Geosci. Model Dev., 13, 2355–2377, 2020
https://doi.org/10.5194/gmd-13-2355-2020

Geoscientific
Model Development
Open Access

Improving climate model coupling through a complete mesh representation: a case study with E3SM (v1) and MOAB (v5.x)

Vijay S. Mahadevan, Iulian Grindeanu, Robert Jacob, and Jason Sarich

Argonne National Laboratory, 9700 S. Cass Avenue, Lemont, IL, USA

Correspondence: Vijay S. Mahadevan (mahadevan@anl.gov)

Received: 3 November 2018 – Discussion started: 20 November 2018
Revised: 24 April 2019 – Accepted: 13 May 2019 – Published: 26 May 2020

Abstract. One of the fundamental factors contributing to the spatiotemporal inaccuracy in climate modeling is the mapping of solution field data between different discretizations and numerical grids used in the coupled component models. The typical climate computational workflow involves evaluation and serialization of the remapping weights during the preprocessing step, which is then consumed by the coupled driver infrastructure during simulation to compute field projections. Tools like Earth System Modeling Framework (ESMF) (Hill et al., 2004) and TempestRemap (Ullrich et al., 2013) offer capability to generate conservative remapping weights, while the Model Coupling Toolkit (MCT) (Larson et al., 2001) that is utilized in many production climate models exposes functionality to make use of the operators to solve the coupled problem. However, such multistep processes present several hurdles in terms of the scientific workflow and impede research productivity. In order to overcome these limitations, we present a fully integrated infrastructure based on the Mesh Oriented datABase (MOAB) (Tautges et al., 2004; Mahadevan et al., 2015) library, which allows for a complete description of the numerical grids and solution data used in each submodel. Through a scalable advancing-front intersection algorithm, the supermesh of the source and target grids are computed, which is then used to assemble the high-order, conservative, and monotonicity-preserving remapping weights between discretization specifications. The Fortran-compatible interfaces in MOAB are utilized to directly link the submodels in the Energy Exascale Earth System Model (E3SM) to enable online remapping strategies in order to simplify the coupled workflow process. We demonstrate the superior computational efficiency of the remapping algorithms in comparison with other state-of-the-science tools and present strong scaling results on large-scale machines for computing remapping weights between the spectral element atmosphere and finite volume discretizations on the polygonal ocean grids.

1 Introduction

Understanding Earth's climate evolution through robust and accurate modeling of the intrinsically complex, coupled ocean–atmosphere–land–ice–biosphere models requires extreme-scale computational power (Washington et al., 2008). In such coupled applications, the different component models may employ unstructured spatial meshes that are specifically generated to resolve problem-dependent solution variations, which introduces several challenges in performing a consistent solution coupling. It is known that operator decomposition and unresolved coupling errors in partitioned atmosphere and ocean model simulations (Beljaars et al., 2017), or physics and dynamics components of an atmosphere, can lead to large approximation errors that cause severe numerical stability issues. In this context, one factor contributing to the spatiotemporal accuracy is the mapping between different discretizations of the sphere used in the components of a coupled climate model. Accurate remapping strategies in such multi-mesh problems are critical to preserve higher-order resolution but are in general computationally expensive given the disparate spatial scales across which conservative projections are calculated. Since the primal solution or auxiliary-derived data defined on a donor physics component mesh (source model) need to be transferred to their coupled dependent physics mesh (target model), robust numerical algorithms are necessary to preserve discretization accuracy during these operations (Grandy, 1999; de Boer et al., 2008), in addition to conservation and monotonicity properties in the field profile.

An important consideration is that in addition to maintaining the overall discretization accuracy of the solution during remapping, global conservation and sometimes local element-wise conservation for critical quantities (Jiao and Heath, 2004) need to be imposed during the workflow. Such stringent requirements on key flux fields that couple components along boundary interfaces are necessary in order to mitigate any numerical deviations in coupled climate simulations. Note that these physics meshes are usually never embedded or are not linked by any trivial linear transformations, which render existence of exact projection or interpolation operators unfeasible, even if the same continuous geometric topology is discretized in the models. Additionally, the unique domain decomposition used for each of the component physics meshes complicates the communication pattern during intra-physics transfer, since aggregation of point location requests needs to be handled efficiently in order to reduce overheads during the remapping workflow (Plimpton et al., 2004; Tautges and Caceres, 2009).

Adaptive block-structured cubed-sphere or unstructured refinement of icosahedral/polygonal meshes (Slingo et al., 2009) are often used to resolve the complex fluid dynamics behavior in atmosphere and ocean models efficiently. In such models, conservative, local flux-preserving remapping schemes are critically important (Berger, 1987) to effectively reduce multi-mesh errors, especially during computation of tracer advection such as water vapor or CO_2 (Lauritzen et al., 2010). This is also an issue in atmosphere models where physics and dynamics are computed on non-embedded grids (Dennis et al., 2012), and the improper spatial coupling between these multiscale models could introduce numerical artifacts. Hence, the availability of different consistent and accurate remapping schemes under one flexible climate simulation framework is vital to better understand the pros and cons of the adaptive multi-resolution choices (Reichler and Kim, 2008).

1.1 Hub-and-spoke vs. distributed coupling workflow

The hub-and-spoke centralized model as shown in Fig. 1a is used in the current Exascale Earth System Model (E3SM; https://www.e3sm.org, last access: 16 May 2020) driver and relies on several tools and libraries that have been developed to simplify the regridding workflow within the climate community. Most of the current tools used in E3SM and the Community Earth System Model (CESM) (Hurrell et al., 2013) are included in a single package called the Common Infrastructure for Modeling the Earth (CIME; http://esmci.github.io/cime, last access: 16 May 2020), which builds on previous couplers used in CESM (Craig et al., 2005, 2012). These modeling tools approach the problem in a two-step computational process:

1. Compute the projection or remapping weights for a solution field from a source component physics to a target component physics as an “offline process”.

2. During runtime, the CIME coupled solver loads the remapping weights from a file and handles the partition-aware “communication and weight matrix” application to project coupled fields between components.

The first task in this workflow is currently accomplished through a variety of standard state-of-the-science tools such as the Earth Science Modeling Framework (ESMF) (Hill et al., 2004), Spherical Coordinate Remapping and Interpolation Package (SCRIP) (Jones, 1999), and TempestRemap (Ullrich et al., 2013; Ullrich and Taylor, 2015). The Model Coupling Toolkit (MCT) (Larson et al., 2001; Jacob et al., 2005) used in the CIME solver provides data structures for the second part of the workflow. Traditionally, the first workflow phase is executed decoupled from the simulation driver during a preprocessing step, and hence any updates to the field discretization or the underlying mesh resolution immediately necessitate recomputation of the remapping weight generation workflow with updated inputs. This process flow also prohibits the component solvers from performing any runtime spatial adaptivity, since the remapping weights have to be recomputed dynamically after any changes in grid positions. To overcome such deficiencies, and to accelerate the current coupling workflow, recent efforts have been undertaken to implement a fully integrated remapping weight generation process within E3SM using a scalable infrastructure provided by the topology, decomposition, and data-aware Mesh Oriented datABase (MOAB; http://sigma.mcs.anl.gov/moab-library, last access: 16 May 2020) (Tautges et al., 2004; Mahadevan et al., 2015) and TempestRemap (Ullrich et al., 2013) software libraries as shown in Fig. 1b. Note that regardless of whether a hub-and-spoke or distributed coupling model is used to drive the simulation, a minimal layer of driver logic is necessary to compute weighted combination of fluxes, validation metrics, and other diagnostic outputs.

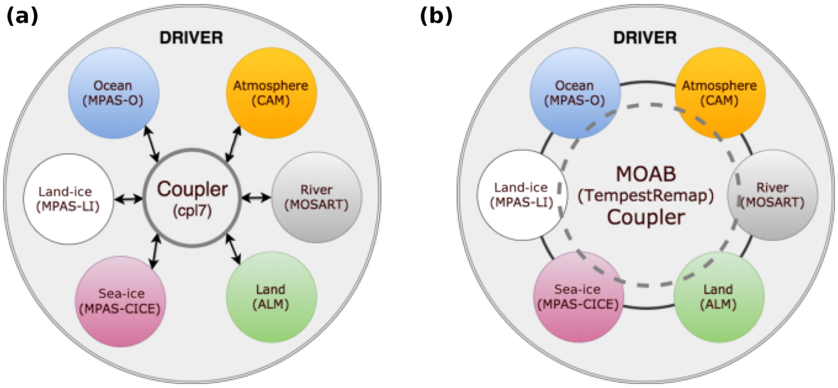


Figure 1. E3SM coupled climate solver: **(a)** current model; **(b)** newer MOAB-based coupler.

The paper is organized as follows. In Sect. 2, we present the necessary background and motivations to develop an online remapping workflow implementation in E3SM. Section 3 covers details on the scalable, mesh- and partition-aware, conservative remapping algorithmic implementation to improve scientific productivity of the climate scientists, and to simplify the overall computational workflow for complex problem simulations. Then, the performance of these algorithms is first evaluated in serial for various grid combinations, and the parallel scalability of the workflow is demonstrated on large-scale machines in Sect. 4.

2 Background

Conservative remapping of nonlinearly coupled solution fields is a critical task to ensure consistency and accuracy in climate and numerical weather prediction simulations (Slingo et al., 2009). While there are various ways to compute a projection of a solution defined on a source grid Ω_S to a target grid Ω_T, the requirements related to global or local conservation in the remapped solution reduce the number of potential algorithms that can be employed for such problems.

Depending on whether (global or local) conservation is important, and if higher-order, monotone interpolators are required, there are several consistent algorithmic options that can be used (de Boer et al., 2008). All of these different remapping schemes usually have one of these characteristic traits: non-conservative (NC), globally conservative (GC), and locally conservative (LC). Note that strong local-conservation prescriptions also guarantee global conservation for the remapped fields.

1. NC/GC: solution interpolation approximations:
   - NC: (approximate or exact) nearest-neighbor interpolation;
   - NC/GC: radial basis function (RBF) (Flyer and Wright, 2007) interpolators and patch-based least-squares reconstructions (Zienkiewicz and Zhu, 1992; Fleishman et al., 2005); and
   - GC: consistent finite element (FE) interpolation (bilinear, biquadratic, etc.) with area renormalization.
2. LC: mass- (L_2) and gradient-preserving (H_1) projections:
   - embedded finite element (FE), finite difference (FD), and finite volume (FV) meshes in adaptive computations;
   - intersection-based field integrators with consistent higher-order discretization (Jones, 1999); and
   - constrained projections to ensure conservation (Berger, 1987; Aguerre et al., 2017) and monotonicity (Rančić, 1995).

Typically, in climate applications, flux fields are interpolated using first-order (locally) conservative interpolation, while other scalar fields use non-conservative but higher-order interpolators (e.g., bilinear or biquadratic). For scalar solutions that do not need to be conserved, consistent FE interpolation, patch-wise reconstruction schemes (Fornberg and Piret, 2008), or even nearest-neighbor interpolation (Blanco and Rai, 2014) can be performed efficiently using Kd-tree-based search-and-locate point infrastructure. Vector fields like velocities or wind stresses are interpolated using these same routines by separately tackling each Cartesian-decomposed component of the field. However, conservative remapping of flux fields requires computation of a supermesh (Farrell and Maddison, 2011), or a global intersection mesh that can be viewed as $\Omega_\mathrm{S} \bigcup \Omega_\mathrm{T}$, which is then used to compute projection weights that contain additional conservation and monotonicity constraints embedded in them.

In general, remapping implementations have three distinct steps to accomplish the projection of solution fields from a source to a target grid. First, the target points of interest are identified and located in the source grid, such that the target cells are a subset of the covering (source) mesh. Next, an intersection between this covering (source) mesh and the target mesh is performed, in order to calculate the individual weight contribution to each target cell, without approximations to the component field discretizations that can be defined with arbitrary-order FV or FE basis. Finally, application of the weight matrix yields the projection required to conservatively transfer the data onto the target grid.

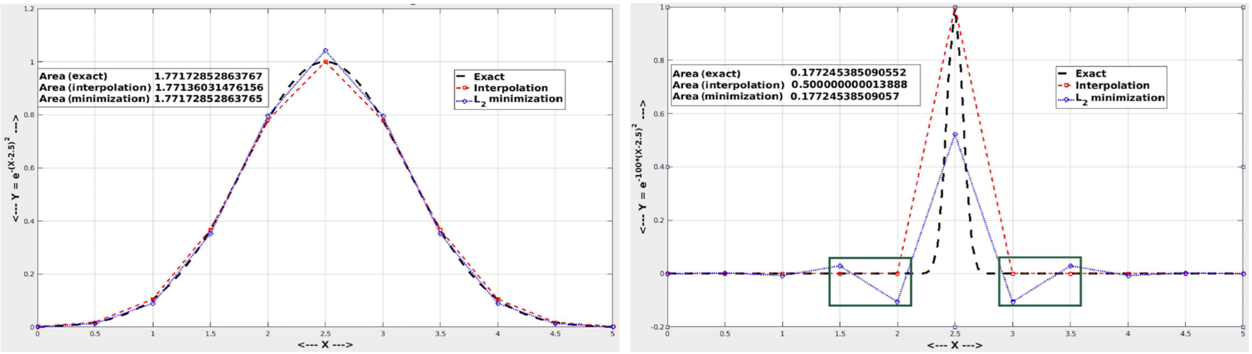
To illustrate some key differences between some NC to GC or LC schemes, we show a 1-D Gaussian hill solution, projected onto a coarse grid through linear basis interpolation and weighted least-squares (L_2) minimization, as shown in Fig. 2. While the point-wise linear interpolator is computationally efficient, and second-order accurate (Fig. 2a) for smooth profiles, it does not preserve the exact area under the curve. In contrast, the L_2 minimizer conserves the global integral area but can exhibit spurious oscillatory modes, as shown in Fig. 2b, when dealing with solutions with strong gradients (Gibbs phenomena; Gottlieb and Shu, 1997). This demonstration confirms that even for the simple 1-D example, a conservative and monotonic projector is necessary to preserve both stability and accuracy for repeated remapping operator applications, in order to accurately transfer fields between grids with very different resolutions. These requirements are magnified manifold when dealing with real-world climate simulation data.

