

Improving climate model coupling through complete mesh representation

Robert Jacob, Iulian Grindeanu, Vijay Mahadevan, Jason Sarich

July 12, 2018

3rd Workshop on Physics Dynamics Coupling

Support: U.S. DOE BER “Climate Model Development and Validation” project

The other coupled problems...

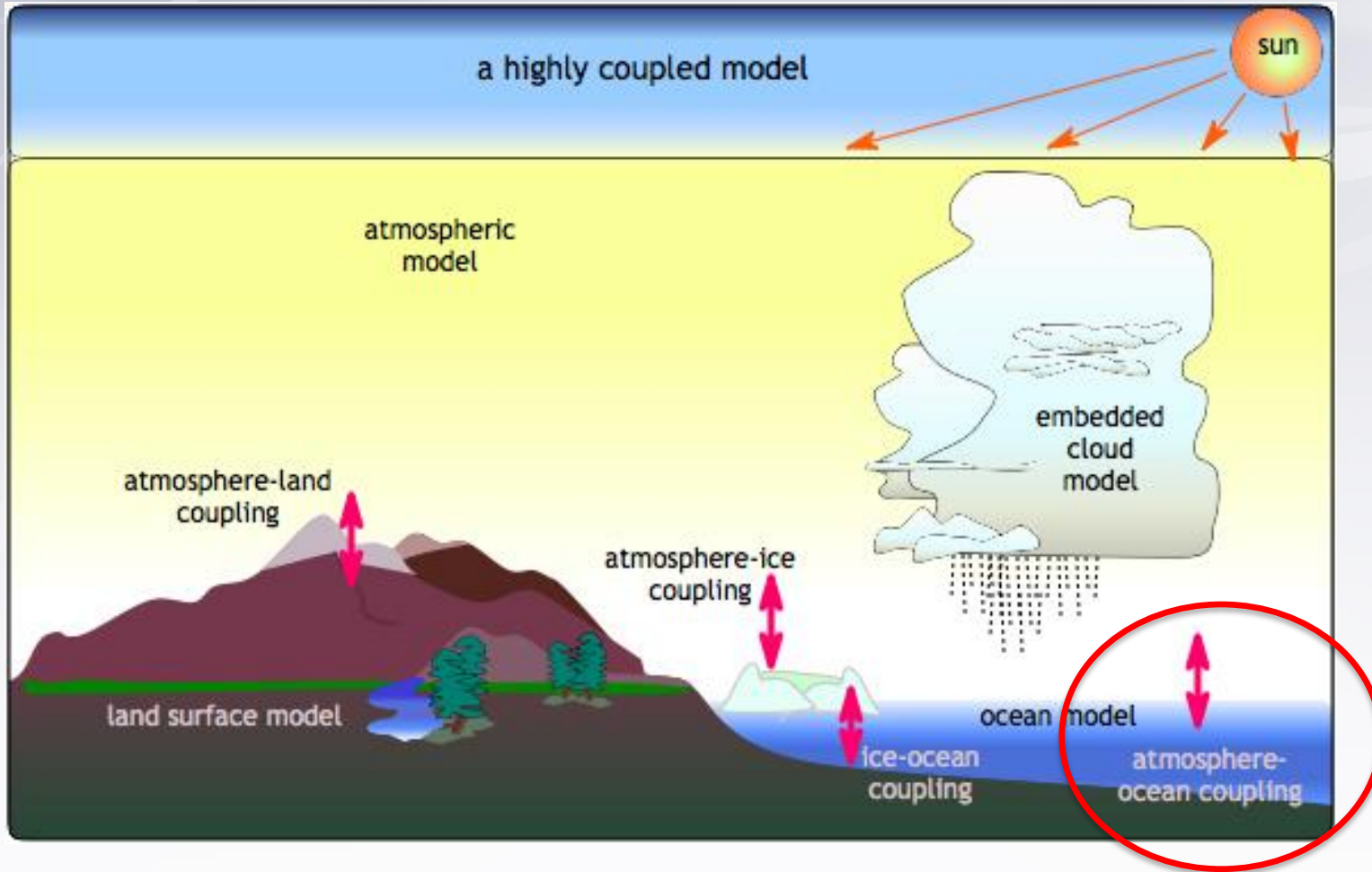
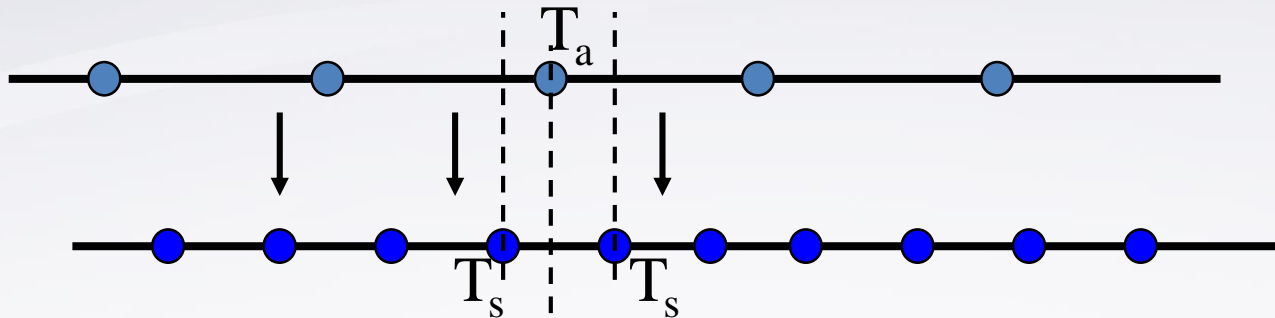


Figure from Center for Multiscale Modeling of Atmospheric Processes

Ocean-atmosphere exchange fluxes of heat, radiation, momentum, water

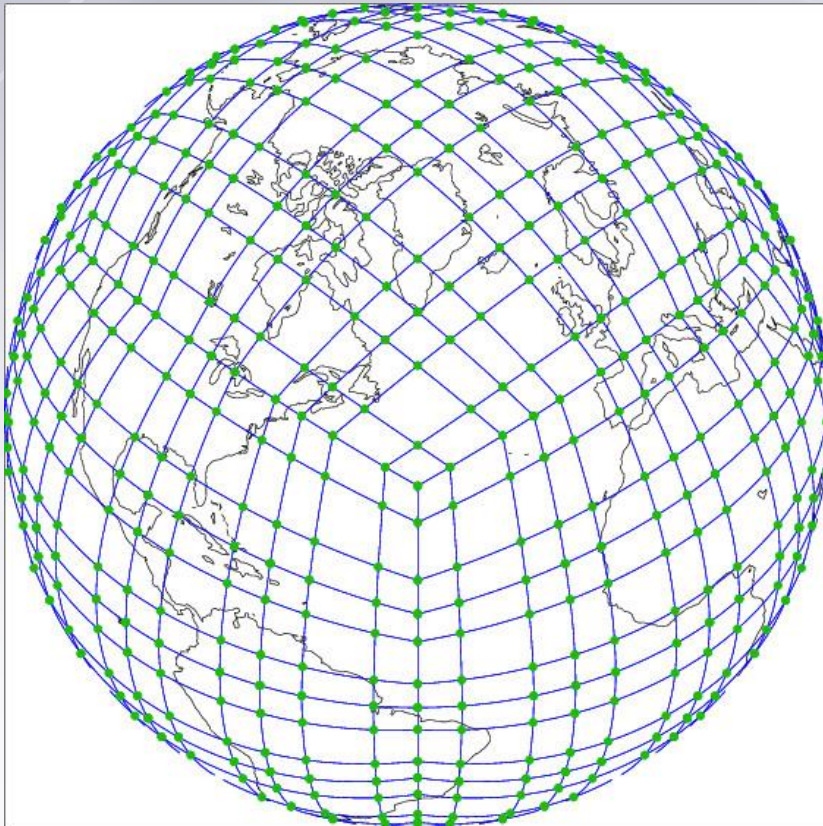
- For interface-fluxes (heat, momentum) choice in DOE E3SM is to interpolate atmosphere to finer ocean grid, then calculate fluxes, then (for atm) interpolate back.

