end type ice_ocean_boundary_type
```

Flux exchange

Three types of flux exchange are permitted: **`REGRID`**, **`REDIST`** and **`DIRECT`**.

REGRID physically distinct grids, requires exchange grid.

REDIST identical global grid, different domain decomposition.

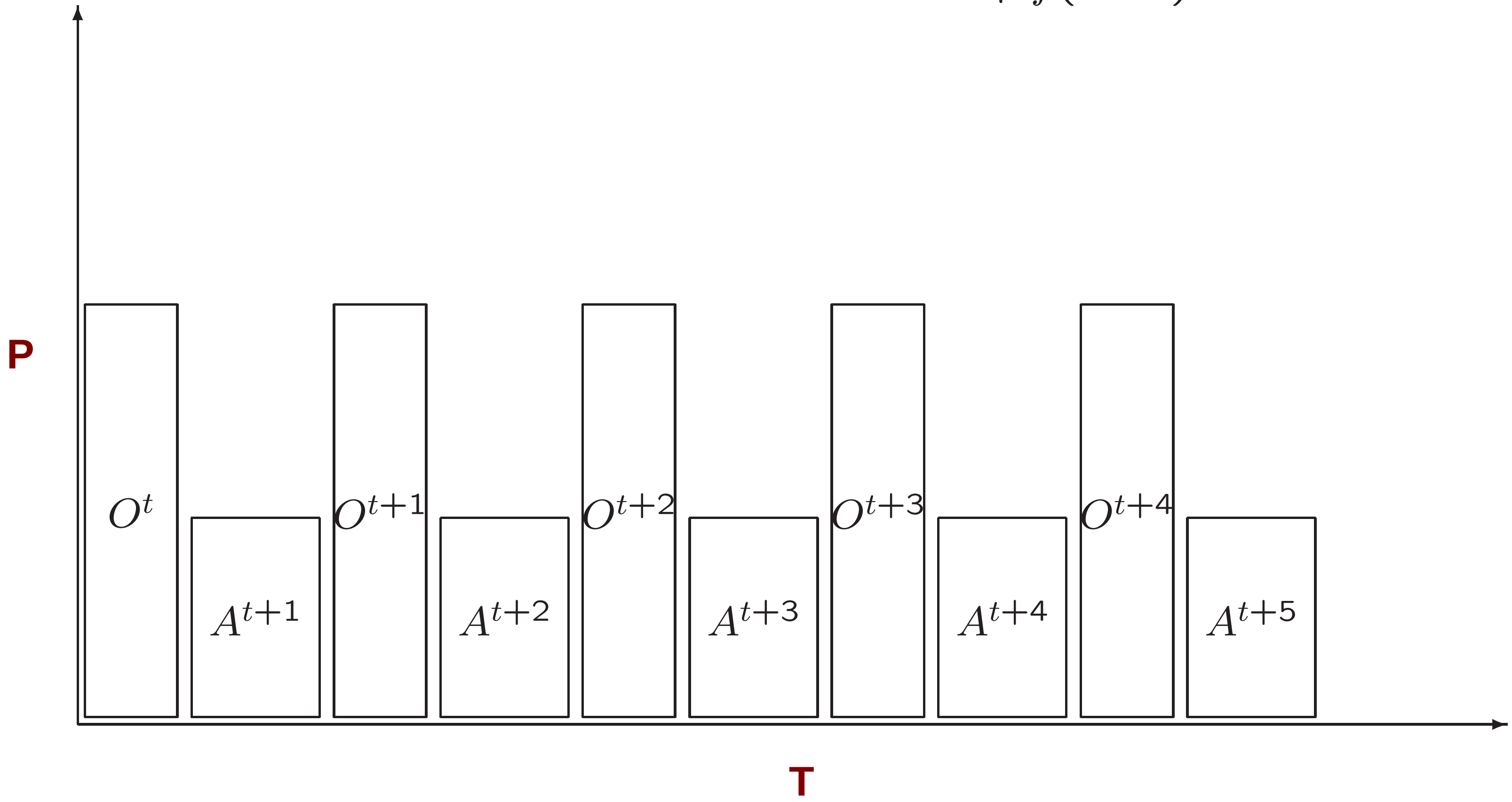
DIRECT identical grid and decomposition.

Current use: **`REGRID`** between atmos$\Longleftrightarrow$ice, atmos$\Longleftrightarrow$land, land$\Longleftrightarrow$ice, **`REDIST`** between ocean$\Longleftrightarrow$ice.

Serial coupling

Uses a forward-backward timestep for coupling.