While there is a delicate balance in optimizing the computational efficiency of these operations without sacrificing the numerical accuracy or consistency of the procedure, several researchers have implemented algorithms that are useful for a variety of problem domains. In recent years, the growing interest to rigorously tackle coupled multiphysics applications has led to research efforts focused on developing new regridding algorithms. The Data Transfer Kit (DTK; https://github.com/ORNL-CEES/DataTransferKit, last access: 16 May 2020) (Slattery et al., 2013) from Oak Ridge National Labs was originally developed for nuclear engineering applications but has been extended for other problem domains through custom adaptors for meshes. DTK is more suited for non-conservative interpolation of scalar variables with either mesh-aware (using consistent discretization bases) or RBF-based meshless (point-cloud) representations (Slattery, 2016) that can be extended to model transport schemes on a sphere (Flyer and Wright, 2007). The Portage (https://github.com/laristra/portage, last access: 16 May 2020) library (Herring et al., 2017) from Los Alamos National Laboratory also provides several key capabilities that are useful for geology and geophysics modeling applications including porous flow and seismology systems. Using advanced clipping algorithms to compute the intersection of axis-aligned squares/cubes against faces of a triangle/tetrahedron in 2-D and 3-D, respectively, general intersections of arbitrary convex polyhedral domains can be computed efficiently (Powell and Abel, 2015). Support for conservative solution transfer between grids and bound preservation (to ensure monotonicity) (Certik et al., 2017) has also been recently added. While Portage does support hybrid-level parallelism (MPI + OpenMP), demonstrations on large-scale machines to compute remapping weights for climate science applications have not been pursued previously. Based on the software package documentation, support for remapping of vector fields with conservation constraints in DTK and Portage is not directly available for use in climate workflows. Additionally, unavailability of native support for projection of high-order spectral element data on a sphere onto a target mesh restricts the use of these tools for certain component models in E3SM.

In Earth science applications, the state-of-the-science regridding tool that is often used by many researchers is the ESMF (https://www.earthsystemcog.org/projects/esmf, last access: 16 May 2020) library, and the set of utility tools that are distributed along with it (Collins et al., 2005; Dunlap et al., 2013), to simplify the traditional offline–online computational workflow as described in Sect. 1.1. ESMF is implemented in a component architecture (Zhou, 2006) and provides capabilities to generate the remapping weights for different discretization combinations on the source and target grids in serial and parallel. ESMF provides a stand-alone tool, ESMF_RegridWeightGen (https://www.earthsystemcog.org/projects/regridweightgen, last access: 16 May 2020), to generate offline weights that can be consumed by climate applications such as E3SM. ESMF also exposes interfaces that enable drivers to directly invoke the remapping algorithms in order to enable the fully online workflow as well.

Currently, the E3SM components are integrated together in a hub-and-spoke model (Fig. 1a), with the inter-model communication being handled by the Model Coupling Toolkit (MCT) (Larson et al., 2001; Jacob et al., 2005) in CIME. The MCT library consumes the offline weights generated with ESMF or similar tools and provides the functionality to interface with models, decompose the field data, and apply the remapping weights loaded from a file during the setup phase. Hence, MCT serves to abstract the communication of data in the E3SM ecosystem. However, without the offline remapping weight generation phase for fixed grid resolutions and model combinations, the workflow in Fig. 1a is incomplete.

Similar to the CIME-MCT driver used by E3SM, OASIS3-MCT (Valcke, 2013; Craig et al., 2017) is a coupler used by many European climate models, where the interpolation weights can be generated offline through SCRIP (included as part of OASIS3-MCT). An option to call SCRIP in an online mode is also available. The OASIS team has recently parallelized SCRIP to speed up its calculation time (Valcke et al., 2018). OASIS3-MCT also supports application of global conservation operations after interpolation and does not require a strict hub-and-spoke coupler. Similar to the coupler in CIME, OASIS3-MCT utilizes MCT to perform both the communication of fields between components and the ap-



(a) Gaussian hill $(x_0, \delta) = (2.5, 1.0)$ (b) Gaussian hill $(x_0, \delta) = (2.5, 0.1)$

Figure 2. An illustration comparing point interpolation vs. L_2 minimization; impact on conservation and monotonicity properties.

plication of the precomputed interpolation weights in parallel.

ESMF and SCRIP traditionally handle only cell-centered data that target finite volume discretizations (FV-to-FV projections), with first- or second-order conservation constraints. Hence, generating remapping weights for atmosphere–ocean grids with a spectral element (SE) source grid definition requires generation of an intermediate and spectrally equivalent “dual” grid, which matches the areas of the polygons to the weight of each Gauss–Lobatto–Legendre (GLL) nodes (Mundt et al., 2016). Such procedures add more steps to the offline process and can degrade the accuracy in the remapped solution since the original spectral order is neglected (transformation from p order to first order). These procedures may also introduce numerical uncertainty in the coupled solution that could produce high solution dispersion (Ullrich et al., 2016).

To calculate remapping weights directly for high-order spectral element grids, E3SM uses the TempestRemap C++ library (https://github.com/ClimateGlobalChange/tempestremap, last access: 16 May 2020) (Ullrich et al., 2013). TempestRemap is a uni-process tool focused on the mathematically rigorous implementations of the remapping algorithms (Ullrich and Taylor, 2015; Ullrich et al., 2016) and provides higher-order conservative and monotonicity-preserving interpolators with different discretization basis such as FV, the spectrally equivalent continuous Galerkin FE with GLL basis (cGLL), and discontinuous Galerkin FE with GLL basis (dGLL). This library was developed as part of the effort to fill the gap in generating consistent remapping operators for non-FV discretizations without a need for intermediate dual meshes. Computation of conservative interpolators between any combination of these discretizations (FV, cGLL, dGLL) and grid definitions is supported by TempestRemap library. However, since this regridding tool can only be executed in serial, the usage of TempestRemap prior to the work presented here has been restricted primarily to generating the required mapping weights in the offline stage.

Even though ESMF and OASIS3-MCT have been used in online remapping studies, weight generation as part of a preprocessing step currently remains the preferred workflow for many production climate models. While this decoupling provides flexibility in terms of choice of remapping tools, the data management of the mapping files for different discretizations, field constraints, and grids can render provenance, reproducibility, and experimentation a difficult task. It also precludes the ability to handle moving or dynamically adaptive meshes in coupled simulations. However, it should be noted that the shift of the remapping computation process from a preprocessing stage in the workflow, to the simulation stage, imposes additional onus on the users to better understand the underlying component grid properties, their decompositions, the solution fields being transferred, and the preferred options for computing the weights. This also raises interesting workflow modifications to ensure verification of the online weights such that consistency, conservation, and dissipation of key fields are within user-specified constraints. In the implementation discussed here, the online remapping computation uses the exact same input grids and specifications like the offline workflow, along with ability to write the weights to file, which can be used to run detailed verification studies as needed.

There are several challenges in scalably computing the regridding operators in parallel, since it is imperative to have both a mesh- and partition-aware data structure to handle this part of the regridding workflow. A few climate models have begun to calculate weights online as part of their regular operation. The ICON GCM (Wan et al., 2013) uses YAC (Hanke et al., 2016) and FGOALS (Li et al., 2013) uses the C-Coupler (Liu et al., 2014, 2018) framework. These codes expose both offline and online remapping capabilities with parallel decomposition management similar to the ongoing effort presented in the current work for E3SM. Both of these packages provide algorithmic options to perform in-memory search-and-locate operations, interpolation of field data between meshes with first-order conservative remapping, higher-order patch-recovery (Zienkiewicz and Zhu, 1992) and RBF schemes and the NC nearest-neighbor queries. The use of non-blocking communication for field data in these packages aligns closely with scalable strategies implemented in MCT (Jacob et al., 2005). While these capabilities are used routinely in production runs for their respective models, the motivation for the work presented here is to tackle coupled high-resolution runs on next-generation architectures with scalable algorithms (the high-resolution E3SM coupler routinely runs on 13 000 MPI tasks), without sacrificing numerical accuracy for all discretization descriptions (FV, cGLL, dGLL) on unstructured grids.

In the E3SM workflow supported by CIME, the ESMF regridder understands the component grid definitions and generates the weight matrices (offline). The CIME driver loads these operators at runtime and places them in MCT data types, which treat them as discrete operators to compute the interpolation or projection of data on the target grids. Additional changes in conservation requirements or monotonicity of the field data cannot be imposed as a runtime or post-processing step in such a workflow. In the current work, we present a new infrastructure with scalable algorithms implemented using the MOAB mesh library and TempestRemap package to replace the ESMF-E3SM-MCT remapper/coupler workflow. A detailed review of the algorithmic approach used in the MOAB-TempestRemap (MBTR) workflow, along with the software interfaces exposed to E3SM, is presented next.

3 Algorithmic approach

Efficient, conservative, and accurate multi-mesh solution transfer workflows (Jacob et al., 2005; Tautges and Caceres, 2009) are a complex process. This is due to the fact that in order to ensure conservation of critical quantities in a given norm, exact cell intersections between the source and target grids have to be computed. This is complicated in a parallel setting since the domain decompositions between the source and target grids may not have any overlaps, making it a potentially all-to-all collective communication problem. Hence, efficient implementations of regridding operators need to be mesh, resolution, field, and decomposition aware in order to provide optimal performance in emerging architectures.

Fully online remapping capability within a complex ecosystem such as E3SM requires a flexible infrastructure to generate the projection weights. In order to fulfill these needs, we utilize the MOAB mesh data structure combined with the TempestRemap libraries in order to provide an in-memory remapping layer to dynamically compute the weight matrices during the setup phase of the simulations for static source–target grid combinations. For dynamically adaptive and moving grids, the remapping operator can be recomputed at runtime as needed. The introduction of such a software stack allows higher-order conservation of fields while being able to transfer and maintain field relations in parallel, within the context of the fully decomposed mesh view. This is an improvement to the E3SM workflow where MCT is oblivious to the underlying mesh data structure in the component models. Having a fully mesh-aware implementation with element connectivity and adjacency information, along with parallel ghosting and decomposition information, also provides opportunities to implement dynamic load-balancing algorithms to gain optimal performance on large-scale machines. Without the mesh topology, MCT is limited to performing trivial decompositions based on global ID spaces during mesh migration operations. YAC interpolator (Hanke et al., 2016) and the multidimensional Common Remapping software (CoR) in C-Coupler2 (Liu et al., 2018) provide similar capabilities to perform a parallel tree-based search for point location and interpolation through various supported numerical schemes.

MOAB is a fully distributed, compact, array-based mesh data structure, and the local entity lists are stored in ranges along with connectivity and ownership information, rather than explicit lists, thereby leading to a high degree of memory compression. The memory constraints per process scale well in parallel (Tautges and Caceres, 2009) and are only proportional to the number of entities in the local partition, which reduces as the number of processes increases (strong scaling limit). This is similar to the Global Segment Map (GSMap) in MCT, which in contrast is stored in every processor, leading to $O(N_x)$ memory requirements. The parallel communication infrastructure in MOAB is heavily leveraged (Tautges et al., 2012) to utilize the scalable crystal router algorithm (Fox et al., 1989; Schliephake and Laure, 2015) in order to scalably communicate the covering cells to different processors. This parallel mesh infrastructure in MOAB provides the necessary algorithmic tools for optimally executing online remapping strategies, so that MCT in E3SM can be replaced with a MOAB-based coupler.

In order to illustrate the online remapping algorithm implemented with the MOAB-TempestRemap infrastructure, we define the following terms. Let $N_{\mathrm{c,S}}$ be the component processes for source mesh, $N_{\mathrm{c,T}}$ be the component processes for target mesh, and N_x be the coupler processes where the remapping operator is computed. More generally, the problem statement can be defined as transferring a solution field U defined on the domain Ω_{S} and processes $N_{\mathrm{c,S}}$ to the domain Ω_{T} and processes $N_{\mathrm{c,T}}$ through a centralized coupler with domain information $\Omega_{\mathrm{S}} \bigcup \Omega_{\mathrm{T}}$ defined on N_x processes. Such a complex online remapping workflow for projecting the field data from a source to target mesh follows the algorithm shown in Algorithm 1.

In the following sections, the new E3SM online remapping interface implemented with a combination of the MOAB and TempestRemap libraries is explained. Details regarding the algorithmic aspects to compute conservative, high-order

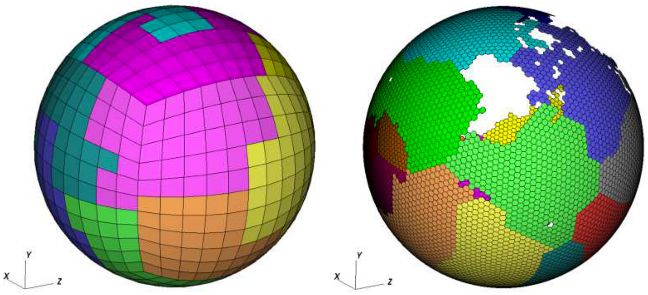
Algorithm 1 MOAB-TempestRemap parallel regridding workflow

```
Input: Partitioned and distributed native component meshes on $N_{c,S}$ source and $N_{c,T}$ target processes
Result: Remapping weight matrix $W_{S\to T}$ computed for a source $(S)$ and target $(T)$ mesh pair on $N_x$ coupler processes

Scope: Coupler $N_x \leftarrow$ component mesh $N_{c,l}$, where $l \in [S,T]$
for each component $l \in [S,T]$ do
    – create in-memory copy of component unstructured mesh and data using MOAB interfaces (Sect. 3.1)
    – migrate MOAB component mesh to coupler; repartition from $N_{c,l} \to N_x$ (Sect. 3.2)
end for

Scope: Compute pair-wise intersection mesh on coupler processes $N_x$
for each mesh pair to be regridded: $\Omega_S$ and $\Omega_T$ in $N_x$ do
    Ensure: {local source mesh fully covers target mesh}
    if $(\Omega_T - \Omega_T \cap \Omega_S) \neq 0$ then
        collectively gather coverage mesh $\Omega_{Sc}$ on $N_x \mid (\Omega_T - \Omega_T \cap \Omega_{Sc}) = 0$ (Sect. 3.4.1)
    end if
    – store communication graph to send/receive between $N_{c,l}$ and $N_x$
    – compute $\Omega_{ST} = \Omega_{Sc} \cap \Omega_T$ through an advancing-front algorithm (Löhner and Parikh, 1988; Gander and Japhet, 2009) (Sect. 3.3.1)
    – evaluate source/target element mapping for $e_i \in \Omega_{ST}$
    – exchange ghost cell information for $\Omega_{ST}$
end for

Scope: Integrate over $\Omega_{ST}$ to compute remapping weights
for each intersection polygon element $e_i \in \Omega_{ST}$ do
    – Tessellate $e_i$ into triangular elements with reproducible ordering
    – Compute projection integral with consistent Triangular quadrature rules
    – Determine row/col DoF coupling through $e_i$ parent association to $\Omega_S/\Omega_T$
    – Assemble local matrix weights such that $W_{S\to T} = \sum_1^{N_x} w_{ij}$, where $w_{ij}$ represents the coupling between local target DoF (row $i$) and source DoF (col $j$) in projection operator (Sect. 3.5)
end for
```

remapping weights in parallel, without sacrificing discretization accuracy on next-generation hardware, are presented.