$$\text{SensibleHeat} = A (T_a - T_s)$$

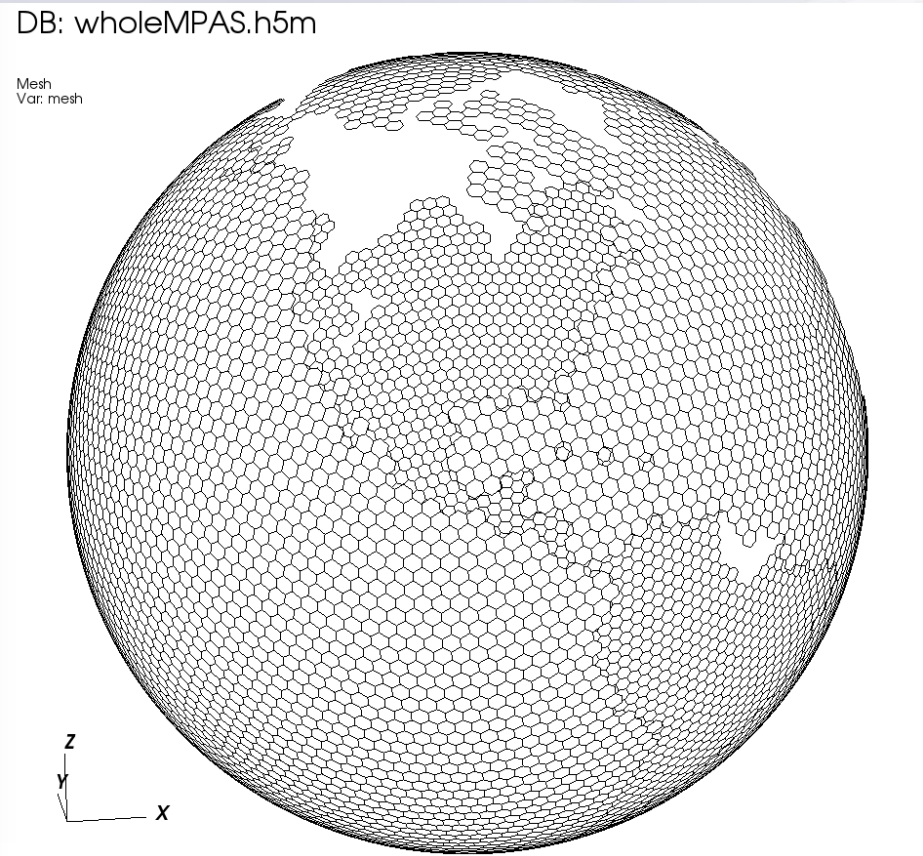


- Atmosphere-computed fluxes (radiation, rain) interpolated directly to ocean
- **Flux interpolation must conserve globally**

Atmosphere and ocean grids to couple in E3SM



HOMME SE (with GLL points)



MPAS FV

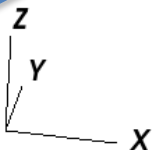
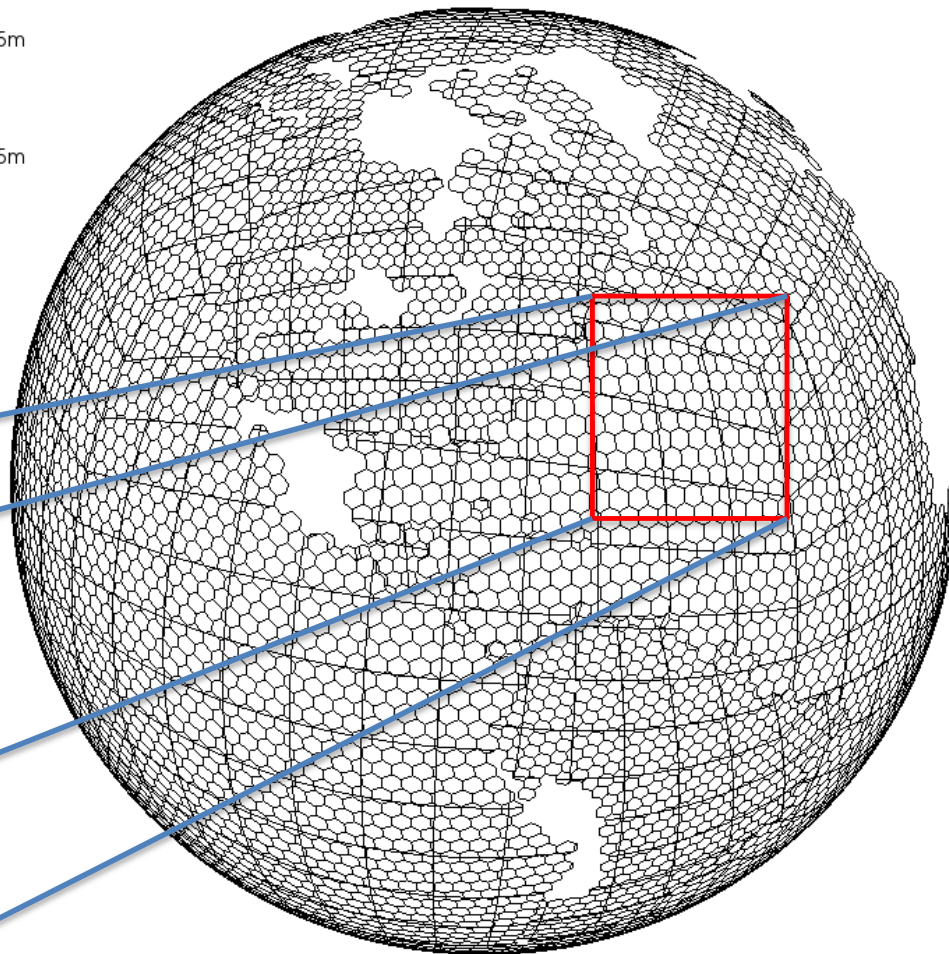
The Intersection (overlap) mesh