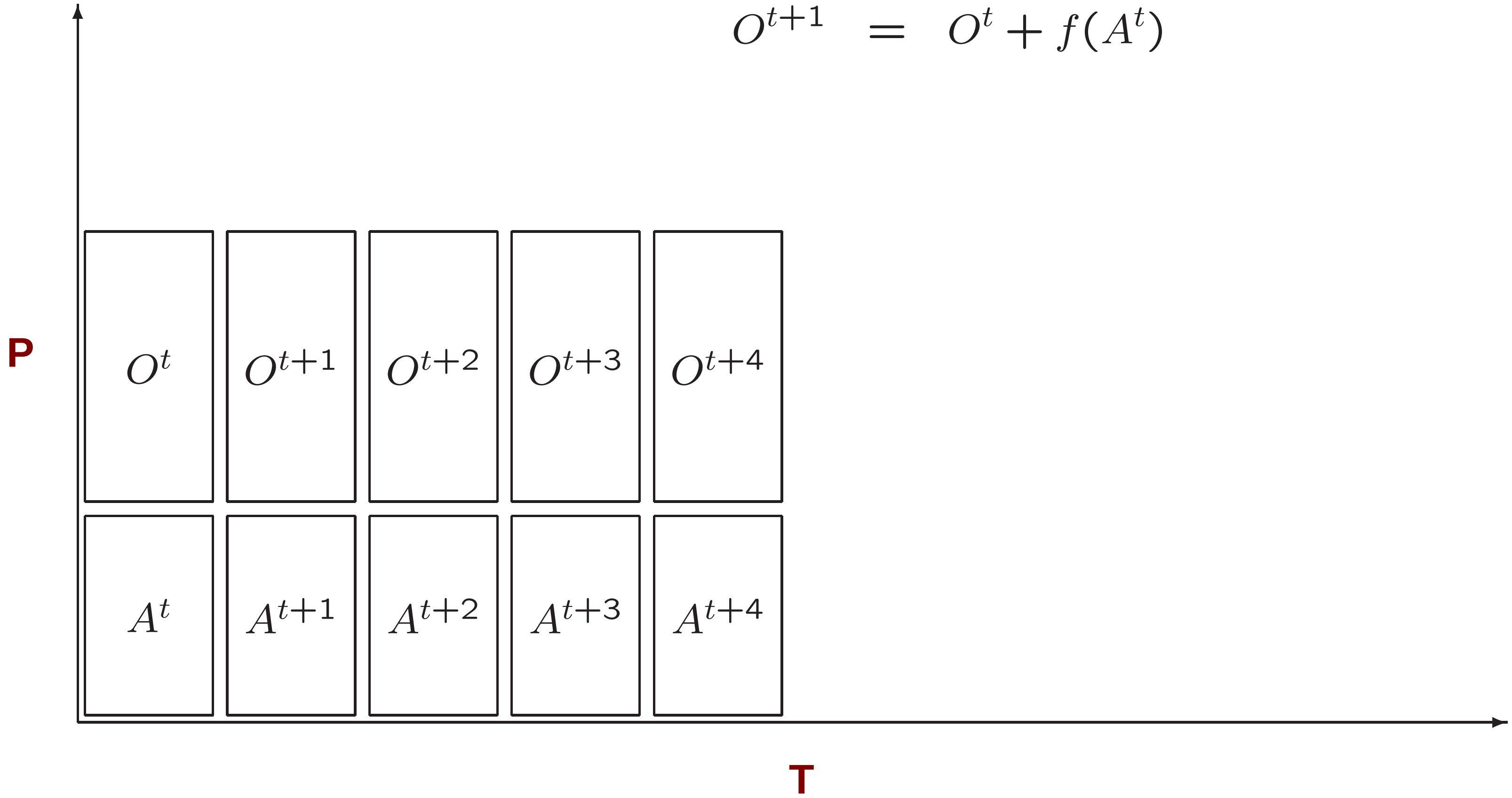
$$A^{t+1} = A^t + f(O^t) \quad (1)$$
$$O^{t+1} = O^t + f(A^{t+1}) \quad (2)$$



Concurrent coupling

This uses a forward-only timestep for coupling. While formally this is unconditionally unstable, the system is strongly damped. Answers vary with respect to serial coupling, as the ocean is now forced by atmospheric state from Δt ago.

$$A^{t+1} = A^t + f(O^t) \tag{3}$$
$$O^{t+1} = O^t + f(A^t) \tag{4}$$



In terms of model code...

```
do nc = 1, num_cpld_calls
   call generate_sfc_xgrid()
   call flux_ocean_to_ice()
   if( concurrent_coupling )call flux_ice_to_ocean()
   if( atmos_pe )then
       call update_ice_model_slow_up()
       call update_atmos... !fast loop
       call update_land_model_slow()
       call flux_land_to_ice()
       call update_ice_model_slow_dn()
   endif
   if( serial_coupling )call flux_ice_to_ocean()
   if( ocean_pe )call update_ocean_model()
enddo
```

- The pelists are set up in the coupler layer, and subsequently all the **`mpp`** calls automatically operate within their pelists, with no changes to the model code.

- Within the atmos pelist, we can further declare land and ice as concurrent if needed. Not currently implemented, since $T_{\text{ice}} \gg T_{\text{land}}$.

FMS component models

- Atmosphere:
  - BGRID: hydrostatic finite difference model on a staggered Arakawa B grid and hybrid σ/P vertical coordinate (Wyman);
  - SPECTRAL: hydrostatic spectral transform model also with the hybrid σ/P vertical coordinate (Held, Phillipps);
  - FV: hydrostatic primitive equations using finite-volume dynamics (S.J Lin);
  - ZETAC: non-hydrostatic C-grid regional/global model on terrain-following coordinates (Garner, Orlanski, Kerr);
  - Spectral shallow water, 2D energy balance, data model, etc.