3.1 Interfacing to component models in E3SM

Within the E3SM simulation ecosystem, there are multiple component models (atmosphere–ocean–land–ice–runoff) that are coupled to each other. While the MCT infrastructure primarily manages the global degree-of-freedom (DoF) partitions without a notion of the underlying mesh, the new MOAB-based coupler infrastructure provides the ability to natively interface to the component mesh and intricately understand the field DoF data layout associated with each model. MOAB can recognize the difference between values on a cell center and values on a cell edge or corner. In the current work, the MOAB mesh database has been used to create the relevant integration abstraction for the High-Order Methods Modeling Environment (HOMME) atmosphere model (Thomas and Loft, 2005; Taylor et al., 2007) (cubed-sphere SE grid) and the Model for Prediction Across Scales (MPAS) ocean model (Ringler et al., 2013; Petersen et al., 2015) (polygonal meshes with holes representing land and ice regions). Since details of the mesh are not available at the level of the coupler interface, additional MOAB (Fortran) calls



(a) HOMME-SE mesh

(b) MPAS mesh

Figure 3. MOAB representation of partitioned component meshes.

via the `iMOAB` interface are added to HOMME and MPAS component models to describe the details of the unstructured mesh to MOAB with explicit vertex and element connectivity information, in contrast to MCT coupler that is oblivious to the underlying grid. The atmosphere–ocean coupling requires the largest computational effort in the coupler (since they cover about 70 % of the coupled domain), and hence the bulk of discussions in the current work will focus on remapping and coupling between these two component models.

MOAB can handle the finite element zoo of elements on a sphere (triangles, quadrangles, and polygons), making it an appropriate layer to store both the mesh layout (vertices, elements, connectivity, adjacencies) and the parallel decomposition for the component models along with information on shared and ghosted entities. While having a uniform partitioning methodology across components may be advantageous for improving the efficiency of coupled climate simulations, the parallel partition of the meshes are chosen according to the requirements in individual component solvers. Figure 3 shows examples of partitioned SE and MPAS meshes, visualized through the native MOAB plugin for VisIt (https://visit.llnl.gov, last access: 16 May 2020) (VisIt, 2005).

The coupled field data that are to be remapped from the source grid to the target grid also need to be serialized as part of the MOAB mesh database in terms of an internally contiguous MOAB data storage structure named a "tag" (Tautges et al., 2004). For E3SM, we use element-based tags to store the partitioned field data that are required to be remapped between components. Typically, the number of DoFs per element (nDoF_e) is determined based on the underlying discretization; $\text{nDoF}_\text{e} = p^2$ values in HOMME, where p is the order of SE discretization, and $\text{nDoF}_\text{e} = 1$ for the FV discretization in MPAS ocean. With this complete description of the mesh and associated data for each component model, MOAB contains the necessary information to proceed with the remapping workflow.

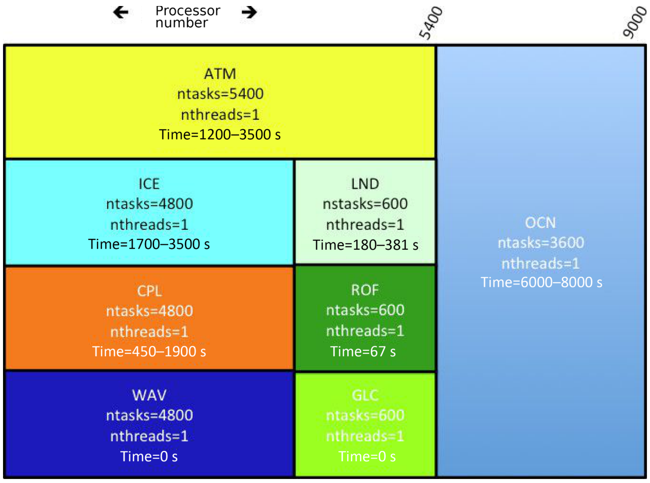


Figure 4. Example E3SM process execution layout for a problem case.

3.2 Migration of component mesh to coupler

E3SM's driver supports multiple modes of partitioning the various components in the global processor space. This is usually fine tuned based on the estimated computational load in each physics, according to the problem case definition. A sample process-execution (PE) layout for a E3SM run on 9000 processes with atmosphere (ATM) on 5400 and ocean (OCN) on 3600 processes is shown in Fig. 4. In the case shown in the schematic, $N_{\text{c,ATM}} = 5400$, $N_{\text{c,OCN}} = 3600$, and $N_x = 4800$. In such a PE layout, the atmosphere component mesh from HOMME, distributed on $N_{\text{c,ATM}}$ (5400) processes, needs to be migrated and redistributed on N_x (4800) processes and similarly from $N_{\text{c,OCN}}$ (3600) to N_x (4800) processes for the MPAS ocean mesh. In the hub-and-spoke coupling model, as shown in Fig. 1, the remapping computation is performed only in the coupler processors. Hence, inference of a communication pattern becomes necessary to ensure scalable data transfers between the components and the coupler. In the existing implementation, MCT handles such communication, which is being replaced by point-to-point communication kernels in MOAB to transfer mesh and data between different components or component–coupler PEs. Note that in a distributed coupler, source and target components can communicate directly, without any intermediate transfers (through the coupler). Under the unified infrastructure provided by MOAB, minimal changes are required to enable either the hub-and-spoke or the distributed coupler for E3SM runs, which offers opportunities to minimize time to solution without any changes in spatial coupling behavior.

For illustration, let N_c be the number of component process elements and N_x be the number of coupler process elements. In order to migrate the mesh and associated data from N_c to N_x, we first compute a trivial partition of elements that map directly in the partition space, the same partitioning as used in the CIME-MCT coupler. In MOAB, we have exposed parallel graph and geometric repartitioning schemes through interfaces to Zoltan (Devine et al., 2002) or ParMetis (Karypis et al., 1997) in order to evaluate optimized migration patterns to minimize the volume of data communicated between component and coupler. We intend to analyze the impact of different migration schemes on the scalability of the remapping operation in Sect. 4. These optimizations have the potential to minimize data movement in the MOAB-based remapper and to make it a competitive data broker to replace the current MCT (Jacob et al., 2005) coupler in E3SM.

We show an example of a decomposed ocean mesh (polygonal MPAS mesh) that is replicated in a E3SM problem case run on two processes in Fig. 5. Figure 5a is the original decomposed mesh on two processes $\in N_\mathrm{c}$, while Fig. 5b and c show the impact of migrating a mesh from $2N_\mathrm{c}$ processes to four processes $\in N_x$ with a trivial linear partitioner and a Zoltan-based geometric online partitioner. The decomposition in Fig. 5b shows that the element-ID-based linear partitioner can produce bad data locality, which may require a large number of nearest-neighbor communications when computing a source coverage mesh. The resulting communication pattern can also make the migration and coverage computation process non-scalable on larger core counts. In contrast, in Fig. 5c, the Zoltan partitioners produce much better load-balanced decompositions with hypergraph (PHG), recursive coordinate bisection (RCB), or recursive inertial bisection (RIB) algorithms to reduce communication overheads in the remapping workflow. In order to better understand the impact of online decomposition strategies on the overall remapping operation, we need to better understand the impact of the repartitioner on two communication-heavy steps:

1. mesh migration from component to coupler involving communication between $N_{\mathrm{c,s/t}}$ and N_x; and
2. computing the coverage mesh requiring gather/scatter of source mesh elements to cover local target elements.

In a hub-and-spoke model with online remapping, the best coupler strategy will require a simultaneous partition optimization for all grids such that mesh migration includes constraints on geometric coordinates of component pairs. While such extensions can be implemented within the infrastructure presented here, the performance discussions in Sect. 4 will only focus on the trivial and Zoltan-based partitioners. It is also worth noting that in a distributed coupler, pair-wise migration optimizations can be performed seamlessly using a master (target)–slave (source) strategy to maximize partition overlaps.

3.3 Computing the regridding operator

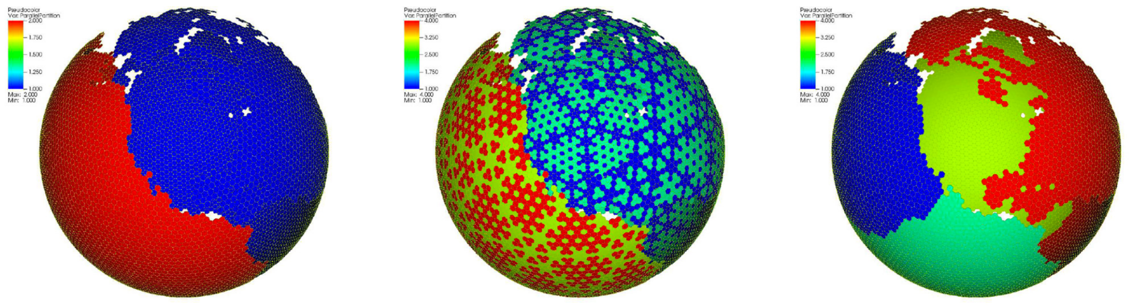
Standard approaches to compute the intersection of two convex polygonal meshes involve the creation of a Kd-tree (Hunt et al., 2006) or bounding volume hierarchy (BVH)-tree data structure (Ize et al., 2007) to enable fast element location of relevant target points. In general, each target point of interest is located on the source mesh by querying the tree data structure, and the corresponding (source) element is then marked as a contributor to the remapping weight computation of the target DoF. This process is repeated to form a list of source elements that interact directly according to the consistent discretization basis. TempestRemap, ESMF, and YAC use variations of this search-and-clip strategy tailored to their underlying mesh representations.

3.3.1 Advancing-front intersection – a linear complexity algorithm

The intersection algorithm used in this paper follows the ideas from Löhner and Parikh (1988) and Gander and Japhet (2013), in which two meshes are covering the same domain. At the core is an advancing-front method that aims to traverse through the source and target meshes to compute a union (super) mesh. First, two convex cells from the source coverage mesh and the target meshes that intersect are identified by using an adaptive Kd-tree search tree data structure. This process also includes determination of the seed (the starting cell) for the advancing front in each of the partitions independently. Advancing in both meshes using face adjacency information, incrementally all possible intersections are computed (Březina and Exner, 2017) accurately to a user defined tolerance (default $= 1e-15$) in linear time.

While the advancing-front algorithm is not restricted to convex cells, the intersection computation is simpler if they are strictly convex. If concave polygons exist in the initial source or target meshes, they are recursively decomposed into simpler convex polygons, by splitting along interior diagonals. Note that the intersection between two convex polygons results in a strictly convex polygon. Hence, the underlying intersection algorithm remains robust to resolve even arbitrary non-convex meshes covering the same domain space.

Figure 6 illustrates how the algorithm advances. In each local partition of the coupler PEs, a pair of source (blue) and target (red) cells that intersect is found (Fig. 6a). Using face adjacency queries for the source mesh, as shown in Fig. 6b, all source cells that intersect the current target cell are found. New possible pairs between cells adjacent to the current target cell and other source cells are added to a local queue. After the current target cell is resolved, a pair from the queue is considered next. Figure 6c shows the resolving of a second target cell, which is intersecting here with three source cells. If both meshes are contiguous, this algorithm guarantees to compute all possible intersections between source and target cells. Figure 6d shows the color map representation of the progression in which the intersection polygons were found, from blue (low count) towards red. This advance/progression is also illustrated through videos in both the serial (Mahade-



(a) Component mesh on two tasks (b) Migrated mesh on four tasks (trivial partitioner) (c) Migrated mesh on four tasks (Zoltan partitioner)

Figure 5. Migration strategies to repartition from N_c to N_x.

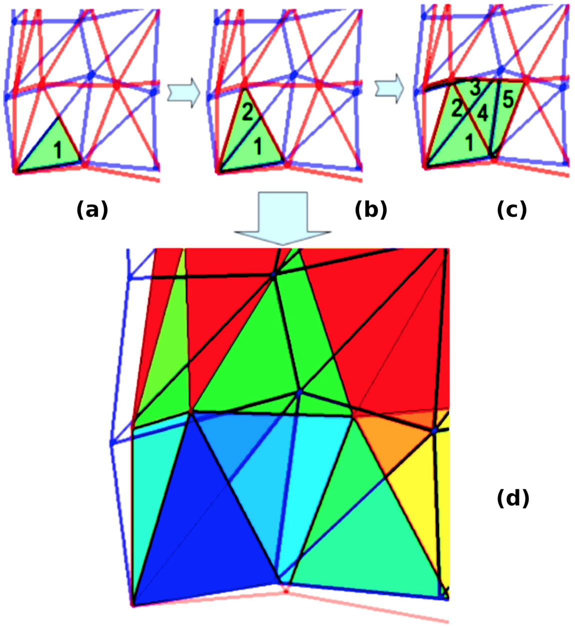


Figure 6. Illustration of the advancing-front intersection algorithm.

van et al., 2018a) and parallel (Mahadevan et al., 2018b) contexts on partitioned meshes.