Mesh
DB: intx00.h5m

Var: mesh

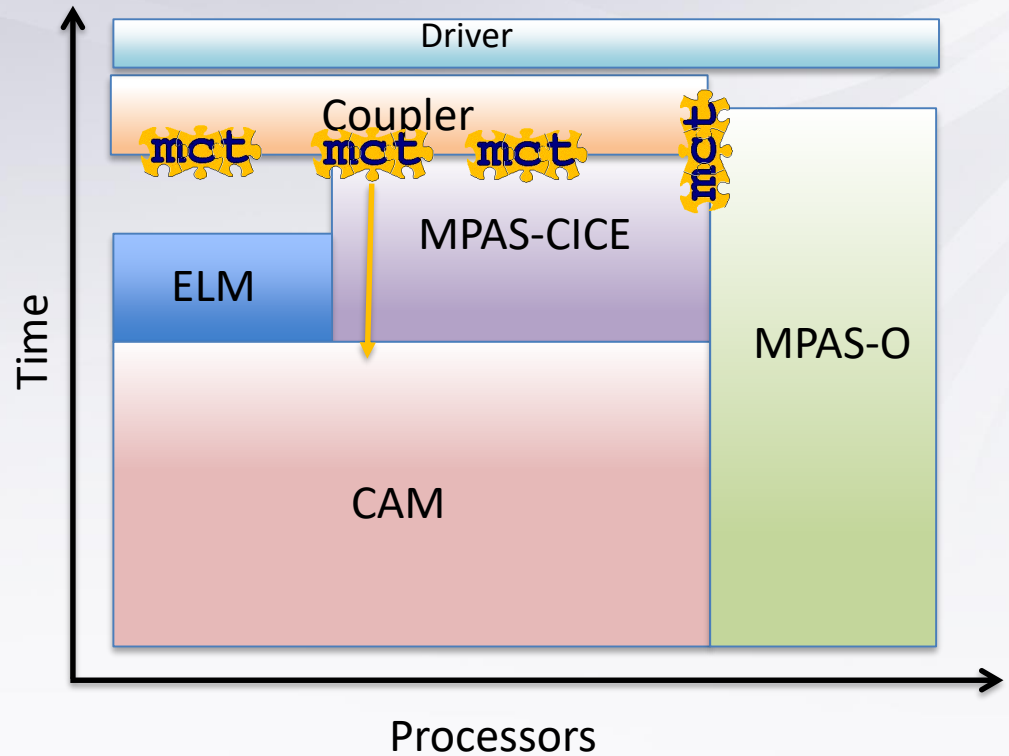
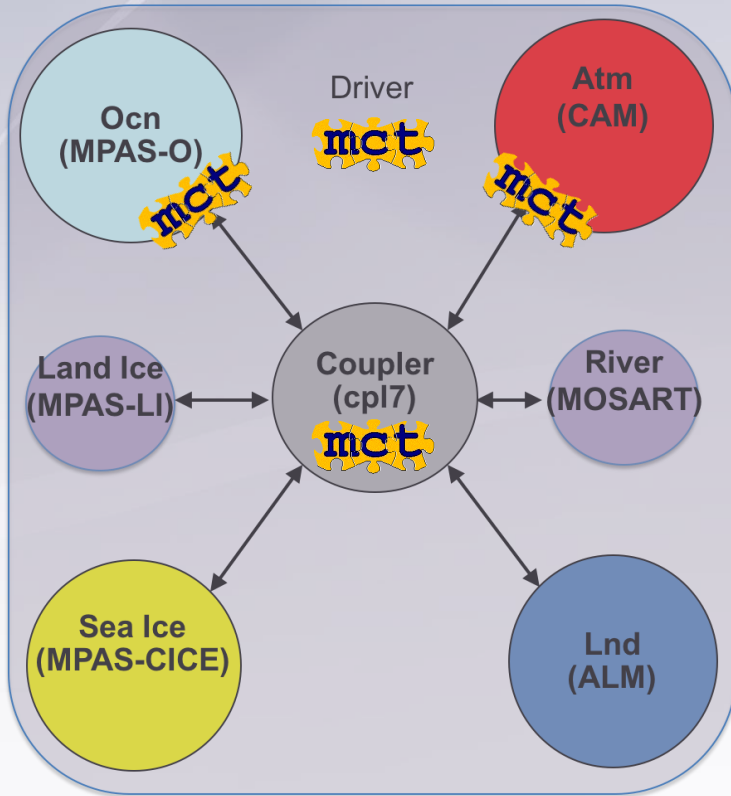
Mesh
DB: intx01.h5m

Var: mesh



Climate Model Coupling software

MCT datatypes used in coupler, driver and model “caps”.



Model Coupling Toolkit



- A library of Fortran90 datatypes and functions for building parallel coupled models.
 - 2-sided parallel MxN send/recvs for moving data (with MPI)
 - Support for online parallel interpolation using **offline-calculated** weights.
 - Model registry, decomposition descriptors, distributed data type, communication tables.
 - Functions for time accumulation, spatial averaging and merging.
 - Programmer must write the “cap layer” or other interface.
 - No prescription on overall model architecture
 - Current release: MCT 2.10.0 (April, 2018)
 - First version in production: MCT 2.1 (CCSM3, June, 2004)
- Used in OASIS3-MCT (many models) and
CIME cpl7 (E3SM1.0, CESM2.0)

Current coupling workflow: offline/online

offline phase

Generate grid

MPAS: grid-gen, HOMME: internal or GenerateCSMesh

Convert to SCRIP
format

(some of the above can output SCRIP)

Feed grids to program
that finds intersection
and calculates weights.

SCRIP or ESMF_Regridder or TempestRemap

Obtain weights file in
SCRIP format

Ex: map_ne30np4_to_oEC60to30v3_conserve.161222.nc

Point model config file
to weights file.

cime/config/(\$model)/config_grid.xml

Current coupling workflow: offline/online

online phase: init

Define MCT GSMaps	Separately in each model. Stores model-preferred decomposition
Communicate grids to coupler	Only the total number of dofs.
Define coupler GSMaps	Trivial decomposition
Read in weight files. Initialize MCT SparseMatrix	Read a block and scatter. Each node picks out what it needs.
Initialize MCT Routers/Rearrangers	Figure out and store communication needed to complete a mapping. Between model and coupler. Within coupler.

Current coupling workflow: offline/online

online phase: run

Receive new data
from source model

Stored in an MCT AttributeVector

Perform local sparse-
matrix-vector multiply

Use MCT sMatAvMult method

Send remapped data
to destination model

Stored in an MCT AttributeVector

Two problems with current system

1. The atmosphere->ocean interpolation does not take advantage of the high-order SE grid
2. The offline-online system is fragile and poorly documented

Current coupling workflow: offline/online

offline phase

Generate grid

MPAS: grid-gen, HOMME: internal or GenerateCSMesh

Generate dual of SE grid!

Convert to SCRIP
format

(some of the above can output SCRIP)

Feed grids to program
that finds intersection
and calculates weights.

