HARD: Leveraging FleCSI for Heterogeneous Multi-Material Hydrodynamics

Korinna Bayer, Teresa Boland, Arturo Burgos, Seth Hoisington, Timothé Kruk,
Marco Pasquale, Luca Pennati, Andrés Yagüe López, Hyun Lim, and Julien Loiseau
Los Alamos National Laboratory
Los Alamos, NM, USA
{ayaguelopez,hyunlim,jloiseau}@lanl.gov
LA-UR-26-27097

***Abstract*—We present a new multi-material hydrodynamics implementation in HARD, a code built upon FleCSI. Using the Ghost Fluid Method enhanced with a Riemann solver, HARD allows us to simulate sharp material interfaces. Performance optimizations achieve significant memory reduction and excellent GPU scalability. Our code is validated with shock tube, Rayleigh-Taylor instability, and shock-bubble interaction benchmarks.**

I. Introduction

Accurate and scalable multiphysics codes are central to many real-world applications. The open-source code HARD (Hydrodynamics And Radiation Diffusion), built on FleCSI (Flexible Computational Science Infrastructure), targets these needs.

FleCSI is a C++ framework for developing scalable, performance-portable scientific applications. It provides a task-based programming model separating application logic from parallel execution, data management, and runtime orchestration. Through a unified interface, FleCSI supports multiple distributed-memory backends (MPI, Legion, and HPX), together with Kokkos for on-node parallelism. This design enables applications to efficiently target modern heterogeneous HPC systems while providing a user-friendly development environment for domain scientists.

HARD utilizes FleCSI capabilities to solve compressible hydrodynamics coupled with radiation diffusion using finite-volume methods, maintaining clear separation between physics and execution while efficiently targeting both CPUs and GPUs.

This poster presents a scalable, memory-efficient implementation for modeling immiscible material evolution using the Ghost Fluid Method enhanced with a Riemann solver. The approach has applications spanning reactive flows to high-energy-density physics applications, with benchmarks including shock tubes, Rayleigh-Taylor instability, and shock-bubble interactions in 2D and 3D.

II. Physics & Numerical Methods

Multi-material hydrodynamics studies systems composed of two or more immiscible materials, such as a solid and a gas. Assuming inviscid flow, each fluid is described by the evolution of the state (density ρ, momentum $\rho\mathbf{v}$, and energy E) via the compressible Euler equations:

$$\partial_t \begin{bmatrix} \rho \\ \rho\mathbf{v} \\ E \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\mathbf{v}^T + p\mathbb{I}_d \\ \mathbf{v}(E+p) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{0} \\ 0 \end{bmatrix}.$$

To ensure accuracy we must avoid diffusion across material interfaces. HARD tracks sharp interfaces using the Ghost Fluid Method, representing them by a signed-distance function, ψ, which defines a level-set. The hydrodynamic equations of the different materials can be solved over extended domains by using ghost states in neighboring domains. These ghost states preserve the properties of the original material while reproducing the flow behavior of the adjacent fluid. Whereas the original GFM extends pressure and velocity by copying and density through extrapolation, our implementation uses a multi-material Riemann solver to construct physically consistent states across the interface.

Level set functions evolve using a WENO5m finite-difference scheme with strong stability preserving Runge-Kutta time integration. Hydrodynamic equations for each material employ a WENO5z finite-volume scheme with second order Runge–Kutta integration. These temporal discretizations leverage time-integration abstractions from FleCSolve (FleCSI linear algebra package), enabling a unified numerical framework across multiple physical models within HARD.

III. Performance Optimization

The parallel structure of HARD's explicit stencil updates enables efficient on-node execution with Kokkos, particularly on GPUs. Performance analysis shows that a naive multimaterial implementation can introduce substantial communication bottlenecks, primarily because of material-state extrapolation. These bottlenecks are addressed through several optimizations to the FleCSI backend, including more efficient use of host buffers, pooled pinned memory, and stream-based overlap of communication and computation. We additionally introduce sparse data structures for the Ghost Fluid Method, storing extrapolated data only near material interfaces. This approach reduces the algorithm's memory footprint without affecting accuracy, yielding a 25% reduction in memory usage across all problem sizes evaluated in the shock–bubble tests.

Further optimizations target task granularity in HARD. For example, small independent tasks are fused to reduce redundant memory reads and writes across GPU kernels.

IV. Experimental Setup and Results

We run simulations on LANL's Chicoma supercomputer, where each compute node is equipped with 128× AMD EPYC 7H12 cores (standard partition) or 4× NVIDIA A100-40GB (accelerated partition). We first validate HARD comparing 1D simulations against analytically known solutions, then we perform scalability test with 2D and 3D simulations.

We perform weak and strong scaling studies on CPU and GPU architectures using one full node as baseline. Each scaling point is evaluated three times and averaged. The 2D helium bubble interaction benchmark (8192×16384 cells, 107 GB memory footprint) demonstrates notable parallel performance on LANL's Chicoma supercomputer. Weak scaling remains highly efficient, reaching 98.3% on 32 CPU nodes and 81.3% on 16 GPU nodes. Strong scaling tests achieve a 33.2× speedup on 32 CPU nodes (103.9% efficiency due to favorable cache effects) and an 8.7× speedup on 16 GPU nodes (54.2% efficiency). A single GPU node equipped with four NVIDIA A100 GPUs provides a 12.2× performance advantage over a 128-core AMD EPYC 7H12 CPU node.

We further prove our implementation comparing 3D high-resolution simulations against experimental results available in literature. We perform these simulations on 4 Chicoma nodes, each with 4 GPUs, for a total runtime of approximately 4 hours.

V. Conclusions

We introduced new multi-material modeling capabilities in the HARD code for hydrodynamics and radiation diffusion. Performance and benchmark results demonstrate that, thanks to FleCSI and the new optimized numerical method, both scalability and physical accuracy are guaranteed, making this tool optimal for large scale multiphysics applications.