This flooding-like advancing front needs a stable and robust methodology of intersecting edges/segments in two cells that belong to different meshes. Any pair of segments that intersect can appear in four different pairs of cells. A list of intersection points is maintained on each target edge, so that the intersection points are unique. Also, a geometric tolerance is used to merge intersection points that are close to each other or if they are proximal to the original vertices in both meshes. Decisions regarding whether points are inside, outside, or at the boundary of a convex enclosure are handled separately. If necessary, more robust techniques such as adaptive precision arithmetic procedures used in Triangle (Shewchuk, 1996) can be employed to resolve the fronts more accurately. Note that the advancing-front strategy can be employed for meshes with topological holes (e.g., ocean meshes in which the continents are excluded) without any further modifications by using a new pair for each disconnected region in the target mesh.

Note on gnomonic projection for spherical geometry

Meshes that appear in climate applications are often on a sphere. Cell edges are considered to be great circle arcs. A simple gnomonic projection is used to project the edges on one of the six planes parallel to the coordinate axis and tangent to the sphere (Ullrich et al., 2013). With this projection, all curvilinear cells on the sphere are transformed to linear polygons on a gnomonic plane, which simplifies the computation of intersection between multiple grids. Once the intersection points and cells are computed on the gnomonic plane, these are projected back onto the original spherical domain without approximations. This is possible due to the fact that intersection can be computed to machine precision as the edges become straight lines in a gnomonic plane (projected from great circle arcs on a sphere). If curves on a sphere are not great circle arcs (splines, for example), the intersections between those curves have to be computed using some nonlinear iterative procedures such as Newton–Raphson (depending on the representation of the curves).

3.4 Parallel implementation considerations

Existing infrastructure from MOAB (Tautges et al., 2004) was used to extend the advancing-front algorithm in parallel. The expensive intersection computation can be carried out independently, in parallel, once we redistribute the source mesh to envelope the target mesh areas fully, in a step we refer to as “source coverage mesh” computation.

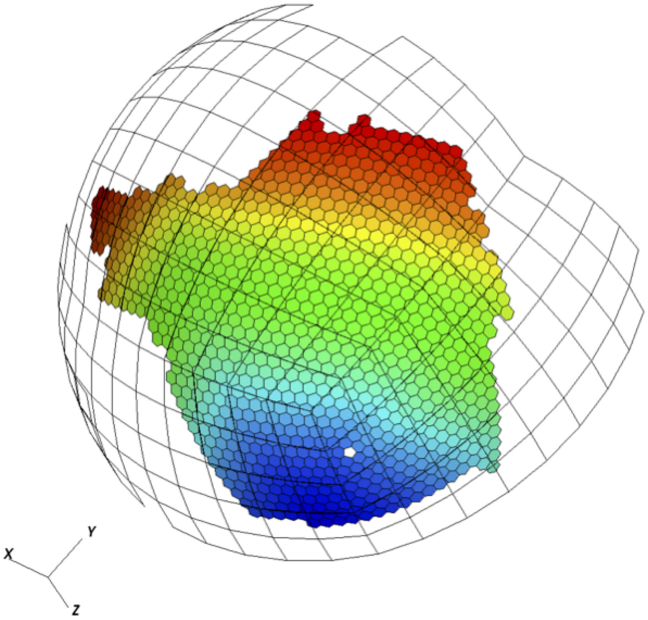


Figure 7. Source coverage mesh fully covers local target mesh; local intersection proceeds between the source atmosphere (quadrangle) and the target ocean (polygonal) grids.

3.4.1 Computation of a source coverage mesh

We select the target mesh as the driver for redistribution of the source mesh. On each task, we first compute the bounding box of the local target mesh. This information is then gathered and communicated to all coupler PEs and used for redistributing the local source mesh. Cells that intersect the bounding boxes of other processors are sent to the corresponding owner task using the aggregating crystal router algorithm that is particularly efficient in performing all-to-all strategies with $O(\log(N_x))$ complexity. This graph is computed once during the setup phase to establish point-to-point communication patterns, which are then used to pack and send/receive mesh elements or data at runtime.

This workflow guarantees that the target mesh on each processor is completely enveloped by the covering mesh repartitioned from its original source mesh decomposition, as shown in Fig. 7. In other words, the covering mesh fully encompasses and bounds the target mesh in each task. It is important to note that some source coverage cells might be sent to multiple processors during this step, depending on the target mesh resolution and decomposition.

Once the relevant covering mesh is accumulated locally on each process, the intersection computation can be carried out in parallel, completely independently, using the advancing-front algorithm (Sect. 3.3.1). After computation of the local intersection polygons, the vertices on the shared edges between processes are communicated to avoid duplication. In order to ensure consistent local conservation constraints in the weight matrix in the parallel setting, there might be a need for additional communication of ghost intersection elements to nearest neighbors. This extra communication step is only required for computing interpolators for flux variables and can generally be avoided when transferring scalar fields with non-conservative bilinear or higher-order interpolations. Note that this ghost exchange on the intersection mesh only requires nearest-neighbor communications within the coupler PEs, since the communication graph has been established a priori.

The parallel advancing-front algorithm presented here to globally compute the intersection supermesh can be extended to expose finer-grained parallelism using hybrid-threaded (OpenMP) programming or a task-based execution model, where each task handles a unique front in the computation queue. Such task or hybrid-threaded parallelism can be employed in combination with the MPI-based mesh decompositions. Using local partitions computed with Metis (Karypis and Kumar, 1998) and through standard coloring approaches, each thread or task can then proceed to compute the intersection elements until the front collides with another and until all the overlap elements have been computed in each process. Such a parallel hybrid algorithm has the potential to scale well even on heterogeneous architectures and provides options to improve the computational throughput of the regridding process (Löhner, 2014).

3.5 Computation of remapping operator with TempestRemap

For illustration, consider a scalar field $\boldsymbol{U}$ discretized with standard Galerkin finite element method (FEM) on source Ω_1 and target Ω_2 meshes with different resolutions. The projection of the scalar field on the target grid is in general given as follows:

$$\boldsymbol{U}_2(\boldsymbol{\Omega}_2) = \boldsymbol{\Pi}_1^2 \boldsymbol{U}_1(\boldsymbol{\Omega}_1), \quad (1)$$

where $\boldsymbol{\Pi}_1^2$ is the discrete solution interpolator of $\boldsymbol{U}$ defined on $\boldsymbol{\Omega}_1$ to $\boldsymbol{\Omega}_2$. This interpolator $\boldsymbol{\Pi}_1^2$ in Eq. (1) is often referred to as the remapping operator, which is precomputed in the coupled climate workflows using ESMF and TempestRemap. For embedded meshes, the remapping operator can be calculated exactly as a restriction or prolongation from the source to target grid. However, for general unstructured meshes and in cases where the source and target meshes are topologically different, the numerical integration to assemble $\boldsymbol{\Pi}_1^2$ needs to be carried out on the supermesh (Ullrich and Taylor, 2015). Since a unique source and target parent element exist for every intersection element belonging to the supermesh $\boldsymbol{\Omega}_1 \bigcup \boldsymbol{\Omega}_2$, $\boldsymbol{\Pi}_1^2$ is assembled as the sum of local mass matrix contributions on the intersection elements, by using the consistent discretization basis for the source and target field descriptions (Ullrich et al., 2016). The intersection mesh typically contains arbitrary convex polygons, and hence subsequent triangulation may be necessary before evaluating the integration. This global linear operator directly couples source and target DoFs based on the participating intersection element parents (Ullrich et al., 2009).

MOAB supports point-wise FEM interpolation (bilinear and higher-order spectral) with local or global subset normalization (Tautges and Caceres, 2009), in addition to a conservative first-order remapping scheme. However, higher-order conservative monotone weight computations are currently unsupported natively. To fill this gap for climate applications, and to leverage existing developments in rigorous numerical algorithms to compute the conservative weights, interfaces to TempestRemap in MOAB were added to scalably compute the remap operator in parallel, without sacrificing field discretization accuracy. The MOAB interface to the E3SM component models provides access to the underlying type and order of field discretization, along with the global partitioning for the DoF numbering. Hence, the projection or the weight matrix can be assembled in parallel by traversing through the intersection elements and associating the appropriate source and target DoF parent to columns and rows, respectively. The MOAB implementation uses a sparse matrix representation using the Eigen3 (https://eigen.tuxfamily.org, last access: 16 May 2020) library (Guennebaud et al., 2010) to store the local weight matrix. Except for the particular case of projection onto a target grid with cGLL description, the matrix rows do not share any contributions from the same source DoFs. This implies that for FV and dGLL target field descriptions, the application of the weight matrix does not require global collective operations and sparse matrix vector (SpMV) applications scale ideally (still memory bandwidth limited). In the cGLL case, we perform a reduction of the parallel vector along the shared DoFs to accumulate contributions exactly. However, it is non-trivial to ensure full bit-for-bit (BFB) reproducibility during such reductions, and currently, the MBTR workflow does not support exact reproducibility. Requirements for rigorous bit-wise reproduction for online remapping need careful implementation to enforce that the advancing-front intersection and weight matrix is computed in the exact same global element order, in addition to ensuring that the parallel SpMV products are reduced identically, independent of the parallel mesh decompositions.

It is also possible to use the transpose of the remapping operator computed between a particular source and target component combination to project the solution back to the original source grid. Such an operation has the advantage of preserving the consistency and conservation metrics originally imposed in finding the remapping operator and reduces computation cost by avoiding recomputation of the weight matrix for the new directional pair. For example, when computing the remap operator between atmosphere and ocean models (with holes), it is advantageous to use the atmosphere model as the source grid, since the advancing-front seed computation may require multiple trials if the initial front begins within a hole in the source mesh. Given that the seed or the initial cell determination on the target mesh is chosen at random, the corresponding intersecting cell on the source mesh found through a linear search could be contained within a hole in the source mesh. In such a case, a new target cell is then chosen and the source cell search is repeated. Hence, multiple trials may be required for the advancing-front algorithm to start propagating, depending on the mesh topology and decomposition. Note that the linear search in the source mesh can easily be replaced with a Kd-tree data structure to provide better computational complexity for cases where both source and target meshes have many holes. Additionally, such transpose vector applications can also make the global coupling symmetric, which may have favorable implications when pursuing implicit temporal integration schemes.

3.6 Note on MBTR remapper implementation

The remapping algorithms presented in the previous section are exposed through a combination of implementations in MOAB and TempestRemap libraries. Since both libraries are written in C++, direct inheritance of key data structures such as the GridElements (mesh) and OfflineMap (projection weights) are available to minimize data movement between the libraries. Additionally, Fortran codes such as E3SM can invoke computations of the intersection mesh and the remapping weights through specialized language-agnostic interfaces in MOAB: `iMOAB` (Mahadevan et al., 2015). These interfaces offer the flexibility to query, manipulate, and transfer the mesh between groups of processes that represent the component and coupler processing elements.

Using the `iMOAB` interfaces, the E3SM coupler can coordinate the online remapping workflow during the setup phase of the simulation and compute the projection operators for component and scalar or vector coupled field combinations. For each pair of coupled components, the following sequence of steps are then executed to consistently compute the remapping operator and transfer the solution fields in parallel.

1. `iMOAB_SendMesh` and `iMOAB_ReceiveMesh`: send the component mesh (defined on $N_{c,1}$ processes) and receive the complete unstructured mesh copy in the coupler processes (N_x). This mesh migration undergoes an online mesh repartition either through a trivial decomposition scheme or with advanced Zoltan algorithms (geometric or graph partitioners).

2. `iMOAB_ComputeMeshIntersectionOnSphere`: the advancing-front intersection scheme is invoked to compute the overlap mesh in the coupler processes.

3. `iMOAB_CoverageGraph`: update the parallel communication graph based on the (source) coverage mesh association in each process.

4. `iMOAB_ComputeScalarProjectionWeights`: the remapping weight operator is computed and assembled with discretization-specific (FV, SE) calls to

TempestRemap and stored in Eigen3 SparseMatrix object.

Once the remapping operator is serialized in memory for each coupled scalar and flux field, this operator is then used at every time step to compute the actual projection of the data.

1. `iMOAB_SendElementTag` and `iMOAB_ReceiveElementTag`: using the coverage graph computed previously, direct one-to-one communication of the field data is enabled between $N_{\mathrm{c,l}}$ and N_x, before and after application of the weight operator.

2. `iMOAB_ApplyScalarProjectionWeights`: in order to compute the field interpolation or projection from the source component to the target component, a matvec product of the weight matrix and the field vector defined on the source grid is performed. The source field vector is received from source processes $N_{\mathrm{c,s}}$ and after weight application, the target field vector is sent to target processes $N_{\mathrm{c,l}}$.

Additionally, to facilitate offline generation of projection weights, a MOAB-based parallel tool `mbtempest` has been written in C++, similar to ESMF and TempestRemap (serial) stand-alone tools. `mbtempest` can load the source and target meshes from files, in parallel, and compute the intersection and remapping weights through TempestRemap. The weights can then be written back to a SCRIP-compatible file format, for any of the supported field discretization combinations in source and destination components. Added capability to apply the weight matrix onto the source solution field vectors, and native visualization plugins in VisIt for MOAB, simplify the verification of conservation and monotonicity for complex remapping workflows. This workflow allows users to validate the underlying assumptions for remapping solution fields across unstructured grids and can be executed in both a serial and a parallel setting.