SCRIP or ESMF_Regridder or TempestRemap

Obtain weights file in
SCRIP format

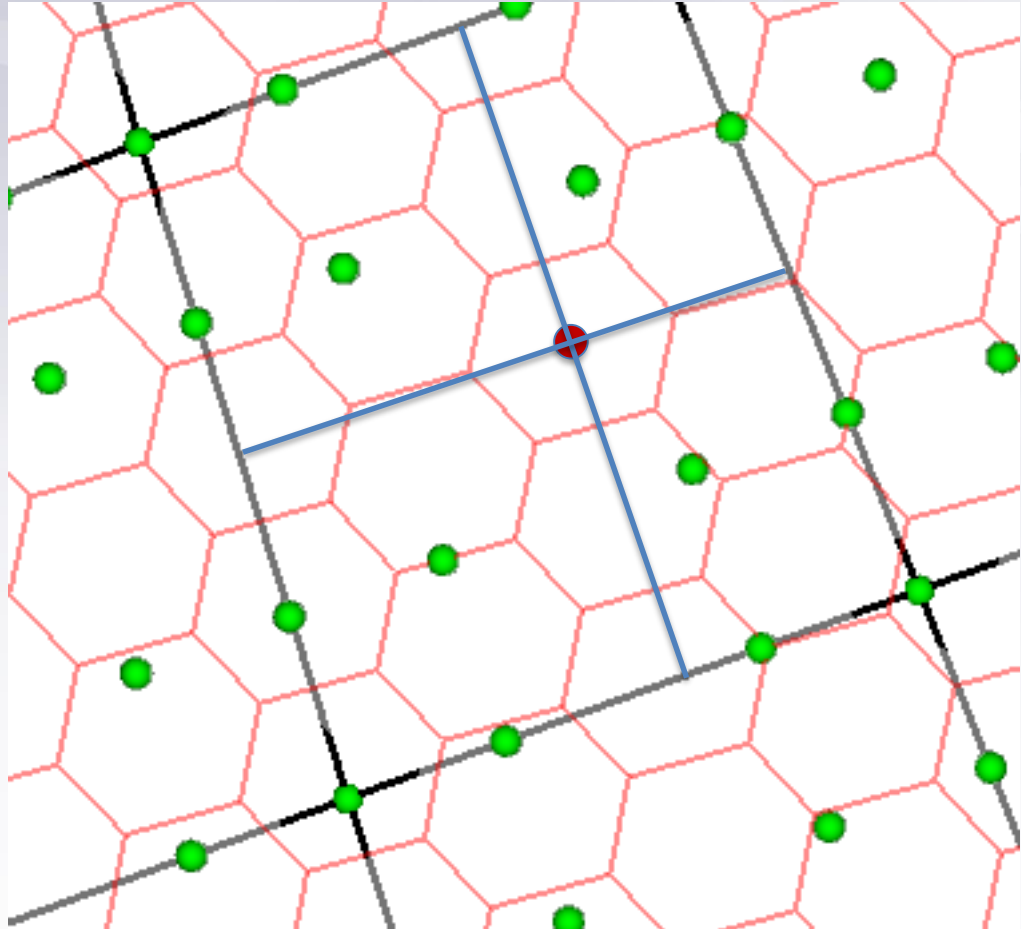
Ex: map_ne30np4_to_oEC60to30v3_conserve.161222.nc

Point model config file
to weights file.

cime/config/(\$model)/config_grid.xml

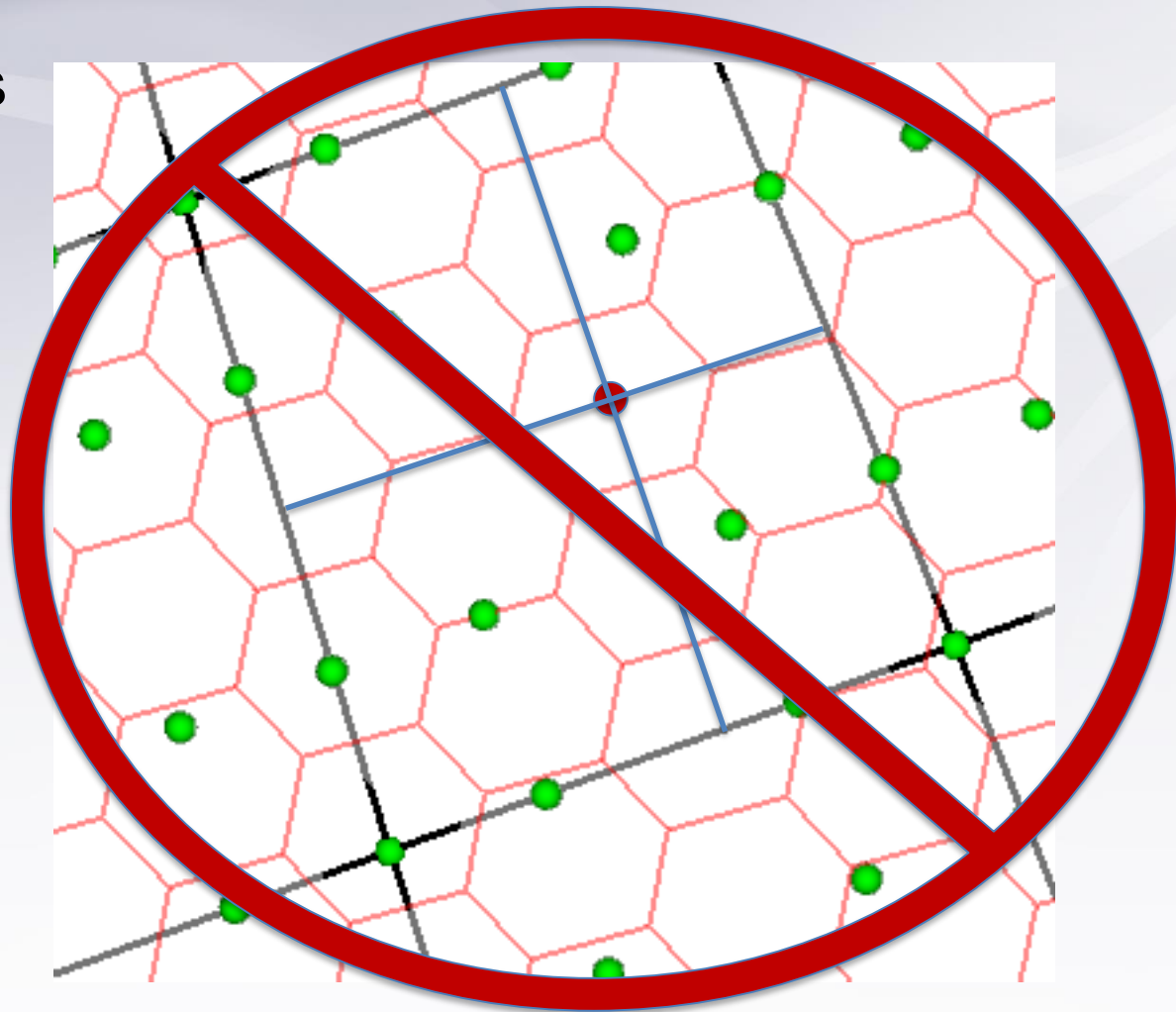
How SE-> FV can take advantage of SE

Use the polynomials to find the value at the point in the intersected ocean mesh (which is Finite Volume with all values at cell centers)



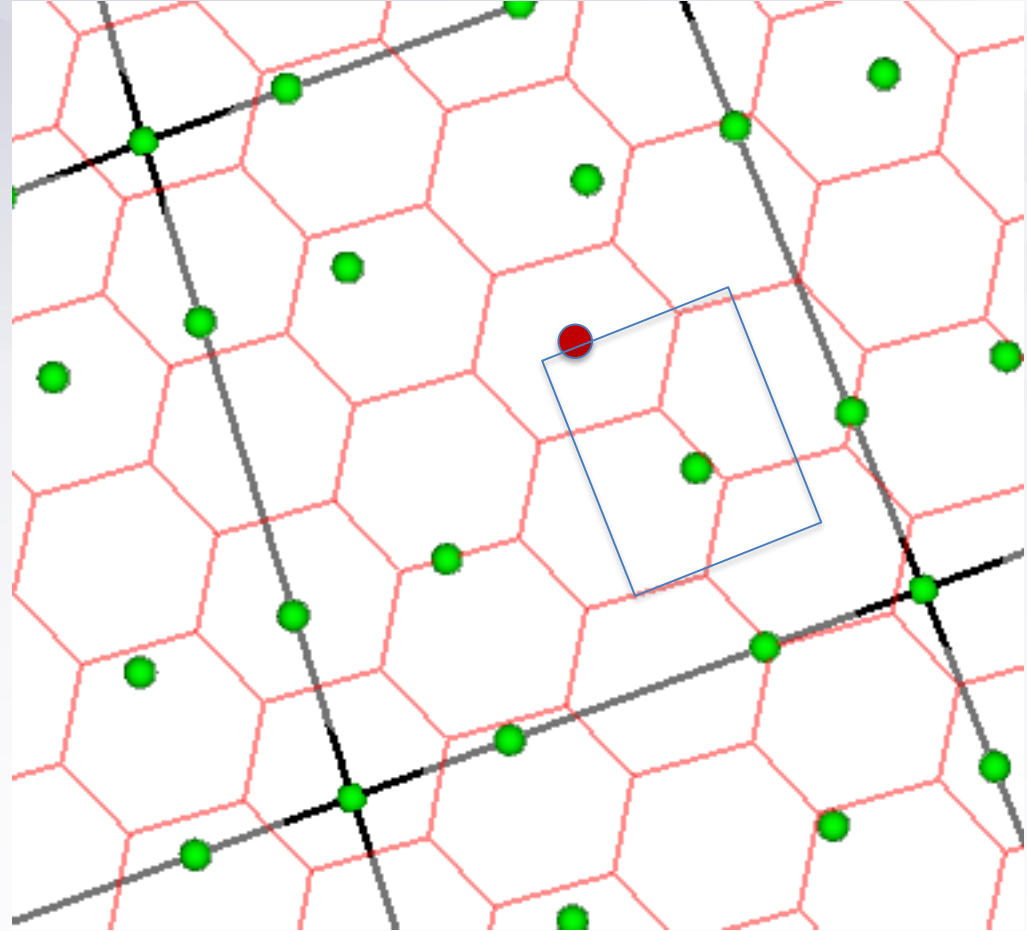
But.....