- Land:
  - Land Dynamics model (LaD) 5 temperature layers, 11 soil/vegetation types, stomatal resistance, bucket hydrology, river routing (Milly);
  - SHE: soil, hydrology and ecology model, interactive biospheric carbon, ecosystem dynamics with dynamic tiling (Malyshev, Shevliakova);

FMS component models

- Ocean:
  - MOM: primitive equation ocean climate model with generalized horizontal coordinates and vertical z-coordinate, full suite of physics options, compatible with state-of-art adjoint compiler (Pacanowski, Griffies, Rosati, Harrison);
  - HIM: isopycnal coordinate ocean model (Hallberg, Cooke);
- Ice: Sea Ice Simulator (SIS) full sea ice dynamics with elastic-plastic-viscous rheology, N-category ice thickness, 3-layer vertical thermodynamics (Winton);

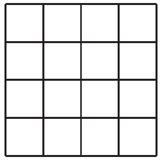
Fitting into FMS

To incorporate your own ocean model (say) into FMS, you have to provide a few key routines (**`ocean_model_init, update_ocean_model`**) and encapsulate your ocean boundary state into **`ocean_boundary_type`**.

It helps to use the FMS infrastructure but not essential.

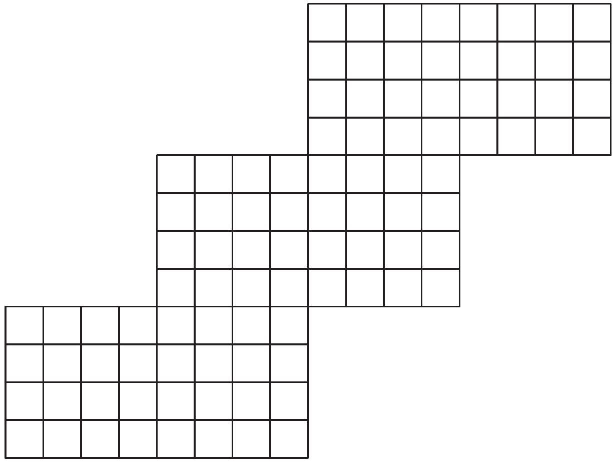
CM2.1 was created from CM2 by importing the new FV core into FMS, and then switching atmospheric cores in the coupled system.

What is a grid mosaic?

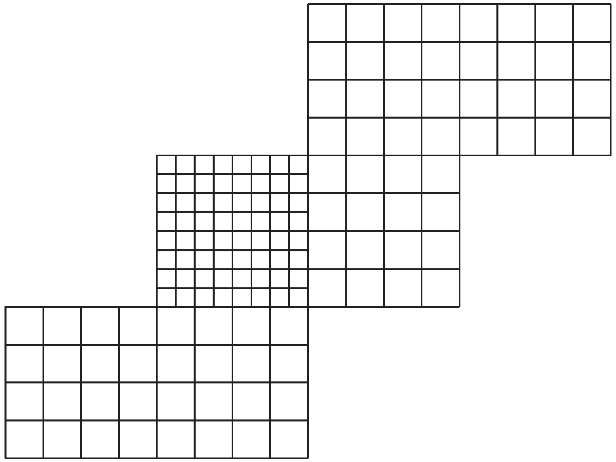


On the left is a basic 4×4 ***tile***; on the right are examples of grids composed of a mosaic of such tiles. The first is a ***continuous grid***, below is a ***refined grid***.

Most current software only supports what we call ***grid tiles*** here. The **`grid 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.

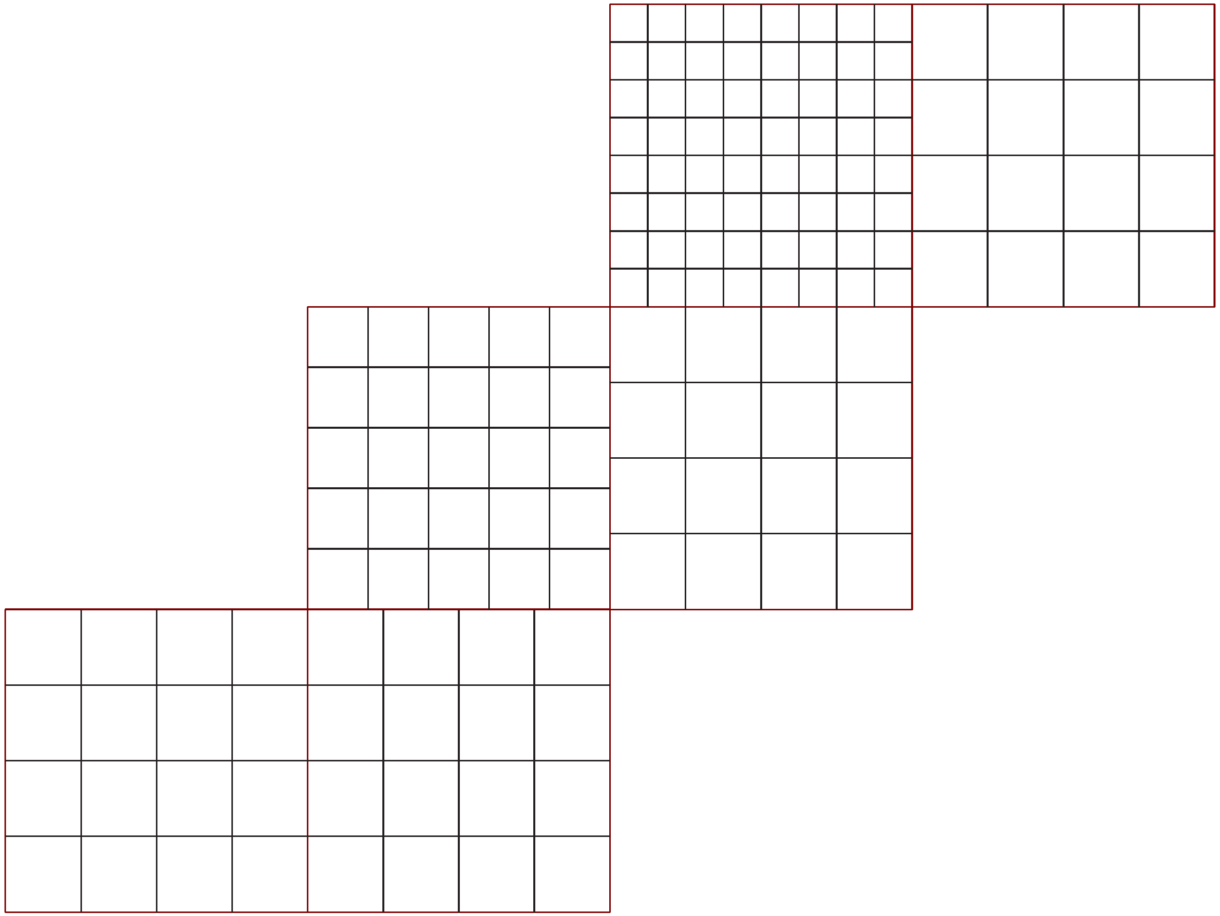


Regular grid mosaic.