4 Results

Evaluating the performance of the in-memory MBTR remapping infrastructure requires recursive profiling and optimization to ensure scalability for large-scale simulations. In order to showcase the advantage of using the mesh-aware MOAB data structure as the MCT coupler replacement, we need to understand the per task performance of the regridder in addition to the parallel point locator scalability and overall time for remapping weight computation. Note that except for the weight application for each solution field from a source grid to a target grid, the in-memory copy of the component meshes, migration to coupler PEs, computation of intersection elements, and remapping weights is done only once during the setup phase in E3SM, per coupled component model pair.

4.1 Serial performance

We compare the total cost for computing the supermesh and the remapping weights for several source and target grid combinations through three different methods to determine the serial computational complexity.

1. ESMF: Kd-tree-based regridder and weight generation for first-/second-order FV→FV conservative remapping.

2. TempestRemap: Kd-tree-based supermesh generation and conservative, monotonic, high-order remap operator for FV→FV, SE→FV, SE→SE projection.

3. MBTempest: advancing-front intersection with MOAB and conservative weight generation with TempestRemap interfaces.

Figure 8 shows the serial performance of the remappers for computing the conservative interpolator from cubed-sphere (CS) grids to polygonal MPAS grids of different resolutions for a FV→FV field transfer. This total time includes the computation of intersection mesh or supermesh, in addition to the remapping weights with field conservation specifications. These serial runs were executed on a machine with 8× Intel Xeon(R) CPU E7-4820 at 2.00 GHz (total of 64 cores) and 1.47 TB of RAM. As the source grid resolution increases, the advancing-front intersection with linear complexity outperforms the Kd-tree intersection algorithms used by TempestRemap and ESMF. The time spent in the remapping task, including the overlap mesh generation, provides an overall metric on the single task performance when memory bandwidth or communication concerns do not dominate in a parallel run. In this comparison with three remapping software libraries, the total computational time in the fine-resolution limit as $\frac{\mathrm{nele(source)}}{\mathrm{nele(target)}} \approx 1$ consistently increases (going diagonally from left to right in Fig. 8). We note that the serial version of TempestRemap is comparable to ESMF and can even provide better timing on the highly refined cases, while the MBTempest remapper consistently outperforms both tools, with a 2× speedup on average. The relatively better performance in MBTempest is accomplished through the linear complexity advancing-front algorithm, which further offers avenues to incorporate finer-grain task or thread-level parallelism to accelerate the on-node performance on multicore and general purpose graphical processing unit (GPGPU) architectures.

4.2 Scalability of the MOAB Kd-tree point locator

In addition to being able to compute the supermesh between Ω_{S} and Ω_{T}, MOAB also offers data structures to query

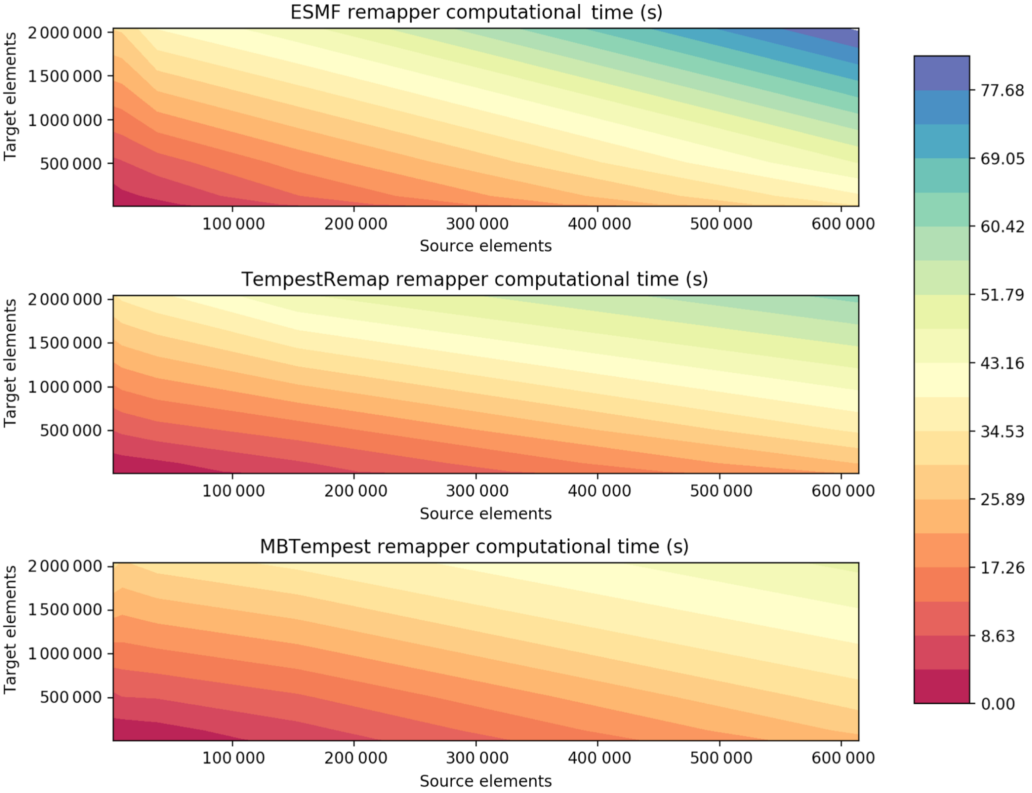


Figure 8. Comparison of serial regridding computation (supermesh and projection weight generation) between ESMF, TempestRemap, and MBTempest.

source elements containing points that correspond to the target DoFs locations. This operation is critical in evaluating bilinear and biquadratic interpolator approximations for scalar variables when conservative projection is not required by the underlying coupled model. The solution interpolation for the multi-mesh case involves two distinct phases.

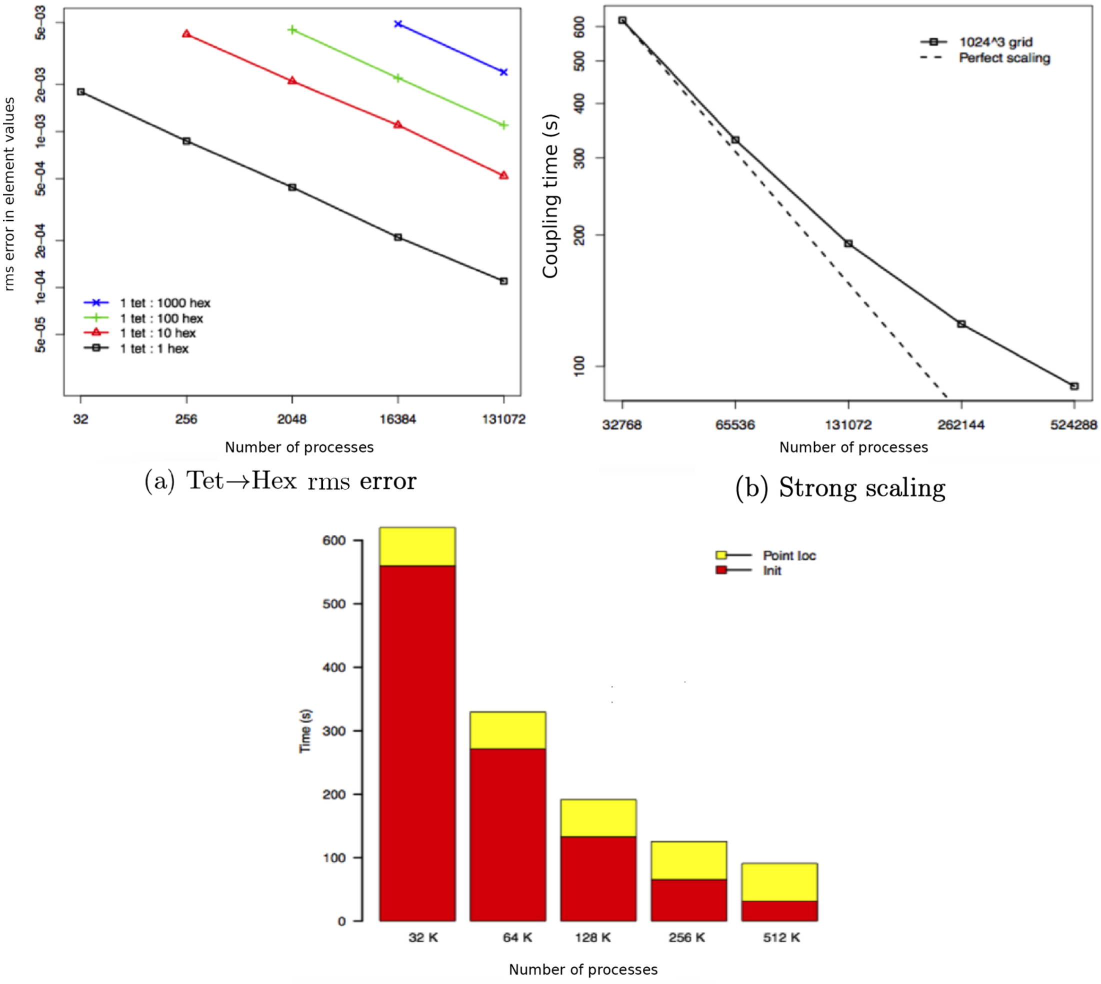
1. Setup phase: use Kd tree to build the search data structure to locate points corresponding to vertices in the target mesh on the source mesh.

2. Run phase: use the elements containing the located points to compute consistent interpolation onto target mesh vertices.

Studies were performed to evaluate the strong and weak scalability of the parallel Kd-tree point search implementation in MOAB. The scalability results were generated with the CIAN2 (https://github.com/tpeterka/cian2, last access: 16 May 2020) coupling mini-app (Morozov and Peterka, 2016), which links to MOAB to handle traversal of the unstructured grids and transfer of solution fields between the grids. For this case, a series of hexahedral and tetrahedral meshes was used to interpolate an analytical solution. By changing the basis interpolation order, and mesh resolutions, the convergence of the interpolator was verified to provide theoretical accuracy orders of convergence in the asymptotic fine limit.

The performance tests were executed on the IBM Blue-Gene/Q Mira at 16 MPI ranks per node, with 2 GB RAM per MPI rank, at up to 500 K MPI processes. The strong scaling results and error convergence were computed with a grid size of 1024^3. The solution interpolation on varying mesh resolutions was performed by projecting an analytical solution from a tetrahedral to a hexahedral to a tetrahedral grid, with total number of points/rank varied between [2 K, 32 K] in the study. Note that the total number of DoFs in this study is much larger than that in typical climate production runs, and hence we use these experiments to showcase the strong scaling of the bilinear interpolation operation at the high-resolution limit.

First, the root-mean-square (rms) error was measured in the bilinearly interpolated solution against the analytical solution and plotted for different source and target mesh resolutions. Figure 9a demonstrates that the error convergence of the interpolants matches the expected theoretical second-order rates, and that the error constant is proportional to ratio of source-to-target mesh resolution. Next, Fig. 9b shows the strong scaling efficiency of around 50 % is achieved on a maximum of 512 K cores (66 % of Mira). We note that the computational complexity of the Kd-tree data structure



(a) Tet→Hex rms error

(b) Strong scaling

(c) Strong scaling by component

Figure 9. MOAB 3-D Kd-tree point location: strong scaling on Mira (BG/Q).

scales as $O(n\log(n))$ asymptotically, and the point location phase during initial search setup dominates the total cost on higher core counts. This is evident in the timing breakdown for each phase shown in Fig. 9c. Since the point location is performed only once during simulation startup, while the interpolation is performed multiple times per time step during the run, we expect the total cost of the projection for scalar variables to be amortized over transient climate simulations with fixed grids. Further investigations with optimal BVH-tree (Larsen et al., 1999) or R-tree implementations for these interpolation cases could help reduce the overall cost.

The full 3-D point location and interpolation operations provided by MOAB are comparable to the implementation in the common remapping component used in the C-Coupler (Liu et al., 2013) and provide relatively much stronger scalability on larger core counts (Liu et al., 2014) for the remapping operation. Such higher-order interpolators for multicomponent physics variables can provide better performance in atmospheric chemistry calculations. Additionally, as component mesh resolutions are increased to sub-kilometer regimes, the expectations from remapping libraries such as MOAB to provide scalable search and location of points become important. Currently, only the NC bilinear or biquadratic interpolation of scalar fields with subset normalization (Tautges and Caceres, 2009) is supported directly in MOAB (via Kd-tree point location and interpolation), and the advancing-front intersection algorithm does not make use of these data structures. In contrast, TempestRemap and ESMF use a Kd-tree search to not only compute the location of points but also to evaluate the supermesh $\Omega_S \bigcup \Omega_T$, and hence the computational complexity for the intersection mesh determination scales as $O(n\log(n))$, in contrast to the linear complexity ($O(n)$) of the advancing-front intersection algorithm implemented in MOAB.

4.3 The parallel MBTR remapping algorithm

The MBTR online weight generation workflow within E3SM was employed to verify and test the projection of real simulation data generated during the coupled atmosphere–ocean model runs. A choice was made to use the model-computed temperature on the lowest level of the atmosphere, since the heat fluxes that nonlinearly couple the atmosphere and ocean models are directly proportional to this interface temperature field. By convention, the fluxes are computed on the ocean mesh, and hence the atmosphere temperature must be interpolated onto MPAS polygonal mesh. We use this scenario as a test case for demonstrating the strong scalability results in this section.

The atmosphere run with approximately 4° grid size and 11 elements per edge on a cubed sphere (NE11) in E3SM, and the projection of its lowest level temperature onto two different MPAS meshes (with approximate grid size of 240 km) are shown in Fig. 10. The conservative field projection from SE to FV on a mesh with holes corresponding to land regions is given in Fig. 10b, where the continents are shown in transparent shading. To contrast, we also present the remapped field on an MPAS mesh without holes (Fig. 10c) to show the differences in the remapped solutions as a function of mesh topology.

4.3.1 Scaling comparison of conservative remappers (FV → FV)

The strong scaling studies for computation of remapping weights to project a FV solution field between CS grids of varying resolutions were performed on the Blues large-scale cluster (with 16 Sandy Bridge Xeon E5-2670 2.6 GHz cores and 32 GB RAM per node) at ANL and the Cori supercomputer at NERSC (with 64 Haswell Xeon E5-2698v3 2.3 GHz cores and 128 GB RAM per node). Figure 11 shows that the MBTR workflow consistently outperforms ESMF on both machines as the number of processes used by the coupler is increased. The timing shown here represents the total remapping time, i.e., cumulative computational time for generating the super mesh and the (conservative) remapping weights.