The two main programs for generating interpolation weights, SCRIP and ESMF_Regridder can only do Finite Volume to Finite Volume (or FD)



For an FV algorithm, have to construct a dual of the SE grid

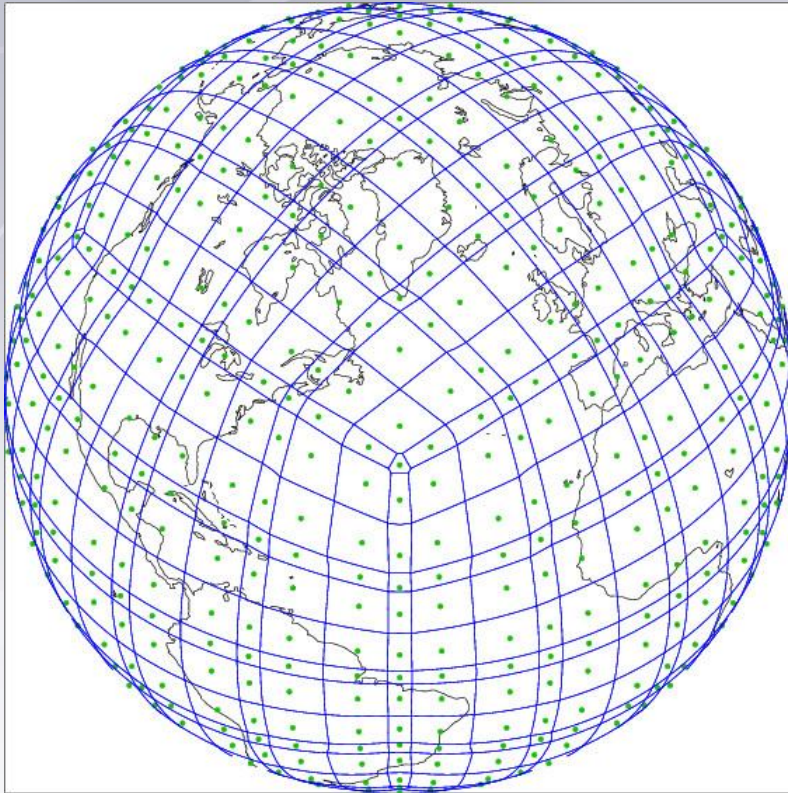
- Treat each GLL point as the center of a polygon.



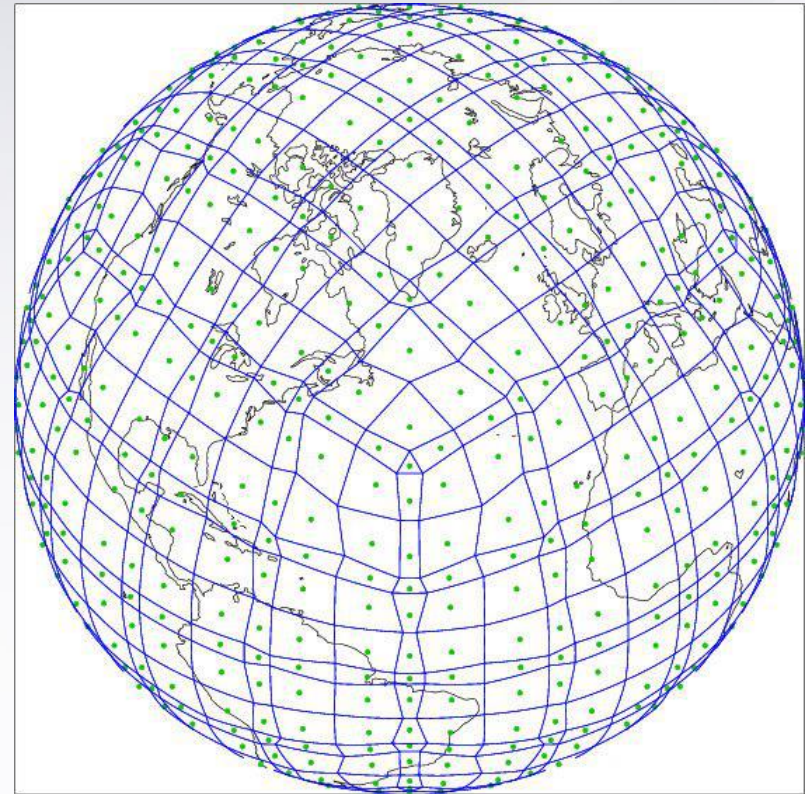
Constructing an SE dual grid is not trivial

- Draw a polygon around each GLL node
- For conservative interpolation, area of polygon around GLL point must match weight of GLL node
- Must use an iterative process, similar to spring dynamics, tweaking the polygons until the areas are correct.

Types of dual grids (“chevron”, “natural”)



