Conservative Field Transfer Functionality in Parallel Coupler for Multimodel Simulations (PCMS)

Abhiyan Paudel

Department of Mechanical, Aerospace, and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180

Cameron Smith

Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180

Jacob S. Merson

Department of Mechanical, Aerospace, and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180

Abstract

PCMS is a coupling framework that offers efficient data transfer, parallel execution control, and field-mapping operations. In multiscale and multiphysics simulations, interpolating discrete fields is frequent requirement, often necessitating the conservation of physical quantities such as mass or energy. This project enhances PCMS field-transfer capabilities by adding a conservative particle-to-mesh mapping method, initially targeting two-dimensional simulations. Its applications span fluid–structure interactions, conjugate heat transfer, electromagnetic–thermal coupling, and fusion-plasma workflows.

A high-performance API functionality is developed to support these conservative field transfers, applying best practices to ensure robustness and maintainability. The development setup uses PETSc for assembly and sparse solvers, MeshField for GPU-optimized storage and handling of unstructured mesh fields and Omega_h for mesh operations. An automated CI/CD pipeline using GitHub Actions streamlines testing and verification. Future work will extend this functionality to higher-order particle-to-mesh schemes and to conservative mesh-to-mesh transfers.

Keywords: particle-to-mesh mapping; conservative interpolation; finite-element Galerkin projection

Conservative Field Transfer Functionality in Parallel Coupler for Multimodel Simulations (PCMS):
Developer's repository link: https://github.com/abhiyanpaudel/CAS_Final_Project
Licensing provisions: BSD 3-clause
Programming language: C++
Supplementary material: None

1. Nature of problem

PCMS is designed to handle data transfer, parallel control, and field transfer operations for multiphysics and multiscale coupling applications. The existing field mapping routine in PCMS [1] relies mainly on radial basis function (RBF) based interpolation [2] that provides flexibility and accuracy for point-to-point transfers across complex geometries with minimal implementation complexity. Even though this method is flexible and has a wide range of applications, it does not account for physics constraints.

But as we know that in many multiphysics and multiscale simulations, conservative data transfer between different discretizations is critical—especially when preserving integral quantities like mass, momentum, and energy, which are essential for maintaining solution fidelity. In tightly coupled applications such as fusion plasma simulations, thermal-hydraulic modeling, and fluid-structure interactions, even minor conservation errors can accumulate over time, potentially resulting in unphysical outcomes, numerical instabilities, and overall degradation of the solution. These inaccuracies effectively introduce artificial sources or sinks, violating core physical laws. This issue becomes particularly severe in simulations where strict conservation is necessary for physical realism. This work aims to address the shortcomings of existing field transfer mechanisms in PCMS. Unlike RBF-based methods, the approach presented here is global in nature.

2. Solution method

The current work deals with the implementation of conservative field transfer method in PCMS. We have developed a conservative particle-to-mesh mapping code in PCMS that guarantees preservation of integral quantities critical to physical simulations. This approach is based on the projection of particle fields onto finite element space. At its core, the approach enforces conservation by formulating the transfer problem as a linear system that explicitly preserves physical quantitites. The conservative mapping is expressed by

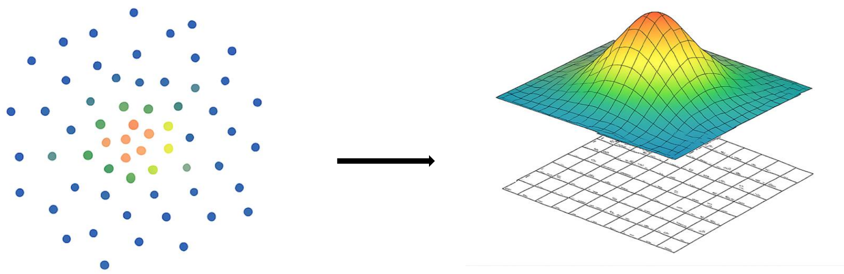

Figure 1: Schematic representation of particle to mesh mapping

the linear equation [3]

$$M\,x \;=\; V\,f$$

where:

- $M_{ij} = \int_\Omega \phi_i\,\phi_j\,d\Omega$ is the finite-element mass matrix.
- $V_{jk} = \phi_j(x_k)$ is the basis-evaluation matrix at the particle positions,

- f is the vector of source (particle) field values, and
- x is the vector of basis functions coefficients

Here, ϕ is the basis function of the target mesh. Following are the steps we take to evaluate the field on the target space.

1. The mass matrix is first computed in CSR (Compressed Sparse Row) format using the `MeshField` [4] library, while `PETSc` [5] is employed to construct the sparse matrix structure.
2. Next, we identify the elements in which the source points are located using `GridPointSearch` class from `PCMS` [6].
3. For each source point, once its containing element is known, we compute the barycentric coordinates using the `Omega_h` library based on its spatial position.
4. The barycentric coordinates provide the values of the basis functions at the nodes of the corresponding element in a linear finite element space.
5. Using atomic operations, we then populate the row pointer array, column index array, and values array, which together define the CSR representation of the matrix V.
6. Following this, we perform matrix-vector multiplications to compute the right-hand side vector.
7. Since both M and V are sparse matrices, we utilize PETSc's parallel sparse solvers to solve for the coefficient vector efficiently.

3. Additional comments

The current implementation supports only two-dimensional meshes with linear (P1) finite elements. Future work will generalize to higher-order (P2, P3) elements and fully three-dimensional domains.

References

[1] Philip E. Davis, Jacob Merson, Pradeep Subedi, Lee Ricketson, Cameron W. Smith, Mark S. Shephard, and Manish Parashar. Benesh: A framework for choreographic coordination of in situ workflows. 2023.

[2] Stuart R. Slattery. Mesh-free data transfer algorithms for partitioned multiphysics problems: Conservation, accuracy, and parallelism. *Journal of Computational Physics*, 307:164–188, February 2016. ISSN 00219991. doi: 10.1016/j.jcp.2015.11.055.

[3] Albert Mollén, M. F. Adams, M. G. Knepley, R. Hager, and C. S. Chang. Implementation of higher-order velocity mapping between marker particles and grid in the particle-in-cell code XGC. *Journal of Plasma Physics*, 87(2):905870229, 2021.

[4] https://github.com/SCOREC/meshFields.

[5] Satish Balay et al. PETSc web page. URL `https://petsc.org/`.

[6] P. Davis, J. Merson, P. Subedi, L. Ricketson, C. Smith, M.S. Shephard, and M. Parashar. Benesh: Choreographic coordination for in-situ workflows. In *30th IEEE International Conference on High Performance Computing, Data, and Analytics (HiPC)*. IEEE, 2023.