The relatively better scaling for MOAB on the Blues cluster is due to faster hardware and memory bandwidth compared to the Cori machine. The strong scaling efficiency approaches a plateau on Cori Haswell nodes as communication costs for the coverage mesh computation start dominating the overall remapping processes, especially in the limit of $\frac{\text{nele}}{\text{process}} \rightarrow 1$ at large node counts.

4.3.2 Strong scalability of spectral projection (SE→FV)

To further evaluate the characteristics of in-memory remapping computation, along with cost of application of the weights during a transient simulation, a series of further studies was executed on the NERSC Cori system to determine the spectral projection of a real dataset between atmosphere and ocean components in E3SM. The source mesh contains fourth-order spectral element temperature data defined on Gauss–Lobatto quadrature nodes (cGLL discretization) of the CS mesh, and the projection is performed on a MPAS polygonal mesh with holes (FV discretization). A direct comparison to ESMF was unfeasible in this study since the traditional workflow requires the computation of a dual mesh transformation of the spectral grid. Hence, only timing for MBTR workflow is shown here.

Two specific cases were considered for this SE→FV strong scaling study with conservation and monotonicity constraints.

1. Case A (NE30): 1° CS (30 edges per side) SE mesh (nele=5400 quads) with $p=4$ to MPAS mesh (nele=235 160 polygons).
2. Case B (NE120): 0.25° CS (120 edges per side) SE mesh (nele=86 400 quads) with $p=4$ to MPAS mesh (nele=3 693 225 polygons).

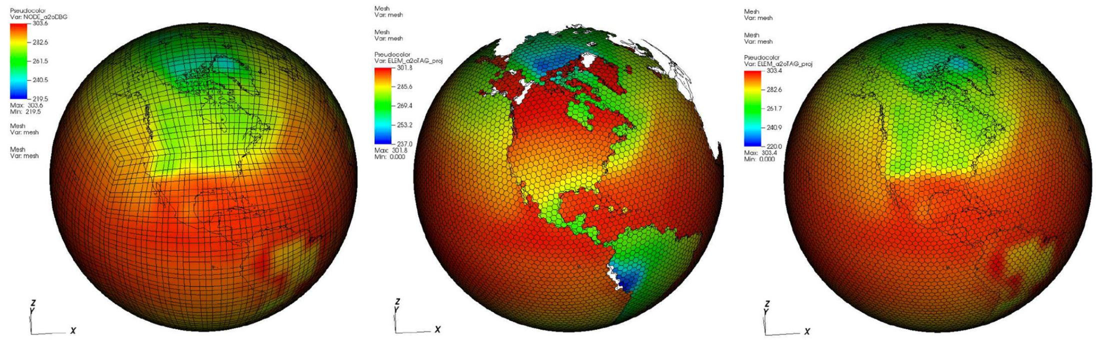
The performance tests for each of these cases were launched with three different process execution layouts for the atmosphere, ocean components, and the coupler.

a. Fully colocated PE layout: $N_{\text{atm}} = N_x$ and $N_{\text{ocn}} = N_x$.

b. Disjoint-ATM model PE layout: $N_{\text{atm}} = N_x/2$ and $N_{\text{ocn}} = N_x$.

c. Disjoint-OCN model PE layout: $N_{\text{atm}} = N_x$ and $N_{\text{ocn}} = N_x/2$.

A breakdown of computational time for key tasks on Cori with up to 1024 processes for both cases is tabulated in Table 1 on a fully colocated decomposition; i.e., $N_{\text{ocn}} = N_{\text{atm}} = N_x$. It is clear that the computation of parallel intersection mesh strong scales well for these production cases, especially for larger mesh resolutions (Case B). For the smaller source and target mesh resolution (Case A), we notice that the intersection time hits a lower bound that is dominated by the computation of the coverage mesh to enclose the target mesh in each task. It is important to stress that this one-time setup call to compute remap operator, per component pair, is relatively much cheaper compared to individual component and solver initializations and gets amortized over longer transient simulations. It is also worth noting that as the I/O bandwidth in emerging architectures is not scaling in line with the compute throughput, such an online workflow can generally be faster than parallel I/O for reading the weights from file at scale. The MBTR implementation is also flexible to allow loading the weights from file directly in order to preserve the existing coupler process with MCT. In comparison to the computation of the intersection mesh, the time to assemble the remapping weight operator in parallel is generally



(a) Original SE field on CS atmosphere mesh

(b) Remapped field on MPAS mesh with holes

(c) Remapped field on MPAS mesh without holes

Figure 10. Projection of the NE11 SE bottom atmospheric temperature field onto the MPAS ocean grid.

Table 1. Strong scaling on Cori for SE→FV projection with two different resolutions on a fully colocated PE layout.

Number of processors	Case A (NE30)		Case B (NE120)	
	Intersection (s)	Compute weights (s)	Intersection (s)	Compute weights (s)
16	0.936846	0.64983	145.623	9.732
32	0.449022	0.429028	53.1244	5.78093
64	0.377767	0.373476	22.7167	4.92151
128	0.255154	0.270574	6.70485	2.79397
256	0.180136	0.18272	2.26435	1.71835
512	0.162388	0.104737	1.25471	0.928622
1024	0.203354	0.0932475	0.680122	0.618943

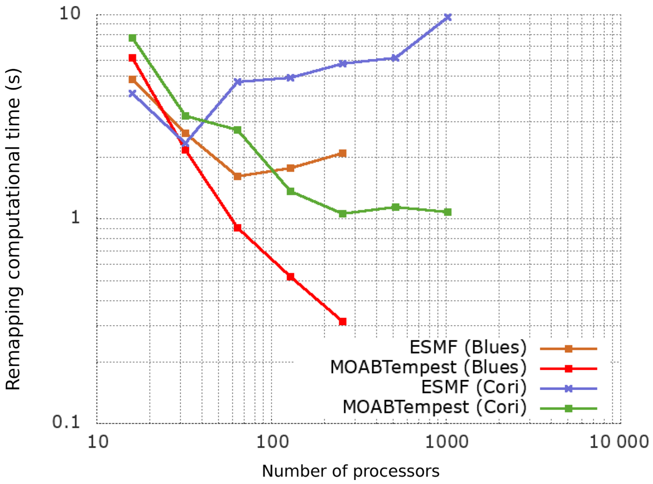
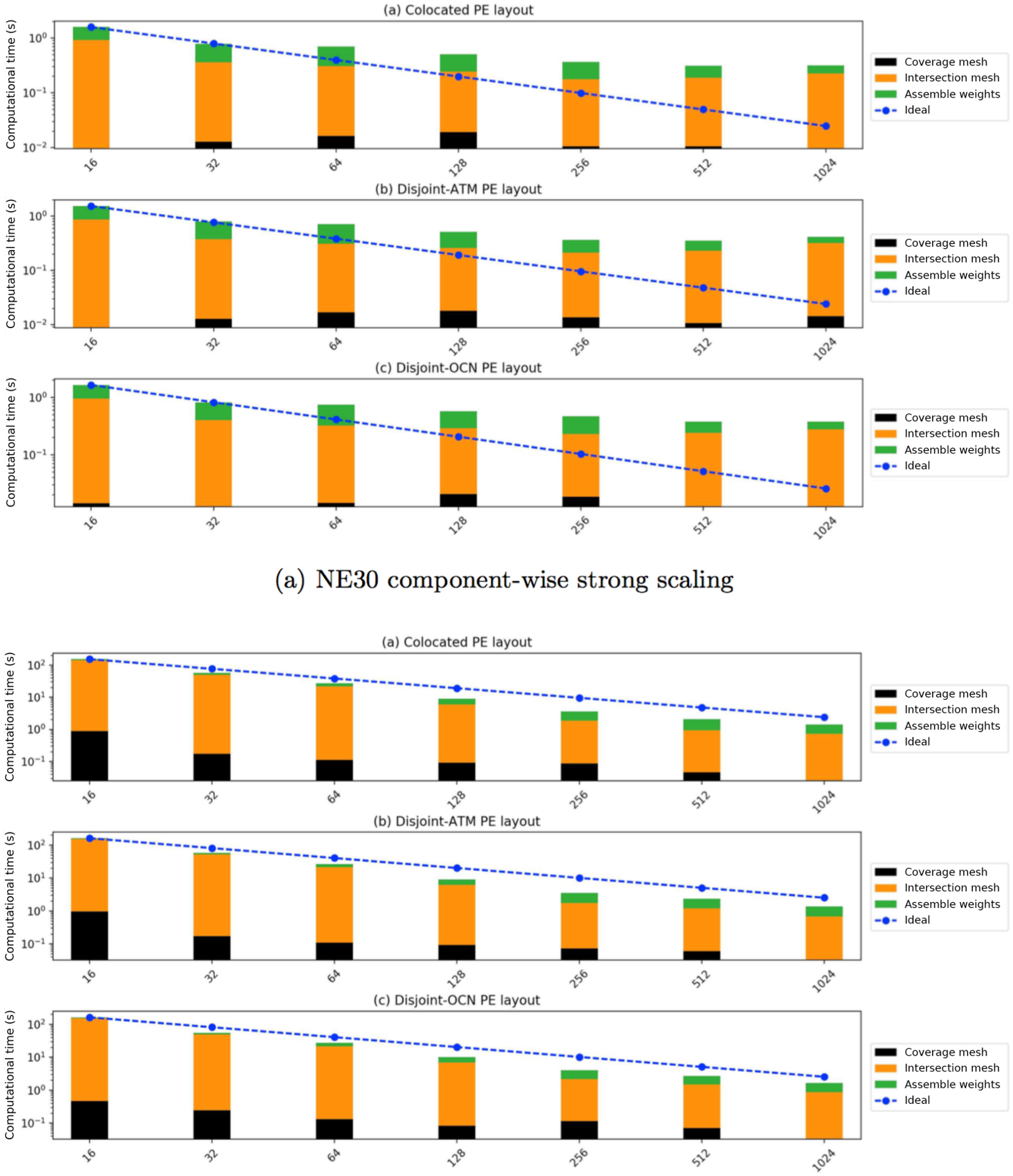


Figure 11. CS ($E = 614\,400$ quads) → CS ($E = 153\,600$ quads) remapping (-m conserve) on LCRC/ALCF and NERSC machines.

smaller. Even though both of these operations are performed only once during the setup phase of the E3SM simulation, the weight operator computation involves several validation checks that utilize collective MPI operations, which do destroy the embarrassingly parallel nature of the calculation, once appropriate coverage mesh is determined in each task.

The component-wise breakdown for the advancing-front intersection mesh, the parallel communication graph for sending and receiving data between component and coupler, and finally, the remapping weight generation for the SE→FV setup for NE30 and NE120 cases is shown in Fig. 12. The cumulative time for this remapping process is shown to scale linearly for the NE120 case, even if the parallel efficiency decreases significantly in the NE30 case, as expected based on the results in Table 1. Note that the MBTR workflow provides a unique capability to consistently and accurately compute SE→FV projection weights in parallel, without any need for an external preprocessing step to compute the dual mesh (as required by ESMF) or running the entire remapping process in serial (TempestRemap).

Another key aspect of the results shown in Fig. 12 is the relative indifference in performance of the algorithms to the type of PE layout used to partition the component process space. Theoretically, we expect the fully disjoint case to perform the worst, and a full colocated case with maximal overlap to perform the best, since the layout directly affects the total amount of data communicated for both the mesh and field data. However, in practice, with online repartitioning



(a) NE30 component-wise strong scaling

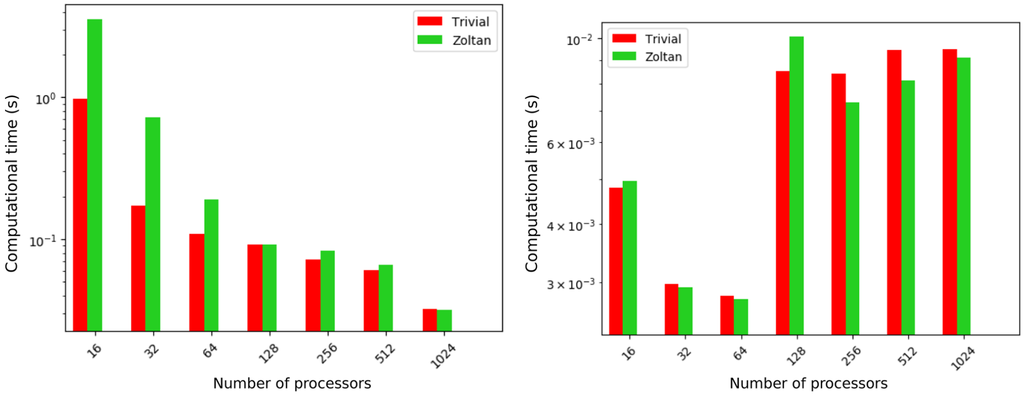
(b) NE120 component-wise strong scaling

Figure 12. Strong scaling study for the NE30 and NE120 cases for spectral projection with Zoltan repartitioner on Cori.

strategies exposed through Zoltan (PHG and RCB), overall scaling of the remapping algorithm is nearly independent of the PE layout for the simulation. This is especially evident from the timing for the coverage mesh computation for the NE120 case for all three PE layouts.

4.4 Effect of partitioning strategy

In order to determine the effect of partitioning strategies described in Fig. 5, the NE120 case with the trivial decomposition and Zoltan geometric partitioner (RCB) was tested in parallel. Figure 13 compares the two strategies for optimizing the mesh migration from the component to coupler. These strategies play a critical role in task mapping and data locality for the source coverage mesh computation, in addition to determining the communication graph complexity between the components and the coupler. This comparison highlights that the coverage mesh cost reduces uniformly at scale, while the trivial partitioning scheme behaves better on lower core counts as shown in Fig. 13a. The communication of field data between the atmosphere component and the coupler resulting from the partitioning strategy is a critical operation during the transient simulation and generally stays within network latency limits in Cori (shown in Fig. 13b).