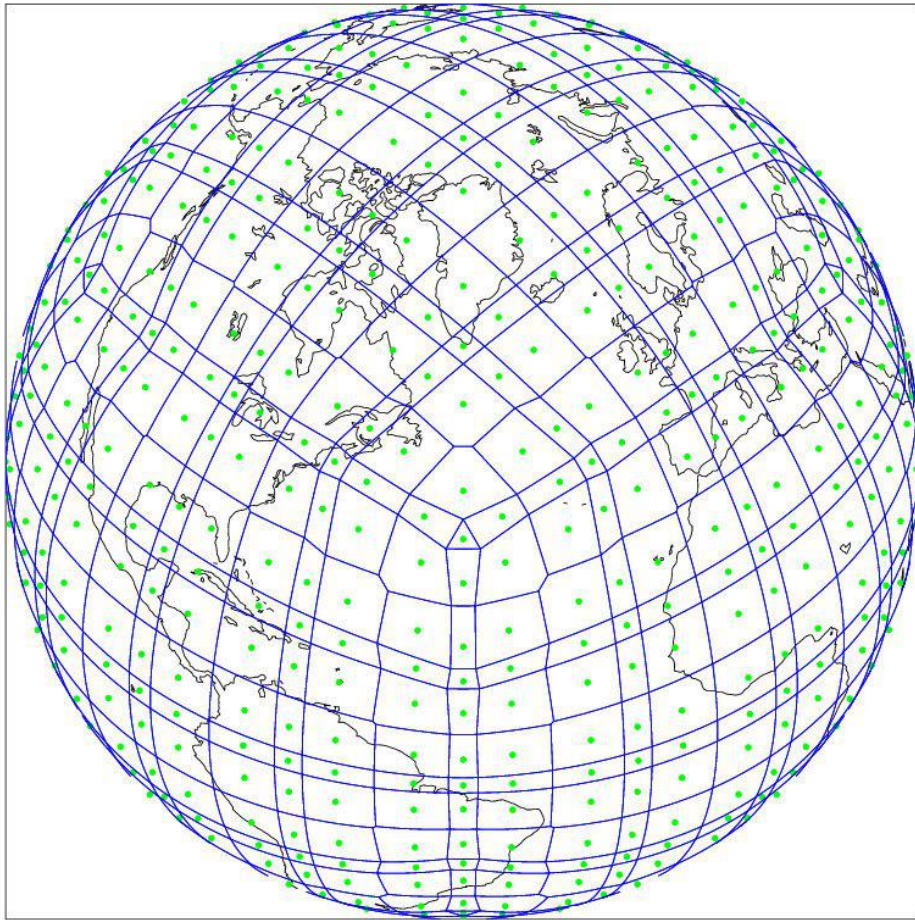
Allow pentagons and hexagons.
Converges fast. But some hexagons become non-convex (chevrons). SCRIP can't do non-convex grids.



Figures by Mark Taylor

Only use centers of the GLL subcell grids to make polygons. All convex but many irregular shapes (after tweaking for area)

Types of dual grids (“pentagons”)

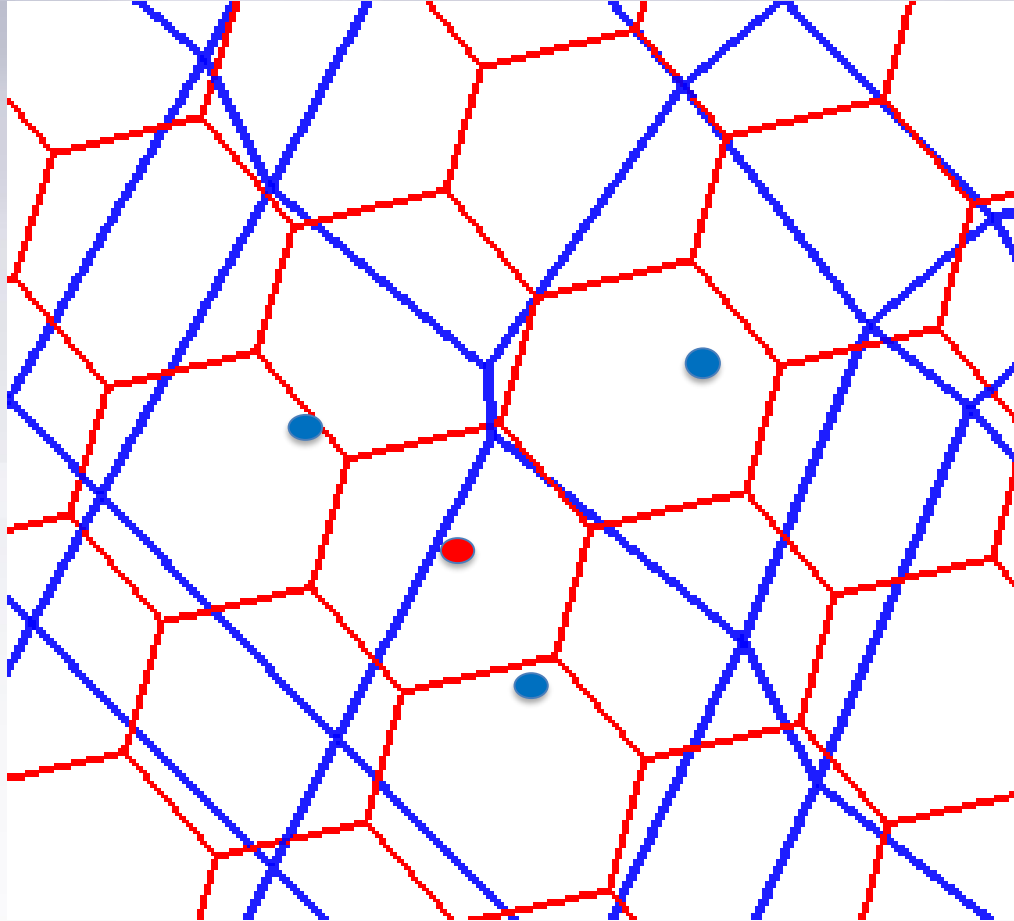


Figures by Mark Taylor

To remove the irregular shapes, selectively allow (convex) pentagons.

This is the dual grid currently used to calculate weights in atmosphere-ocean interpolation for E3SM/CESM.

Intersection of SE-dual and MPAS



A solution to first problem: TempestRemap

- TempestRemap provides offline generation of conservative, consistent and/or monotone linear maps between meshes.
- Supports both source and target grids which are either finite-volume based **or finite element**.
- <https://github.com/ClimateGlobalChange/tempestremap>
- Paul A. Ullrich and Mark A. Taylor, 2015: Arbitrary-Order Conservative and Consistent Remapping and a Theory of Linear Maps: Part 1. *Mon. Wea. Rev.*, 143, 2419–2440, doi: 10.1175/MWR-D-14-00343.1
- Paul A. Ullrich, Darshi Devendran and Hans Johansen, 2016: Arbitrary-Order Conservative and Consistent Remapping and a Theory of Linear Maps, Part 2. *Mon. Weather Rev.*, 144, 1529-1549, doi: 10.1175/MWR-D-15-0301.1.

The other problem with the current method...

- Total offline workflow poorly documented.
- Easy to lose provenance at each step.
- Need to generate map sets for different resolution combinations.
 - E3SM 1 degree fully coupled case (no ice sheet) has 6 maps.
 - 100s of mapping files in our input data repository
- Bottleneck to trying simulations with different grids.

Possible Solution

- Support online interpolation by replacing MCT with MOAB!

MCT's idea of the mesh (grid)

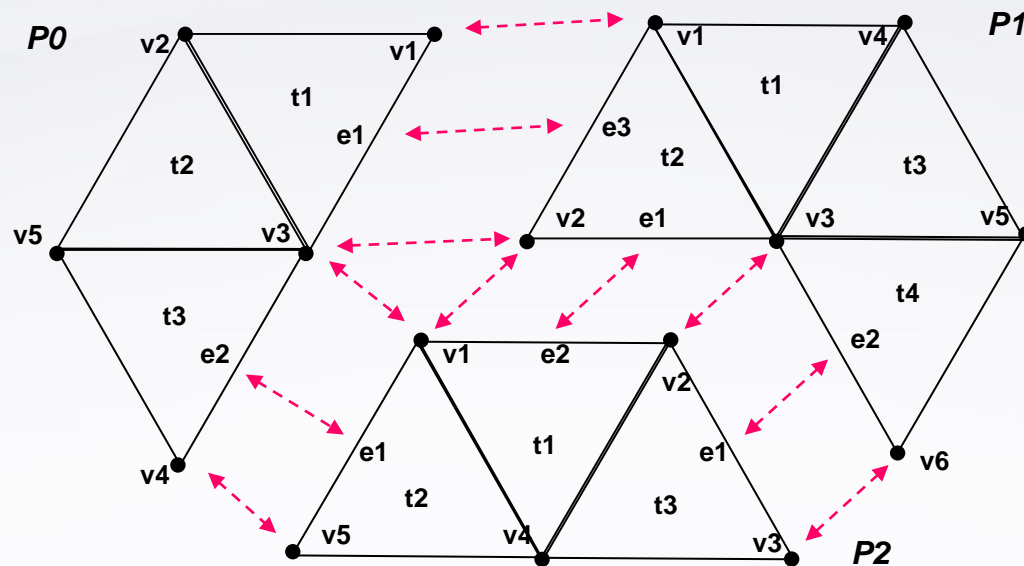
- MCT only knows about the number of grid points or “degrees of freedom”, dof.
 - Each grid must have its points numbered with integers. One unique integer per unique point.
 - Numbering up to user/model.
 - Must be consistent if the same grid appears in more than one model.
 - MCT GlobalSegmentMap records how grid points are assigned to processors.
 - Does not have enough information to find intersection of 2 meshes

