



MULTISCALE UNIVERSAL INTERFACE (MUI)

<http://mxui.github.io>

Dr Wendi Liu (wendi.liu@stfc.ac.uk)

Multiscale Universal Interface (MUI) is a lightweight, header-only C++ library that enables scalable, minimally intrusive coupling of heterogeneous multi-physics and multi-scale solvers through MPI-based data exchange, advanced interpolation, and built-in coupling algorithms.

Value for Research

MUI empowers researchers to couple independent simulation codes into robust, scalable multi-physics & multi-scale workflows without rewriting legacy solvers, enabling high-fidelity, conservative data exchange across non-conformal interfaces and delivering strong performance on modern high-performance computing systems. By lowering the technical barriers to solver integration, MUI allows scientists and engineers to focus on the underlying physics rather than infrastructure, accelerating discovery in areas such as thermal-fluid-structure interaction, wave-structure interaction, and other complex multi-physics & multi-scale problems. Its lightweight design, built-in coupling algorithms, and multi-language support make it suitable for both established research codes and emerging computational frameworks, supporting reproducible, extensible, and future-ready scientific computing.

Development

MUI is a joint effort between Brown University, Lawrence Berkeley National Laboratory, UK Research & Innovation's Science & Technology Facilities Council and IBM Research. It was introduced in 2015 as a lightweight C++ library for concurrent coupling of heterogeneous solvers. Following its initial stable release in 2020, MUI has delivered 11 production releases, reflecting continuous refinement, extension, and hardening of its architecture. This trajectory culminated in a major architectural refactor and significant feature expansion in version 2.0, introducing enhanced interpolation methods, built-in coupling algorithms, a dedicated linear algebra subsystem, and expanded multi-language support.

Functionality

MUI provides a general-purpose, high-performance framework for coupling independent simulation codes into coherent multi-physics and multi-scale workflows. It enables heterogeneous solvers to exchange data efficiently and consistently at runtime, even if the solvers are written in different languages, using different

discretisation, having different time-stepping schemes, or with different parallelisation strategies. At its core, MUI implements MPI-based Multiple Program Multiple Data (MPMD) communication, allowing separate executables to run concurrently while sharing interface data through a lightweight, header-only C++ library. Exchanged quantities are represented as point clouds and interpreted via configurable spatial and temporal samplers, supporting mesh-to-mesh, mesh-to-particle, and particle-to-particle data transfer.

Advanced interpolation methods, including Radial Basis Function (RBF) interpolation, allow accurate data exchange across non-conformal interfaces. To support strongly coupled simulations, MUI includes built-in coupling algorithms such as fixed under-relaxation and Aitken approaches, integrated directly into the data exchange layer. These stabilise partitioned iterations and reduce convergence time, even in the presence of dynamic or evolving interfaces. A dedicated linear algebra subsystem underpins interpolation and solver acceleration without introducing external dependencies. The library supports C++, C, Fortran, and Python interfaces, enabling integration into legacy and modern codebases alike.

Communication efficiency is enhanced through the smart-send mechanism that reduces unnecessary message traffic in large-scale HPC environments. In practical terms, MUI allows research teams to connect existing solvers with minimal intrusion, preserve solver independence, and scale coupled simulations to thousands of cores, providing a robust and extensible infrastructure for high-fidelity multi-physics and multi-scale research.

The Future

The development roadmap for MUI is focused on strengthening its role as a robust, scalable infrastructure for next-generation multi-physics & multi-scale simulation on emerging HPC architectures. The built-in linear algebra subsystem will evolve to include optional high-performance backends and parallel solvers, enabling users to leverage optimised vendor libraries where appropriate while preserving the library's lightweight core. Exploration of GPU-enabled pathways and heterogeneous programming models, such as SYCL and modern C++ abstractions, will further prepare MUI for next-generation heterogeneous HPC systems. Language interoperability will continue to expand, with ongoing refinement of the existing wrappers for workflow integration and potential support for additional ecosystems where there is clear scientific demand. Improved tooling for diagnostics, benchmarking, and reproducibility will also strengthen long-term sustainability and transparency.

LICENCE

Dual-licensed under GPLv3 & Apache-2.0

Programming Languages

C++. Wrappers are provided for C, Fortran and Python.

Repository URL

<https://github.com/MxUI/MUI>