(a) Source coverage mesh computation

(b) Send/ receive field data between N_x and N_{atm}

Figure 13. Scaling of the communication kernels driven with the parallel graph computed with a trivial redistribution and the Zoltan geometric (RCB) repartitioner for the NE120 case with $N_{\text{ocn}} = N_x$ and $N_{\text{atm}} = N_x/2$ on Cori.

Even though the communication kernel does not show ideal scaling on increasing node counts, the relative cost of the operation should be insignificant in comparison to total time spent in individual component solvers. Note that production climate model solvers require multiple data fields to be remapped at every rendezvous time step, and hence the size of the packed messages may be larger for such simulations. We also note that there is a factor of 3 increase in the communication time to send and receive data, which occurs after the 64 process counts on Cori in Fig. 13b. This is an artifact of the additional communication latency due to the transition from an intra-node (each Haswell node in Cori accommodates 64 processes) to inter-node nearest-neighbor data transfer when using multiple nodes. In this strong scaling study, the net data size being transferred reduces with increasing core counts, and hence the point-to-point communication beyond 128 processes is primarily dominated by network latency and task placement. As part of future extensions, we will further explore task mapping strategies with the Zoltan2 (Leung et al., 2014) library, in addition to online partition rebalancing to maximize geometric overlap and minimize communication time during remapping.

4.5 Note on application of weights

Generally, operations involving SpMV products are memory bandwidth limited (Bell and Garland, 2009) and occur during the application of remapping weights operator onto the source solution field vector, in order to compute the field projection onto the target grid. In addition to the communication of field data shown in Fig. 13b, the cost of remapping weight application in parallel (presented in Fig. 14) determines the total cost of the remapping operation during runtime. Except for the case of cGLL target discretizations, the parallel SpMV operation during the weight application does not involve any global collective reductions. In the current E3SM and OASIS3-MCT workflow, these operations are handled by the MCT library. In high-resolution simulations of E3SM, the total time for the remapping operation in MCT is primarily dominated by the communication costs based on the communication graph, similar to the MBTR workflow. However, a direct comparison of the communication kernels in these two workflows is not yet possible, since the offline maps for MCT that are generated with ESMF use the “dual” grid, while the online maps generated with MBTR utilize the original spectral grid with no approximations, which results in very different communication graph and non-zero pattern in the remap weight matrices.

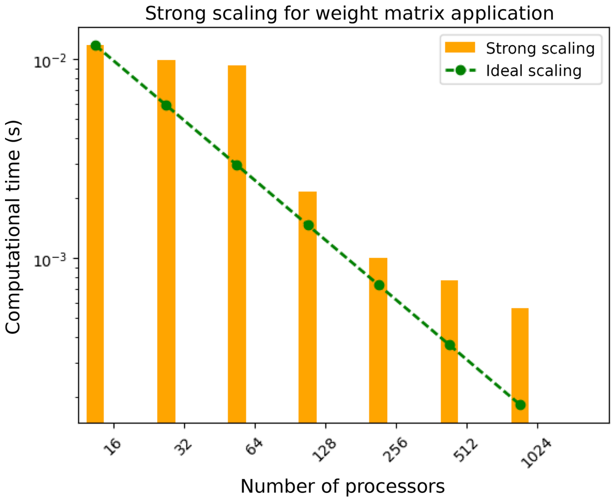


Figure 14. SE→FV remapping weight operator application for the NE120 case on Cori.

5 Conclusions

Understanding and controlling primary sources of errors in a coupled system dynamically will be key to achieving predictable and verifiable climate simulations on emerging architectures. Traditionally, the computational workflow for coupled climate simulations has involved two distinct steps, with an offline preprocessing phase using remapping tools to generate solution field projection weights (ESMF, TempestRemap, SCRIP), which are then consumed by the coupler to transfer field data between the component grids.

The offline steps include generating grid description files and running the offline tools with the problem-specific options. Additionally many of state-of-the-science tools such as ESMF and SCRIP require additional steps to specially handle interpolators from SE grids. Such workflows create bottlenecks that do not scale and can inhibit scientific research productivity. When experimenting with refined grids, a goal for E3SM, this tool chain has to exercised repeatedly. Additionally, when component meshes are dynamically modified, either through mesh adaptivity or dynamical mesh movement to track moving boundaries, the underlying remapping weights must be recomputed on the fly.

To overcome some of these limitations, we have presented scalable algorithms and software interfaces to create a direct component coupling with online regridding and weight generation tools. The remapping algorithms utilize the numerics exposed by TempestRemap and leverage the parallel mesh handling infrastructure in MOAB to create a scalable inmemory remapping infrastructure that can be integrated with existing coupled climate solvers. Such a methodology invalidates the need for dual grids, preserves higher-order spectral accuracy, and locally conserves the field data, in addition to monotonicity constraints, when transferring solutions between grids with non-matching resolutions.

The serial and parallel performance of the MOAB advancing-front intersection algorithm with linear complexity ($O(n)$) was demonstrated for a variety of source and target mesh resolution combinations and compared with the current state-of-the-science regridding tools such as ESMF (serial/parallel) and TempestRemap (serial) that have a $O(n\log(n))$ complexity using the Kd-tree data structure. The MOAB-TempestRemap (MBTR) software infrastructure yields a balance of the scalable performance on emerging architectures without sacrificing discretization accuracy for component field interpolators. There are also several optimizations in the MBTR algorithms that can be implemented to improve finer-grained parallelism on heterogeneous architectures and to minimize data movement with better partitioning in combination with load rebalancing strategies. Such a software infrastructure provides a foundation to build a new coupler to replace the current offline–online, hub-and-spoke MCT-based coupler in E3SM and offers extensions to enable a fully distributed coupling paradigm (without the need for a centralized coupler) to minimize computational bottlenecks in a task-based workflow.

Code availability. Information on the availability of source code for the algorithmic infrastructure and models featured in this paper is tabulated below.

- E3SM (E3SM Project, 2018) is under active development funded by the US Department of Energy. E3SM version 1.1 has been publicly released under an open-source three-clause BSD license in August 2018 and is available on GitHub (https://github.com/E3SM-Project/E3SM, last access: 16 May 2020, E3SM Project, 2018).
- MOAB (Tautges et al., 2004) is an open-source library under the umbrella of the SIGMA toolkit (2014) (Mahadevan et al., 2015) and is publicly available under the Lesser GNU Public License (v3) on BitBucket https://bitbucket.org/fathomteam/moab (last access: 16 May 2020, Tautges et al., 2004). Version 5.1.0 was released on 7 January 2019 and available at http://ftp.mcs.anl.gov/pub/fathom/moab-5.1.0.tar.gz (last access: 16 May 2020). DOI: https://doi.org/10.5281/zenodo.2584863 (Mahadevan er al., 2020).
- TempestRemap (Ullrich and Taylor, 2015; Ullrich et al., 2016) source code is available under a BSD open-source license and hosted on GitHub (https://github.com/ClimateGlobalChange/tempestremap, last access: 16 May 2020, Ullrich and Mahadevan, 2020). Version 2.0.2 was released on 19 December 2018 and available at https://github.com/ClimateGlobalChange/tempestremap/archive/v2.0.2.tar.gz (last access: 16 May 2020).

Video supplement. The video supplements for the serial and parallel advancing-front mesh intersection algorithms to compute the supermesh ($\boldsymbol{\Omega}_{\mathrm{S}} \bigcup \boldsymbol{\Omega}_{\mathrm{T}}$) of a source ($\boldsymbol{\Omega}_{\mathrm{S}}$) and target ($\boldsymbol{\Omega}_{\mathrm{T}}$) grid are demonstrated.

- Serial advancing-front mesh intersection: intersection between CS and MPAS grids on a single task is illustrated; https://doi.org/10.6084/m9.figshare.7294901.v1 (Mahadevan et al., 2018a).
- Parallel advancing-front mesh intersection: simultaneous parallel intersections between CS and MPAS grids on two different tasks are illustrated side by side; https://doi.org/10.6084/m9.figshare.7294919.v2 (Mahadevan et al., 2018b).

Author contributions. VSM and RJ wrote the paper (with comments from IG and JS). VSM and IG designed and implemented the MOAB integration with TempestRemap library, along with exposing the necessary infrastructure for online remapping through iMOAB interfaces. IG and JS configured the MOAB-TempestRemap remapper within E3SM and verified weight generation to transfer solution fields between atmosphere and ocean component models. VSM conducted numerical verification studies and executed both the serial and parallel scalability studies on Blues and Cori LCF machines to quantify performance characteristics of

the remapping algorithms. The broader project idea was conceived by Andy Salinger (SNL), RJ, VSM, and IG.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the US Department of Energy under contract DEAC02-06CH11357 and resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the US Department of Energy under contract no. DEAC02-05CH11231. We gratefully acknowledge the computing resources provided on Blues, a high-performance computing cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory. We would also like to thank Dr. Paul Ullrich at University of California, Davis, for several helpful discussions regarding remapping schemes and implementations in TempestRemap.

Financial support. This research has been supported by the Department of Energy, Labor and Economic Growth (grant no. FWP 66368). Funding for this work was provided by the Climate Model Development and Validation – Software Modernization project, and partially by the SciDAC Coupling Approaches for Next Generation Architectures (CANGA) project, which is funded by the US Department of Energy (DOE) and Office of Science Biological and Environmental Research programs. CANGA is also funded by the DOE Office of Advanced Scientific Computing Research (ASCR).

Review statement. This paper was edited by Sophie Valcke and reviewed by two anonymous referees.

References

Aguerre, H. J., Damián, S. M., Gimenez, J. M., and Nigro, N. M.: Conservative handling of arbitrary non-conformal interfaces using an efficient supermesh, J. Comput. Phys., 335, 21–49, 2017.

Beljaars, A., Dutra, E., Balsamo, G., and Lemarié, F.: On the numerical stability of surface–atmosphere coupling in weather and climate models, Geosci. Model Dev., 10, 977–989, https://doi.org/10.5194/gmd-10-977-2017, 2017.

Bell, N. and Garland, M.: Implementing sparse matrix-vector multiplication on throughput-oriented processors, in: Proceedings of the conference on high performance computing networking, storage and analysis, p. 18, ACM, 2009.

Berger, M. J.: On conservation at grid interfaces, SIAM J. Numer. Anal., 24, 967–984, 1987.

Blanco, J. L. and Rai, P. K.: nanoflann: a C++ header-only fork of FLANN, a library for Nearest Neighbor (NN) wih KD-trees, available at: https://github.com/jlblancoc/nanoflann (last access: 16 May 2020), 2014.

Březina, J. and Exner, P.: Fast algorithms for intersection of non-matching grids using Plücker coordinates, Comput. Math. Appl., 74, 174–187, https://doi.org/10.1016/j.camwa.2017.01.028, 2017.

Certik, O., Ferenbaugh, C., Garimella, R., Herring, A., Jean, B., Malone, C., and Sewell, C.: A Flexible Conservative Remapping Framework for Exascale Computing, available at: https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-17-21749 (last access: 16 May 2020), sIAM Conference on Computational Science & Engineering, February 2017.

Collins, N., Theurich, G., Deluca, C., Suarez, M., Trayanov, A., Balaji, V., Li, P., Yang, W., Hill, C., and Da Silva, A.: Design and implementation of components in the Earth System Modeling Framework, Int. J. High Perform. C., 19, 341–350, 2005.

Craig, A., Valcke, S., and Coquart, L.: Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0, Geosci. Model Dev., 10, 3297–3308, https://doi.org/10.5194/gmd-10-3297-2017, 2017.

Craig, A. P., Jacob, R., Kauffman, B., Bettge, T., Larson, J., Ong, E., Ding, C., and He, Y.: CPL6: The New Extensible, High Performance Parallel Coupler for the Community Climate System Model, Int. J. High Perform. C., 19, 309–327, https://doi.org/10.1177/1094342005056117, 2005.

Craig, A. P., Vertenstein, M., and Jacob, R.: A new flexible coupler for earth system modeling developed for CCSM4 and CESM1, Int. J. High Perform. C., 26, 31–42, https://doi.org/10.1177/1094342011428141, 2012.

de Boer, A., van Zuijlen, A., and Bijl, H.: Comparison of conservative and consistent approaches for the coupling of non-matching meshes, Comput. Method. Appl. M., 197, 4284–4297, https://doi.org/10.1016/j.cma.2008.05.001, 2008.

Dennis, J. M., Edwards, J., Evans, K. J., Guba, O., Lauritzen, P. H., Mirin, A. A., St-Cyr, A., Taylor, M. A., and Worley, P. H.: CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model, Int. J. High Perform. C., 26, 74–89, 2012.

Devine, K., Boman, E., Heaphy, R., Hendrickson, B., and Vaughan, C.: Zoltan Data Management Services for Parallel Dynamic Applications, Comput. Sci. Eng., 4, 90–97, 2002.

Dunlap, R., Rugaber, S., and Mark, L.: A feature model of coupling technologies for Earth System Models, Comput. Geosci., 53, 13–20, 2013.

E3SM Project: Energy Exascale Earth System Model (E3SM), [Computer Software], https://doi.org/10.11578/E3SM/dc.20180418.36, 2018.

Farrell, P. and Maddison, J.: Conservative interpolation between volume meshes by local Galerkin projection, Comput. Method. Appl. M., 200, 89–100, https://doi.org/10.1016/j.cma.2010.07.015, 2011.

Fleishman, S., Cohen-Or, D., and Silva, C. T.: Robust moving least-squares fitting with sharp features, ACM T. Graphic, 24, 544–552, 2005.

Flyer, N. and Wright, G. B.: Transport schemes on a sphere using radial basis functions, J. Comput. Phys., 226, 1059–1084, 2007.

Fornberg, B. and Piret, C.: On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere, J. Comput. Phys., 227, 2758–2780, 2008.