Mesh-Oriented datABase (MOAB)

- Library for representing, manipulating structured, unstructured mesh models.
- Supported mesh types: (not just dof numbers)
 - FE zoo (vertices, edges, tri, quad, tet, pyramid, wedge, knife, hex)
 - Polygons/polyhedra
 - Structured mesh
- Optimized for memory usage first, speed second
- Implemented in C++, but uses array-based storage model
 - Avoids C++ object-based allocation/deallocation
 - Allows access in contiguous arrays of data
- Mesh I/O from/to various formats
 - HDF5 (custom), vtk, CCMIO (Star CD/CCM+), Abaqus, CGM, Exodus, NetCDF
- Allows a complete representation of the mesh!

What does a complete representation of the mesh mean?

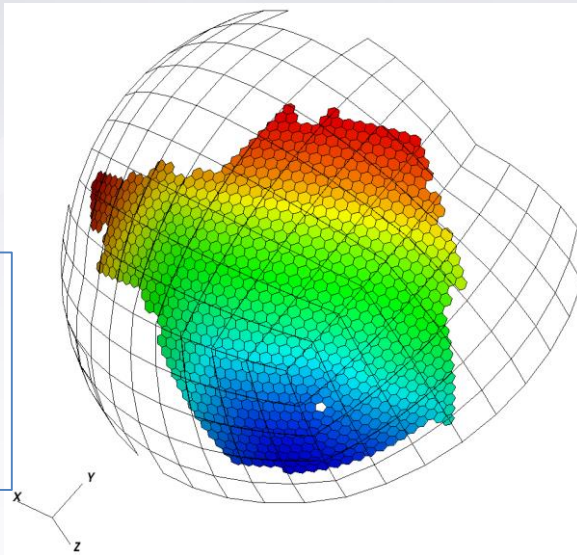
- Not just dofs, know what is an edge, a vertex, a face, what is shared on a node and across nodes.



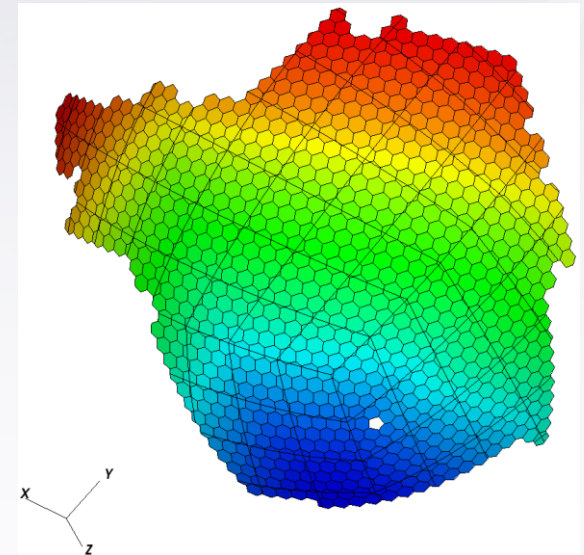
Compute mesh intersection

- Compute the "coverage", and migrate the source mesh to cover the target elements
- MOAB uses an [advancing-front algorithm](#) with fast adjacency queries for intersection

Compute the coverage source (HOMME) mesh to cover the local MPAS elements



Compute the mesh intersection between HOMME and MPAS meshes



MOAB-TempestRemap tool (mbtempest)

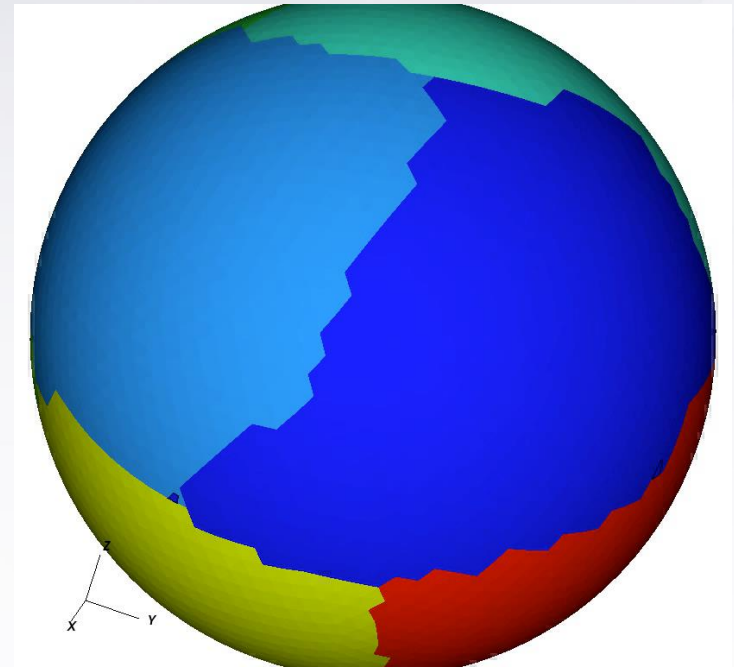
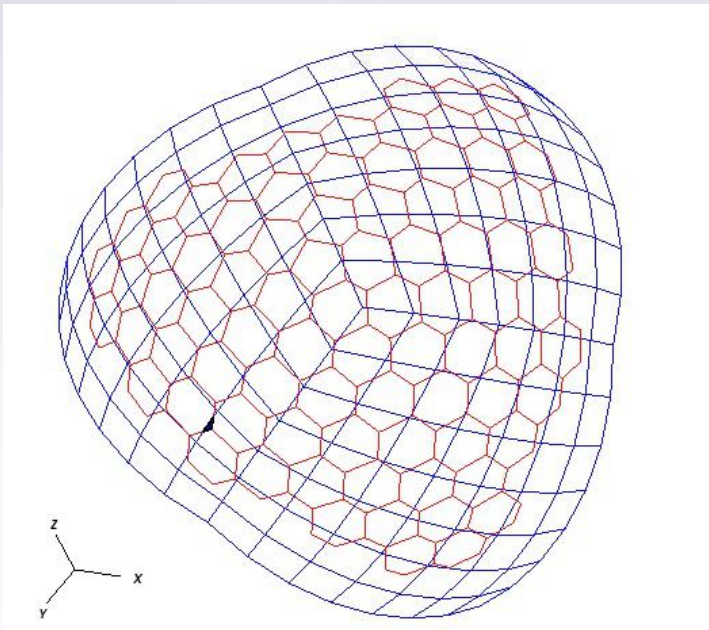
- `mbtempest` is a parallel, offline, conservative remapping tool that is based on MOAB and TempestRemap libraries
- Support conservative projections between FV ↔ FV, DG ↔ CG, SE ↔ FV field discretization combinations in source and destination components
- `mbtempest` supports a scalable workflow to load pre-partitioned input component meshes, compute remapping weights and apply weight matrix onto solution vector
- The development of parallel NetCDF writing of the weight matrix is in progress
- The integration of the online remapper with E3SM is ongoing; Scalability studies being performed on DOE machines Theta and Cori-KNL.
- `mbtempest` dependencies: MPI, HDF5, NetCDF (C++), TempestRemap, Eigen3
- A third tool along with SCRIP, ESMF_Regridder
- MOAB repository: <https://bitbucket.org/fathomteam/moab>