Refined grid mosaic.

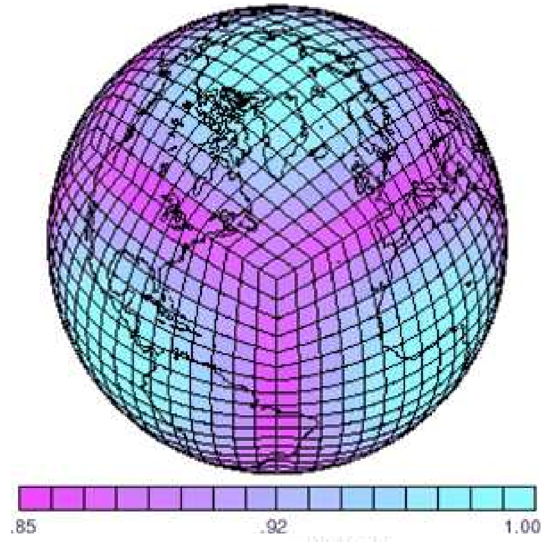
“True” refinement

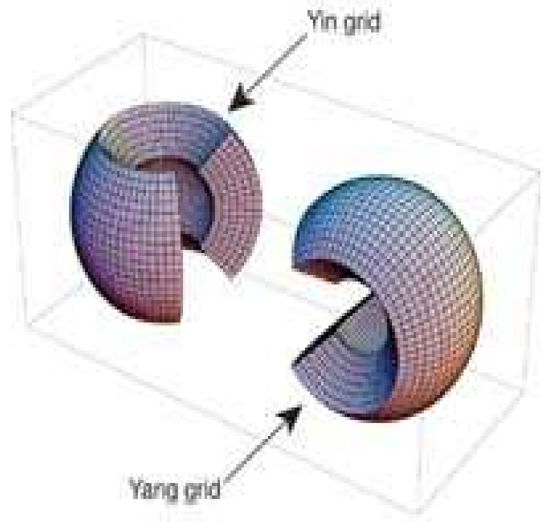


The tile at **`(8,8)`** shows ***true*** refinement; the tile at **`(4,4)`** does not.

Boundaries and contact regions

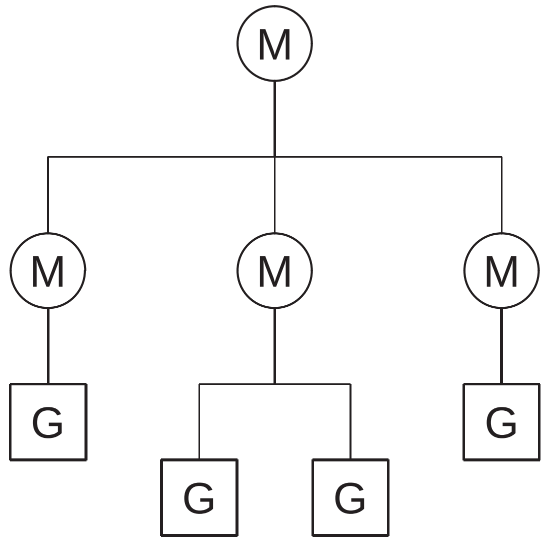
Aside from the grid information in the grid tiles, the grid mosaic additionally specifies connections between pairs of tiles in the form of ***contact regions*** between ***pairs*** of grid tiles.
Contact regions can be ***boundaries***, topologically of one dimension less than the grid tiles (i.e, planes between volumes, or lines between planes), or ***overlaps***, topologically equal in dimension to the grid tile. In the cubed-sphere example the contact regions between grid tiles are 1D boundaries: other grids may contain tiles that overlap. In the example of the ***yin-yang*** grid (Kageyama et al 2004) the grid mosaic contains two grid tiles that are each lon-lat grids, with an overlap. The overlap is also specified in terms of a ***contact region*** between pairs of grid tiles. Issues relating to boundaries are described below. Overlaps are described in terms of an exchange grid, more on which below.





Grid mosaic definition

A ***grid mosaic*** is constructed recursively by referring to child mosaics, with the tree terminating in leaves defined by ***grid tiles***.