Fox, G., Johnson, M., Lyzenga, G., Otto, S., Salmon, J., Walker, D., and White, R. L.: Solving Problems On Concurrent Processors Vol. 1: General Techniques and Regular Problems, Comput. Phys., 3, 83–84, 1989.

Gander, M. J. and Japhet, C.: Algorithm 932: PANG: software for nonmatching grid projections in 2D and 3D with linear complexity, ACM T. Math. Software, 40, 1–25, https://doi.org/10.1145/2513109.2513115, 2013.

Gottlieb, D. and Shu, C.-W.: On the Gibbs phenomenon and its resolution, SIAM Rev., 39, 644–668, 1997.

Grandy, J.: Conservative remapping and region overlays by intersecting arbitrary polyhedra, J. Comput. Phys., 148, 433–466, 1999.

Guennebaud, G., Jacob, B., et al.: Eigen v3, available at: http://eigen.tuxfamily.org (last access: 16 May 2020), 2010.

Hanke, M., Redler, R., Holfeld, T., and Yastremsky, M.: YAC 1.2.0: new aspects for coupling software in Earth system modelling, Geosci. Model Dev., 9, 2755–2769, https://doi.org/10.5194/gmd-9-2755-2016, 2016.

Herring, A. M., Certik, O., Ferenbaugh, C. R., Garimella, R. V., Jean, B. A., Malone, C. M., and Sewell, C. M.: (U) Introduction to Portage, Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), available at: https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-17-20831 (last access: 16 May 2020), 2017.

Hill, C., DeLuca, C., Balaji, Suarez, M., and Silva, A. D.: The architecture of the earth system modeling framework, Comput. Sci. Eng., 6, 18–28, 2004.

Hunt, W., Mark, W. R., and Stoll, G.: Fast kd-tree construction with an adaptive error-bounded heuristic, in: Interactive Ray Tracing 2006, 81–88, 2006.

Hurrell, J. W., Holland, M. M., Gent, P. R., Ghan, S., Kay, J. E., Kushner, P. J., Lamarque, J.-F., Large, W. G., Lawrence, D., Lindsay, K., Lipscomb, W. H., Long, M. C., Mahowald, N., Marsh, D. R., Neale, R. B., Rasch, P., Vavrus, S., Vertenstein, M., Bader, D., Collins, W. D., Hack, J. J., Kiehl, J., and Marshall, S. L.: The community earth system model: a framework for collaborative research, B. Am. Meteorol. Soc., 94, 1339–1360, 2013.

Ize, T., Wald, I., and Parker, S. G.: Asynchronous BVH construction for ray tracing dynamic scenes on parallel multi-core architectures, in: Proceedings of the 7th Eurographics conference on Parallel Graphics and Visualization, Eurographics Association, 101–108, 2007.

Jacob, R., Larson, J., and Ong, E.: M×N communication and parallel interpolation in Community Climate System Model Version 3 using the model coupling toolkit, Int. J. High Perform. C., 19, 293–307, 2005.

Jiao, X. and Heath, M. T.: Common-refinement-based data transfer between non-matching meshes in multiphysics simulations, Int. J. Numer. Meth. Eng., 61, 2402–2427, 2004.

Jones, P. W.: First-and second-order conservative remapping schemes for grids in spherical coordinates, Mon. Weather Rev., 127, 2204–2210, 1999.

Karypis, G. and Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput., 20, 359–392, 1998.

Karypis, G., Schloegel, K., and Kumar, V.: Parmetis: Parallel graph partitioning and sparse matrix ordering library, Version 1.0, Dept. of Computer Science, University of Minnesota, 22 pp., 1997.

Larsen, E., Gottschalk, S., Lin, M. C., and Manocha, D.: Fast proximity queries with swept sphere volumes, Tech. rep., Technical Report TR99-018, Department of Computer Science, University of North Carolina, 1999.

Larson, J. W., Jacob, R. L., Foster, I., and Guo, J.: The model coupling toolkit, in: International Conference on Computational Science, Springer, 185–194, 2001.

Lauritzen, P. H., Nair, R. D., and Ullrich, P. A.: A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid, J. Comput. Phys., 229, 1401–1424, https://doi.org/10.1016/j.jcp.2009.10.036, 2010.

Leung, V. J., Rajamanickam, S., Pedretti, K., Olivier, S. L., Devine, K. D., Deveci, M., Catalyurek, U., and Bunde, D. P.: Zoltan2: Exploiting Geometric Partitioning in Task Mapping for Parallel Computers, Tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2014.

Li, L., Lin, P., Yu, Y., Wang, B., Zhou, T., Liu, L., Liu, J., Bao, Q., Xu, S., Huang, W., Xia, K., Pu, Y., Dong, L., Shen, S., Liu, Y., Hu, N., Liu, M., Sun, W., Shi, X., Zheng, W., Wu, B., Song, M., Liu, H., Zhang, X., Wu, G., Xue, W., Huang, X., Yang, G., Song, Z., and Qiao, F.: The flexible global ocean-atmosphere-land system model, Grid-point Version 2: FGOALS-g2, Adv. Atmos. Sci., 30, 543–560, https://doi.org/10.1007/s00376-012-2140-6, 2013.

Liu, L., Yang, G., and Wang, B.: CoR: a multi-dimensional common remapping software for Earth System Models, in: The Second Workshop on Coupling Technologies for Earth System Models (CW2013), available at: https://wiki.cc.gatech.edu/CW2013/index.php/Program (last access: 8 May 2014), 2013.

Liu, L., Yang, G., Wang, B., Zhang, C., Li, R., Zhang, Z., Ji, Y., and Wang, L.: C-Coupler1: a Chinese community coupler for Earth system modeling, Geosci. Model Dev., 7, 2281–2302, https://doi.org/10.5194/gmd-7-2281-2014, 2014.

Liu, L., Zhang, C., Li, R., Wang, B., and Yang, G.: C-Coupler2: a flexible and user-friendly community coupler for model coupling and nesting, Geosci. Model Dev., 11, 3557–3586, https://doi.org/10.5194/gmd-11-3557-2018, 2018.

Löhner, R.: Recent advances in parallel advancing front grid generation, Arch. Comput. Meth. E., 21, 127–140, 2014.

Löhner, R. and Parikh, P.: Generation of three-dimensional unstructured grids by the advancing-front method, Int. J. Numer. Meth. Fl., 8, 1135–1149, 1988.

Mahadevan, V., Grindeanu, I. R., Ray, N., Jain, R., and Wu, D.: SIGMA Release v1.2 – Capabilities, Enhancements and Fixes, https://doi.org/10.2172/1224985, 2015.

Mahadevan, V., Grindeanu, I., Jacob, R., and Sarich, J.: MOAB: Serial Advancing Front Intersection Computation, https://doi.org/10.6084/m9.figshare.7294901.v1, 2018a.

Mahadevan, V., Grindeanu, I., Jacob, R., and Sarich, J.: MOAB: Parallel Advancing Front Mesh Intersection Algorithm, https://doi.org/10.6084/m9.figshare.7294919.v2, 2018b.

Mahadevan, V., Grindeanu, I., Jain, R., Shriwise, P., and Wilson, P.: MOAB v5.1.0, https://doi.org/10.5281/zenodo.2584863, 2020.

Morozov, D. and Peterka, T.: Block-Parallel Data Analysis with DIY2, 2016 IEEE 6th Symposium on Large Data Analysis and Visualization (LDAV), 29–36, 2016.

Mundt, M., Boslough, M., Taylor, M., and Roesler, E.: A Modification to the remapping of Gauss-Lobatto nodes to the cubed sphere, Tech. rep., Technical Report SAND2016-0830R, Sandia National Laboratory, 2016.

Petersen, M. R., Jacobsen, D. W., Ringler, T. D., Hecht, M. W., and Maltrud, M. E.: Evaluation of the arbitrary Lagrangian–Eulerian vertical coordinate method in the MPAS-Ocean model, Ocean Model., 86, 93–113, 2015.

Plimpton, S. J., Hendrickson, B., and Stewart, J. R.: A parallel rendezvous algorithm for interpolation between multiple grids, J. Parallel Distr. Com., 64, 266–276, 2004.

Powell, D. and Abel, T.: An exact general remeshing scheme applied to physically conservative voxelization, J. Comput. Phys., 297, 340–356, https://doi.org/10.1016/j.jcp.2015.05.022, 2015.

Rančić, M.: An efficient, conservative, monotonic remapping for semi-Lagrangian transport algorithms, Mon. Weather Rev., 123, 1213–1217, 1995.

Reichler, T. and Kim, J.: How well do coupled models simulate today's climate?, B. Am. Meteorol. Soc., 89, 303–312, 2008.

Ringler, T., Petersen, M., Higdon, R. L., Jacobsen, D., Jones, P. W., and Maltrud, M.: A multi-resolution approach to global ocean modeling, Ocean Model., 69, 211–232, 2013.

Schliephake, M. and Laure, E.: Performance Analysis of Irregular Collective Communication with the Crystal Router Algorithm, in: Solving Software Challenges for Exascale, edited by: Markidis, S. and Laure, E., Springer International Publishing, Cham, 130–140, 2015.

Shewchuk, J. R.: Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator, in: Applied computational geometry towards geometric engineering, Springer, 203–222, 1996.

SIGMA toolkit: Scalable Interfaces for Geometry and Mesh based Applications (SIGMA) toolkit, available at: http://sigma.mcs.anl.gov (last access: 16 May 2020), 2014.

Slattery, S., Wilson, P., and Pawlowski, R.: The data transfer kit: a geometric rendezvous-based tool for multiphysics data transfer, in: International conference on mathematics & computational methods applied to nuclear science & engineering (M&C 2013), 5–9, 2013.

Slattery, S. R.: Mesh-free data transfer algorithms for partitioned multiphysics problems: Conservation, accuracy, and parallelism, J. Comput. Phys., 307, 164–188, 2016.

Slingo, J., Bates, K., Nikiforakis, N., Piggott, M., Roberts, M., Shaffrey, L., Stevens, I., Vidale, P. L., and Weller, H.: Developing the next-generation climate system models: challenges and achievements, Philos. T. R. Soc. S-A, 367, 815–831, 2009.

Tautges, T. J. and Caceres, A.: Scalable parallel solution coupling for multiphysics reactor simulation, J. Phys. Conf. Ser., 180, 012017, https://doi.org/10.1088/1742-6596/180/1/012017, 2009.

Tautges, T. J., Meyers, R., Merkley, K., Stimpson, C., and Ernst, C.: MOAB: A Mesh-Oriented datABase, SAND2004-1592, Sandia National Laboratories, 2004.

Tautges, T. J., Kraftcheck, J., Bertram, N., Sachdeva, V., and Magerlein, J.: Mesh Interface Resolution and Ghost Exchange in a Parallel Mesh Representation, IEEE, Shanghai, China, 2012.

Taylor, M., Edwards, J., Thomas, S., and Nair, R.: A mass and energy conserving spectral element atmospheric dynamical core on the cubed-sphere grid, J. Phys. Conf. Ser., 78, p. 012074, https://doi.org/10.1088/1742-6596/78/1/012074, 2007.

Thomas, S. J. and Loft, R. D.: The NCAR spectral element climate dynamical core: Semi-implicit Eulerian formulation, J. Sci. Comput., 25, 307–322, 2005.

Ullrich, P. and Mahadevan, V.: TempestRemap v2.0.2, available at: https://github.com/ClimateGlobalChange/tempestremap, last access: 16 May 2020.

Ullrich, P. A. and Taylor, M. A.: Arbitrary-order conservative and consistent remapping and a theory of linear maps: Part I, Mon. Weather Rev., 143, 2419–2440, 2015.

Ullrich, P. A., Lauritzen, P. H., and Jablonowski, C.: Geometrically Exact Conservative Remapping (GECoRe): Regular latitude–longitude and cubed-sphere grids, Mon. Weather Rev., 137, 1721–1741, 2009.

Ullrich, P. A., Lauritzen, P. H., and Jablonowski, C.: Some considerations for high-order "incremental remap"-based transport schemes: edges, reconstructions, and area integration, Int. J. Nume. Meth. Fl., 71, 1131–1151, 2013.

Ullrich, P. A., Devendran, D., and Johansen, H.: Arbitrary-order conservative and consistent remapping and a theory of linear maps: Part II, Mon. Weather Rev., 144, 1529–1549, 2016.

Valcke, S.: The OASIS3 coupler: a European climate modelling community software, Geosci. Model Dev., 6, 373–388, https://doi.org/10.5194/gmd-6-373-2013, 2013.

Valcke, S., Craig, A., and Coquart, L.: OASIS3-MCT User Guide, OASIS3-MCT4.0, Tech. rep., CECI, Université de Toulouse, CNRS, CERFACS – TR-CMGC-18-77, Toulouse, France, available at: https://cerfacs.fr/wp-content/uploads/2018/07/GLOBC-TR-oasis3mct_UserGuide4.0_30062018.pdf (last access: 19 May 2020), 2018.

VisIt: VisIt User's Guide, Tech. Rep. UCRL-SM-220449, Lawrence Livermore National Laboratory, 2005.

Wan, H., Giorgetta, M. A., Zängl, G., Restelli, M., Majewski, D., Bonaventura, L., Fröhlich, K., Reinert, D., Rípodas, P., Kornblueh, L., and Förstner, J.: The ICON-1.2 hydrostatic atmospheric dynamical core on triangular grids – Part 1: Formulation and performance of the baseline version, Geosci. Model Dev., 6, 735–763, https://doi.org/10.5194/gmd-6-735-2013, 2013.

Washington, W., Bader, D., and Collins, B. E. A.: Challenges in climate change science and the role of computing at the extreme scale, in: Proc. of the Workshop on Climate Science, 2008.

Zhou, S.: Coupling climate models with the earth system modeling framework and the common component architecture, Concurr. Comp.-Pract. E., 18, 203–213, 2006.

Zienkiewicz, O. C. and Zhu, J. Z.: The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique, Int. J. Numer. Meth. Eng., 33, 1331–1364, 1992.