Online MOAB-regridder coupling workflow

- **Offline phase:** only the mesh generation step
- **Online phase init:**
 - For each component in E3SM
 - Replicate component meshes with MOAB database
 - Migrate MOAB mesh from component PEs to coupler PEs
 - Compute pair-wise mesh intersections in parallel
 - Compute projection weights with TempestRemap
 - Store weights in a parallel SparseMatrix datastructure (not MCT)
- **Online phase run:**
 - Send solution data from a source component to coupler
 - Execute parallel matvec operation to compute remapped solution
 - Communicate solution to target component from coupler

Advancing front intersection

- Computing mesh intersection with the advancing front algorithm is an independent task in parallel once we have the full coverage mesh
- Several opportunities to accelerate with fine-grained parallelism (threads)



Implementation in E3SM

```
cime_driver (Initialize mesh in MPAS)
  cime_init
    component_init_cc
      ocn_mct_init (in MPAS)
#ifdef HAVE_MOAB
      call mpas_moab_instance(domain_ptr) ! should return MPOID
      call mpas_log_write('initialized MOAB MPAS ocean instance')
#endif
#endif
```

```
cime_driver (Initialize mesh in coupler)
  cime_init
    component_init_cx !called once for each model
      call seq_mctext_gsmapiInit(comp(1))
#ifdef HAVE_MOAB
      call cplcomp_moab_Init(comp(1))
#endif
#endif
```

Current status

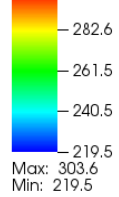
- Can compile and build E3SM that
 - Instantiates MOAB
 - Defines full meshes in HOMME and MPAS
 - Migrates full meshes to coupler
 - Computes intersection in coupler
 - Computes weights
 - Migrates data from atmosphere to coupler
 - Applies weights
 - Migrates regridded data to ocean

Online interpolation results

DB: flntxTarget.h5m

Pseudocolor
DB: wholeFineATM_T_02.h5m

Var: NODE_a2oDBG

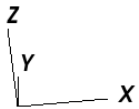
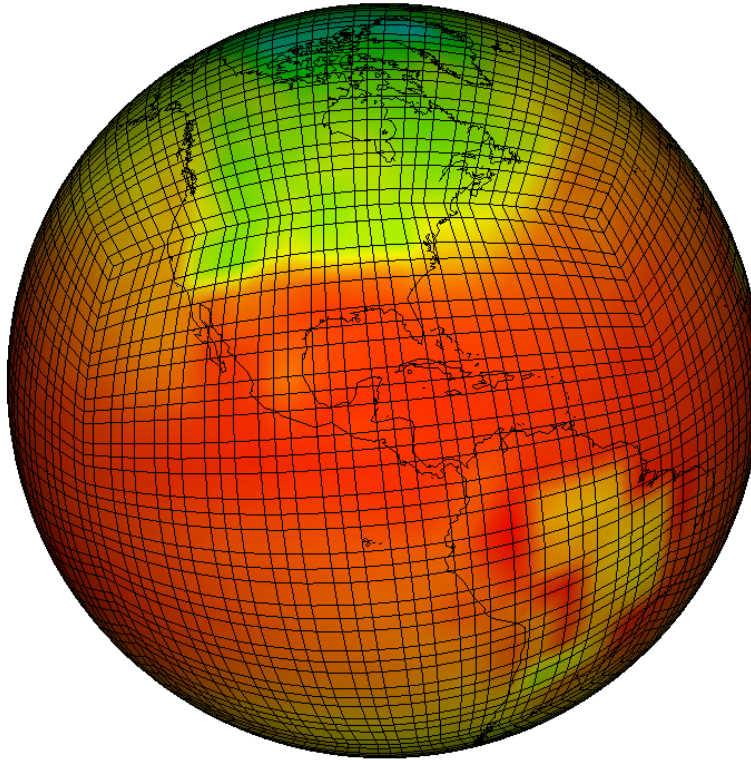


Mesh
DB: bound.vtk

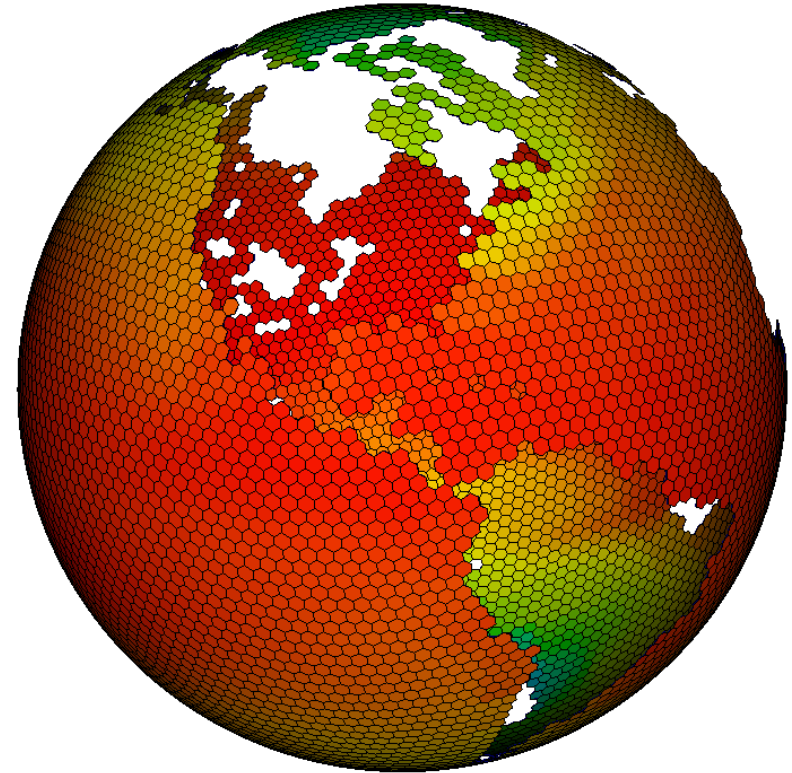
Var: mesh

Mesh
DB: wholeFineATM_T_02.h5m

Var: mesh



r
32oTAG_proj
.5
1.9
1.3
1.6
1.0



x

Interpolation of lowest atmosphere layer temp from HOMME to MPAS working in parallel

Next steps

- Use TempestRemap-generated maps in current online-offline system (for E3SM v1.x)
- Implement full data pathway (all fields exchanged) with MOAB-based system
- Scaling and performance studies of MOAB system.
- Replace MCT-based driver with MOAB-based driver (E3SM v2.0)

- More info: <http://sigma.mcs.anl.gov/moab-library/>
- Soon in <http://github.com/E3SM-Project/E3SM>

```
cime/src/drivers/mct/  
cime/src/drivers/moab/
```