(There is a very useful analogy to be made between mosaic hierarchies and model component hierarchies).



It is not necessarily possible to deduce contact regions by geospatial mapping: there can be applications where geographically collocated regions do ***not*** exchange data, and also where there is implicit contact between non-collocated regions.

Applications of grid mosaics

The grid mosaic is a powerful abstraction making possible an entire panoply of applications. These include:

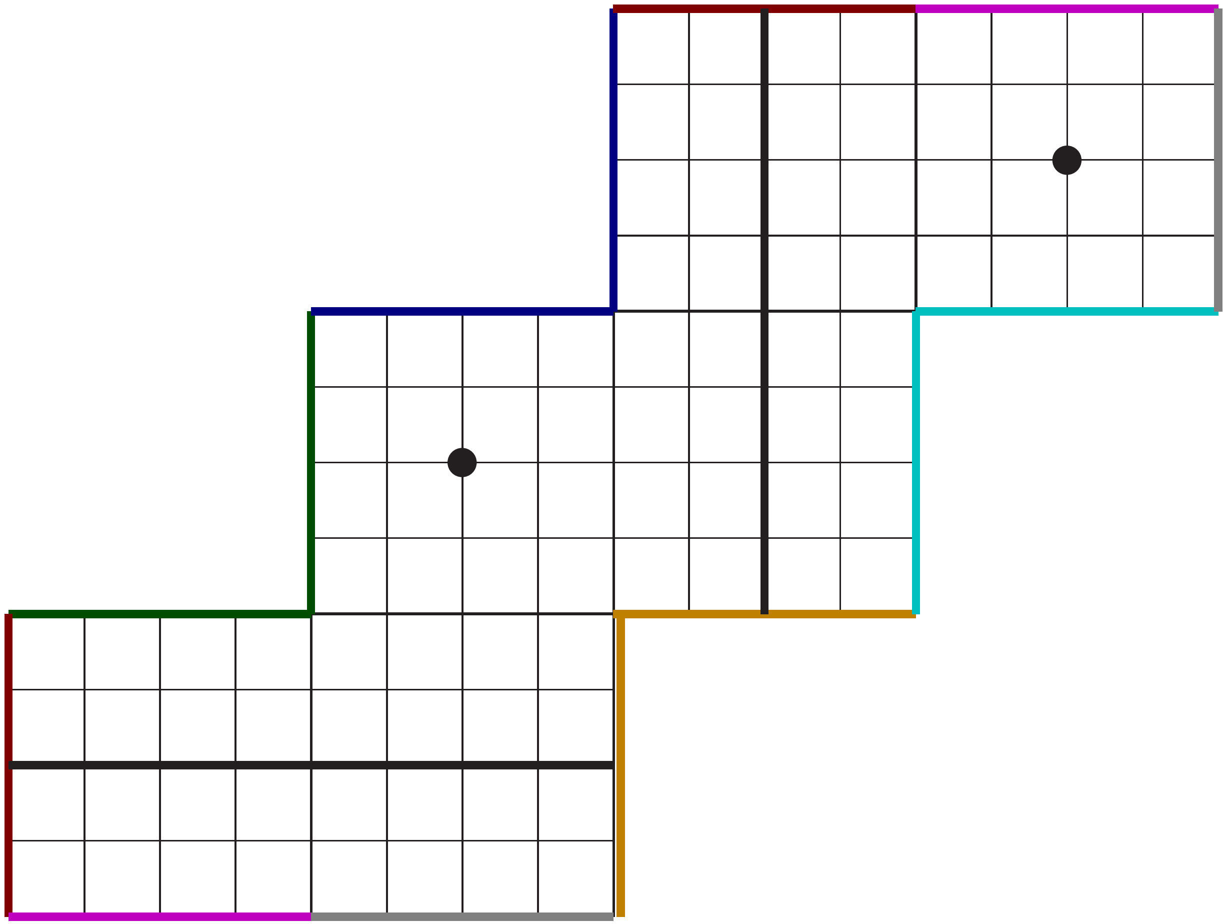
- the use of overset grids such as the yin-yang grid;
- the representation of nested grids (e.g Kurihara et al 1990);
- the representation of reduced grids (e.g Rasch 1994). Currently these typically use full arrays and a specification of the “ragged edge”. A reduced grid can instead be written as a grid mosaic where each reduction appears as a separate grid tile.
- An entire coupled model application or dataset can be constructed as a hierarchical mosaic. Grid mosaics representing atmosphere, land, ocean components and so on, as well as contact regions between them, all can be represented using this abstraction. This approach is already in use at many modeling centres including GFDL, though not formalized.
- Finally, grid mosaics can be used to overcome performance bottlenecks associated with parallel I/O and very large files. Representing the model grid by a mosaic permits one to save data to multiple files, and the step of ***aggregation*** is deferred. This approach is already used at GFDL to perform distributed I/O from a parallel application, where I/O aggregation is deferred and performed on a separate I/O server sharing a filesystem with the compute server.

Boundary spec for a cubed sphere

Boundaries for LRG tiles are specified in terms of an ***anchor point*** and an ***orientation***.

