

Terascale Optimal PDE Simulations (TOPS): Building an Holistic Approach to PDE-based Modeling

Large-scale simulations of importance to the Department of Energy often involve the solution of partial differential equations (PDEs). In such simulations, continuous (infinite-dimensional) mathematical models are approximated with finite-dimensional models. To obtain the required accuracy and resolve the multiple scales of the underlying physics, the finite-dimensional models must often be extremely large, thus requiring terascale computers. Fortunately, continuous problems provide a natural way to generate a hierarchy of approximate models, through which the required solution may be obtained efficiently by various forms of “bootstrapping.” The most dramatic examples are multigrid methods, but other hierarchical representations are also exploitable.

Under the Scientific Discovery through Advanced Computing (SciDAC) initiative, a nine-institution team is building an integrated software infrastructure center (ISIC) that focuses on developing, implementing, and supporting optimal or near optimal schemes for PDE simulations and closely related tasks, including optimization of PDE-constrained systems, eigenanalysis, and adaptive time integration, as well as implicit linear and nonlinear solvers. The Terascale Optimal PDE Simulations (TOPS) Center is researching and developing and will deploy a toolkit of open source solvers for the nonlinear partial differential equations that arise in many application areas, including fusion, accelerator design, global climate change, and the collapse of supernovae. These algorithms — primarily multilevel methods — aim to reduce computational bottlenecks by one or more orders of magnitude on terascale computers, enabling scientific simulation on a scale heretofore impossible.

Along with usability, robustness, and algorithmic efficiency, an important goal of this ISIC is to attain the highest possible computational performance in its implementations by accommodating to the memory bandwidth limitations of hierarchical memory architectures.

Background and Significance

Multicomponent nonlinear partial differential equations (PDEs) provide the common mathematical expression of many DOE simulations. PDE simulation codes require implicit solvers for multiscale, multiphase, multiphysics phenomena from hydrodynamics, electromagnetism, radiation transport, chemical kinetics, and quantum chemistry. Problem sizes are typically now in the millions of unknowns; and with emerging large-scale computing systems and inexpensive clusters, we expect this size to increase by a factor of a thousand over the next five years. Moreover, these simulations are increasingly used for design optimization, parameter identification, and process control applications that require many repeated, related simulations.

Unfortunately, the implicit solution algorithms currently used in many contemporary codes have far from optimal computational complexities and are invariably bottlenecks that limit the scalability of the entire application, independent of the quality of the implementation. For example, an increase in problem size of a factor of 100 can easily result in an increase in work requirements of 1000. In comparison, optimal complexity algorithms have work (and memory) requirements that grow only linearly with problem size. Multi-level (or multigrid) methods make up a class of optimal complexity algorithms that have produced spectacular improvements in overall simulation time. However, current multi-level software tends to be problem-specific and is mature only for scalar (as opposed to multicomponent) PDEs. Because of the potential payoff, the TOPS ISIC will expend much

of its effort on developing practical, usable multilevel methods for comprehensive aspects of PDE simulations.

The TOPS ISIC is concerned with five PDE simulation capabilities: adaptive time integrators for stiff systems, nonlinear implicit solvers, optimization, linear solvers, and eigenanalysis. The relationship between these areas is depicted in Figure 1. In addition, the ISIC will contain two cross-cutting topics: software integration (or interoperability) and high-performance coding techniques for PDE applications.

