Performance Portable Finite Element Discretization Tools Applied to Plasma Applications

Roger P. Pawlowski*, Eric C. Cyr, Matthew Bettencourt, Eric T. Phipps, Christian R. Trott, Harold C. Edwards, John N. Shadid, Richard Kramer and Allen Robinson

Sandia National Laboratories
Albuquerque, NM 87185
USA
*e-mail: rppawlo@sandia.gov

ABSTRACT

Supporting scalable and performant equation assembly across multiple next generation architectures can add significant development overhead, code bloat and complexity to applications. Additionally, supporting multiple discretizations and analysis modes (e.g. steady-state, transient, multiscale couplings, bifurcation analysis, sensitivity analysis, and uncertainty quantification) can unduly burden the application developers with implementation requirements and increase code complexity. This presentation will discuss the design of a general set of interoperable tools for finite element assembly to address these issues. The toolset is comprised of a number of simple components that are designed to be stand alone, when possible, for reuse across various application settings. The tools include a degree of freedom manager for mapping mixed finite element bases and heterogeneous equation sets into parallel distributed linear algebra objects (e.g. residuals and Jacobians). The multiphysics assembly library uses a directed acyclic graph (DAG) for composable and reusable physics kernels in a multiphysics setting [2]. Assembly with hybrid data parallelism and task-based parallelism over the DAG is explored and evaluated. Performance portability across hardware node types, without requiring multiple application implementations, is achieved via the Kokkos programming model [1]. An embedded automatic differentiation component (Sacado), applied via templated scalar types and operator overloading, is used to generate sensitivities for implicit and IMEX solvers [3, 4]. The toolset is demonstrated on CFD, magnetohydrodynamics and multi-fluid plasma applications. Performance results for a mixed-basis multi-fluid plasma finite element formulation will be shown for Intel Phi, NVIDIA GPU and Intel Haswell architectures. The toolset is available as part of the Trilinos framework (trilinos.org).

REFERENCES

[1] Edwards, H. C., Trott, C. R. and Sunderland, D. Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *J. Parallel Distrib. Comput.* (2014) **73**:3202–3216.

[2] Notz, P. K., Pawlowski, R. P. and Sutherland, J. C. Graph-Based Software Design for Managing Complexity and Enabling Concurrency in Multiphysics PDE Software. *ACM T. Math. Software* (2012) **39**:1–21

[3] Pawlowski R. P. and Phipps, E. T. and Salninger, A. G. Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, Part I: Template-based generic programming. *Sci. Program.* (2012) **20**:197–219.

[4] Pawlowski, R. P., Phipps, E. T., Salinger, A. G., Owen, S. J., Siefert, C. M. and Staten, M. L. Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, Part II: Application to partial differential equations. *Sci. Program.* (2012) **20**:327-345.