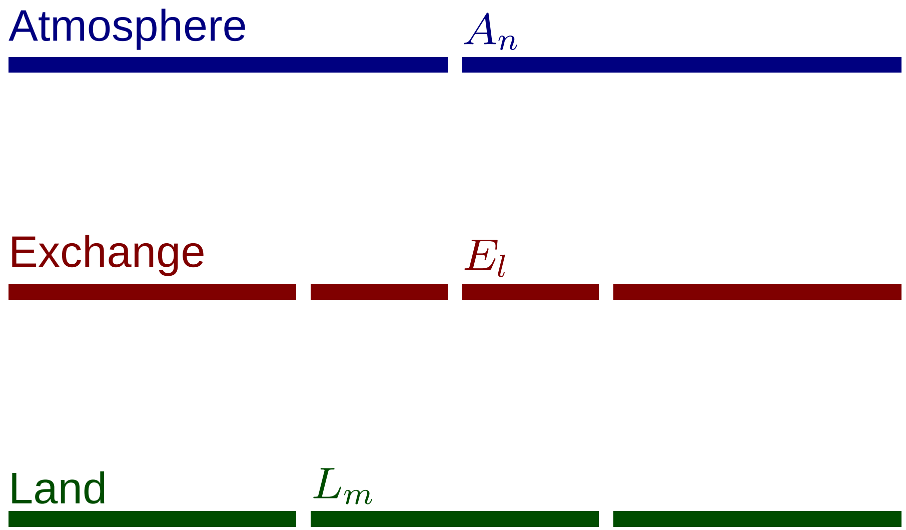
An anchor point is a boundary point that is common to the two grid tiles in contact. When possible, it is specified as integers giving index space locations of the anchor point on the two grid tiles. When there is no common grid point, the anchor point is specified in terms of floating point numbers giving a geographic location.

The ***orientation*** of the boundary specifies the index space direction of the running boundary on each grid tile: the point just to the "west" of **`(5,6)`** is in fact **`(3,4)`**



Overlap contact regions: the 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.



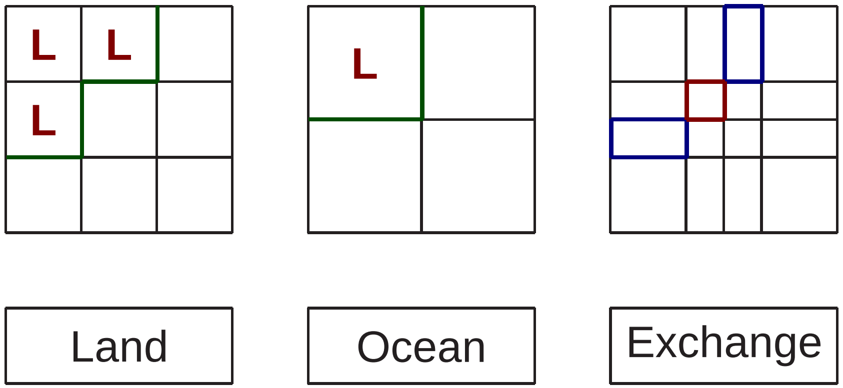
- Exchange: interpolate from source grid using one set of fractional areas; then average onto the target grid using the other set of fractional areas.

