



Enabling End-to-End Accelerated Multiphysics Simulations in the Exascale Era Using PETSc

Richard Tran Mills¹, Mark Adams², Satish Balay¹, Jed Brown³, Jacob Faibussowitsch¹, Matthew G. Knepley⁴, Scott Kruger⁵, Hannah Morgan⁶, Todd Munson¹, Karl Rupp⁷, Barry Smith⁸, Stefano Zampini⁹, Hong Zhang¹, and Junchao Zhang¹

¹Argonne National Laboratory

²Lawrence Berkeley National Laboratory

³University of Colorado Boulder

⁴University at Buffalo

⁵Tech-X Corporation

⁶University of Chicago

⁷Technische Universität Wien

⁸Flatiron Institute

⁹King Abdullah University of Science & Technology (KAUST)

The Portable Extensible Toolkit for Scientific Computation (PETSc) library provides scalable solvers for nonlinear time-dependent differential and algebraic equations and for numerical optimization; it is used in dozens of scientific fields, and has been an important building block for many computational geoscience applications. Starting from the terascale era in the 1990s and continuing into the present dawn of the exascale era, a major goal of PETSc development has been achieving the scalability required to fully utilize leadership-class supercomputers. We will describe some of the algorithmic developments made during the era in which achieving inter-node scalability was the primary challenge to enabling extreme-scale computation, and then survey the challenges posed by the current era in which harnessing the abundant fine-scale parallelism within compute nodes -- primarily in the form of GPU-based accelerators -- has assumed at least equal importance. We will discuss how the PETSc design for performance portability addresses these challenges while stressing flexibility and extensibility by separating the programming model used by application code from that used by the library. Additionally, we will discuss recent developments in PETSc's communication module, PetscSF, that enable flexibility and scalable performance across large GPU-based systems while overcoming some of the difficulties posed by working directly with the Message Passing Interface (MPI) on such systems. A particular goal of this talk will be to go beyond simply describing the work performed to prepare PETSc and simulation codes that rely on it to run on exascale-class systems, but to enumerate the challenges we encountered and to share the essential lessons learned that can help other developers to prepare and optimize their high-performance scientific computing codes for the exascale era.