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 [1]. 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 [2], 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 [3] for sparse solvers, MeshField [4] for GPU-optimized storage and handling of unstructured mesh fields and Omega_h [5] 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

Multiphysics and multiscale simulations require accurate data transfer between disparate discretizations, where the conservation of integral quantities such as mass, momentum and energy is often essential for solution fidelity. In many coupled physics applications, such as fusion plasma simulations, thermal-hydraulics modeling, and fluid-structure interactions, even small conservation errors can accumulate over time steps, leading to unphysical results, numerical instabilities, and solution degradation. These errors essentially constitute artificial sources or sinks that violate fundamental physical principles.

The existing field-mapping methods in PCMS primarily rely on radial basis function (RBF) based interpolation [6] which provides flexibility and accuracy for point-to-point transfers across complex geometries with minimal implementation complexity. However, RBF methods are fundamentally non-conservative, as they focus on maintaining field continuity rather than integral preservation. This limitation becomes particularly problematic in tightly-coupled simulations where exact conservation is a prerequisite for physical validity.

2. Solution method

In this work, we have implemented a conservative particle-to-mesh mapping technique in PCMS that guarantees preservation of integral quantities critical to physical simulations. 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 the linear equation

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

where:

- M is the finite-element mass matrix
- V 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

We assemble M by aggregating element-level contributions and build V by evaluating basis functions at each particle's location. Both matrices are sparse, and we use PETSc's parallel sparse solvers to compute the coefficient vector x.

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] 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.

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

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

[5] https://github.com/sandialabs/omega_h.

[6] 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.