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

Overlap contact regions: 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.

Summary: approaches to grid coupling

- The generalization of ***grids*** to ***mosaics*** is a key development in GFDL's approach to nested grid models (Balaji, Adcroft, Zhi Liang).

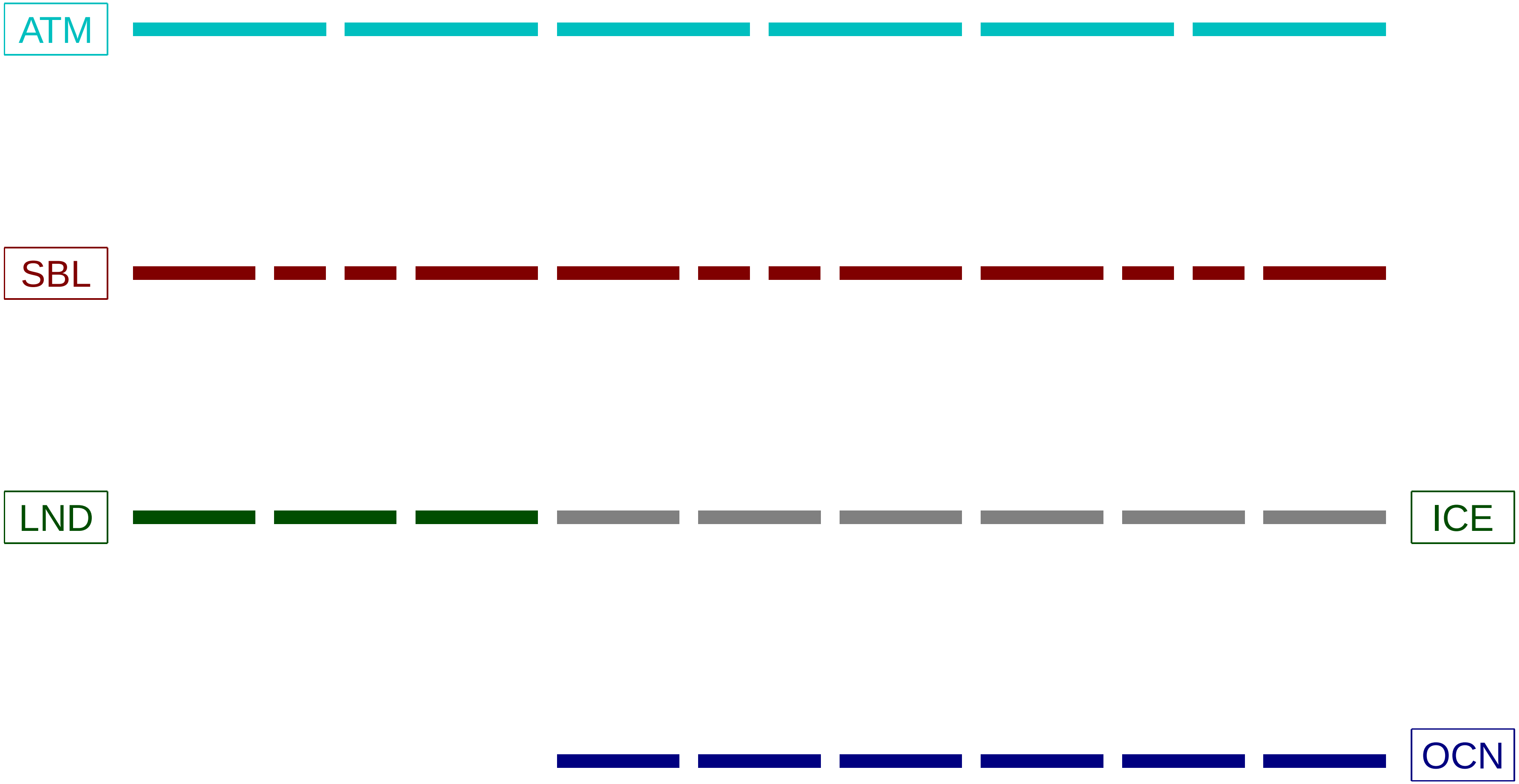
- Mosaics are to be used in leading-edge model discretizations such as the ***cube-sphere*** (S-J Lin, Michael Herzog).

- ***Simple*** refinement will be used for two-way nesting of models within themselves, including adaptive grids (Adcroft, Hallberg, Herzog).

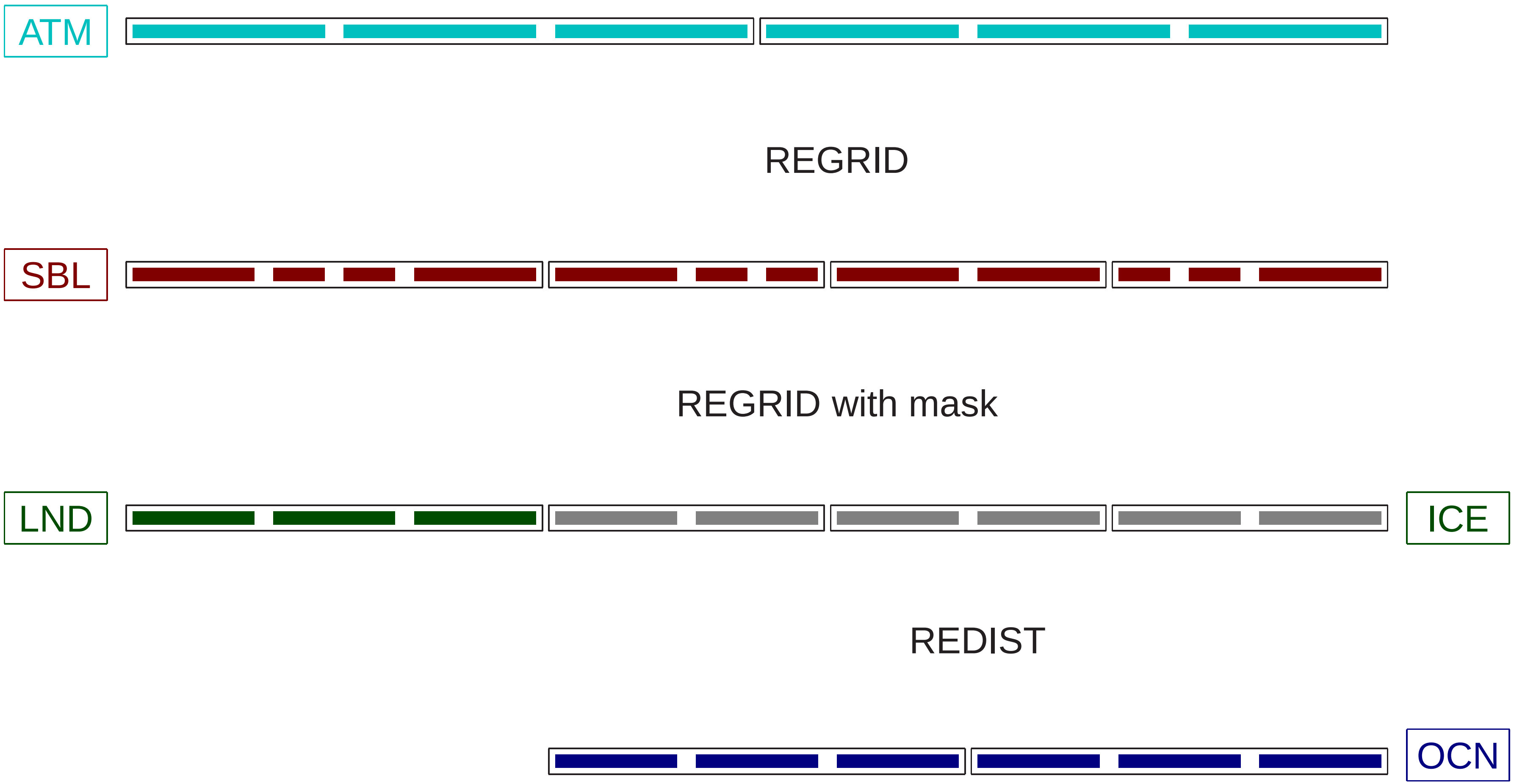
- Interpolation between arbitrary grids for model ***coupling*** and model ***chaining*** is accomplished using ***exchange grids*** (Anderson, Balaji, Winton).

- A ***standard*** for describing grids and mosaics is in draft stage and is expected to be adopted by the CF community (Balaji). This is expected to lead to the development of standardized regridding methods and tools to aid in the process of model nesting, coupling and chaining.

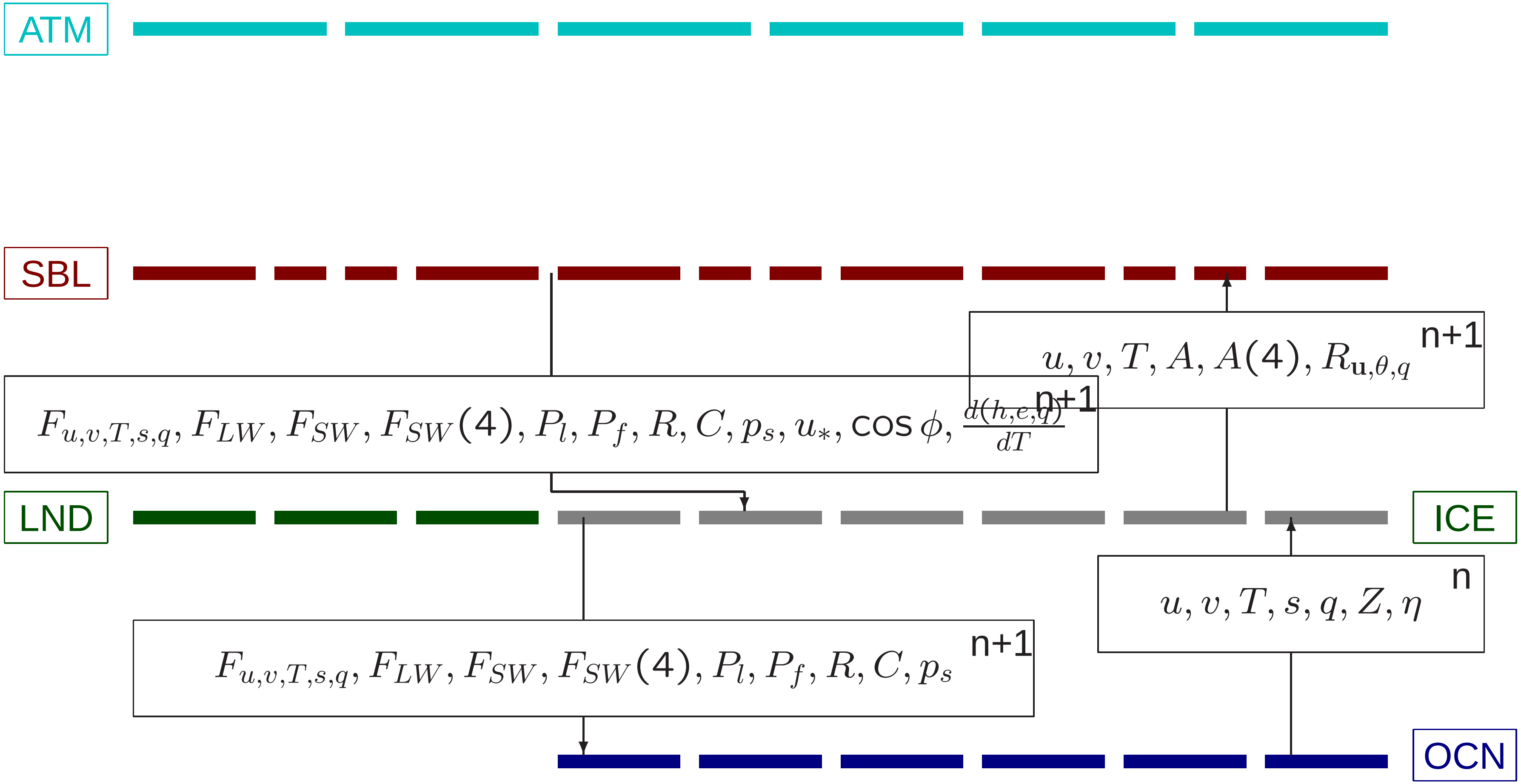
FMS coupled architecture



FMS coupled architecture: parallelism



FMS coupled architecture: ice-ocean coupling



MAPL conventions

General:

- Export states may not be modified by wrappers; import states can.
- Vectors are passed on an A-grid, aligned to geographic coordinates (proposal: metadata to stipulate geographic- or grid-aligned).
- Wrapper layer defines its own precisions (R8 v R4) and does not adopt precisions of its children.
- Vertical flux sign conventions must be part of metadata.
- 2D/3D arrays have no halos (proposal: supplied with halos); wrapper layer adopts 1-based indexing.
- The `ESMF_Clock` that a component receives must not be modified.

Physical:

- Orad will export the surface flux and a 3D opacity field.
- O will export thicknesses on A-grid (i.e thickness of a T-cell).