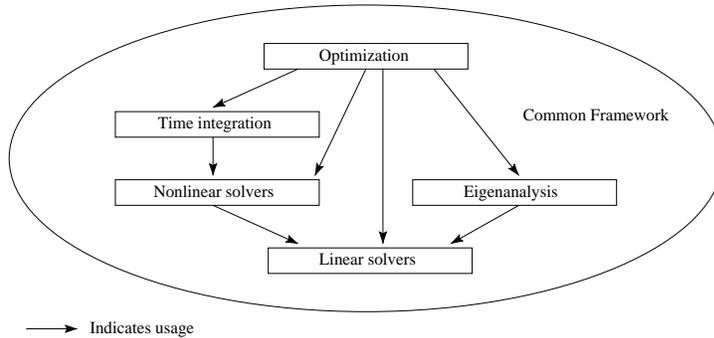


Figure 1: An arrow from A to B indicates that A typically uses B. Optimization of systems governed by PDEs requires repeated access to a PDE solver. The PDE system may be steady-state or time-dependent. Time-dependent PDEs are typically solved with implicit temporal differencing. After choice of the time-integration scheme, they, in turn, require the same types of nonlinear solvers that are used to solve steady-state PDEs. Many algorithms for nonlinear problems of high dimension generate a sequence of linear problems, so linear solver capability is at the core. Eigenanalysis arises inside of or independently of optimization. Like direct PDE analysis, eigenanalysis generally depends upon solving a sequence of linear problems. All of these five classes of problems, in a PDE context, share grid-based data structures and considerable parallel software infrastructure. Therefore, it is compelling to undertake them together.

Optimal (and nearly optimal) complexity numerical algorithms almost invariably depend upon a hierarchy of approximations to “bootstrap” to the required highly accurate final solution. Generally, an underlying continuum (infinite-dimensional) high fidelity mathematical model of the physics is discretized to “high” order on a “fine” mesh to define the top level of the hierarchy of approximations. The representations of the problem at lower levels of the hierarchy may employ other models (possibly of lower physical fidelity), coarser meshes, lower order discretization schemes, inexact linearizations, and even lower floating-point precisions. The philosophy that underlies our algorithmics and software is to make the majority of progress towards the highly resolved result through possibly low-resolution stages that run well on high-end distributed hierarchical memory computers.

The ingredients for constructing hierarchy-of-approximations-based methods are remarkably similar, be it for solving linear systems, nonlinear problems, eigenvalue problems, or optimization problems, namely:

1. A method for generating several discrete problems at different resolutions (for example on several grids)
2. An inexpensive (requiring few floating point operations, loads, and stores per degree of freedom) method for iteratively improving an approximate solution at a particular

resolution

3. A means of interpolating (discrete) functions at a particular resolution to the next finer resolution
4. A means of transferring (discrete) functions at a particular resolution to the next coarser resolution (often obtained trivially from interpolation).

We believe that software should reflect the simplicity and uniformity of these ingredients over the five problem classes and over a wide range of applications. We expect that with experience we will achieve a reduction in the number of lines of code that need to be written and maintained, because the same code can be reused in many circumstances.

Algorithms and software for the solution of linear and nonlinear systems of equations, especially those arising from PDEs, have been principal emphases of the Department of Energy research portfolio for decades. This points both to the central importance of this project, and also to the historical difficulty of reconciling the conflicting objectives of solver software technology. Solvers are supposed to be of *general purpose*, since a great diversity of applications require them, but they are also supposed to be *highly performant*, since they are often the inner loops of such applications. However, high performance usually requires exploitation of special structure (e.g., symmetry, dense blocking, geometrical or coefficient regularity), which may be different in different applications. Then, too, solvers for PDEs are supposed to be *robust* across all regimes of use, since scientists trained in the application domain cannot also be required to be expert in tuning solvers, but they are also supposed to have *optimal complexity*, since desired discrete problem size is limited only by the validity of the continuum model. Once again, algorithmic optimality (work and memory requirements a small multiple of their information-theoretic minima) is generally achieved by exploitation of special structure that cannot be assumed in a robust code.

These conflicting objectives do not describe a hopeless situation, however. The opportunity for 21st century solver developers is to exploit advances in object-oriented programming to construct highly versatile and adaptive software that finds, creates, and exploits structure wherever possible, while automatically “falling back” to conservative approaches in the remaining (hopefully lower-dimensional) parts of a problem. The solver toolkit of the future will be a collection of objects with rich and recursive interconnections, rather than a collection of subroutines through which a relatively small number of calling sequences are predefined. Algorithmic theory, scientific software engineering, and understanding of architecturally-motivated performance optimizations have all advanced significantly since the last time many applications communities “fastened onto” their canonical solver technology. Advances along these three fronts must be packaged, refined, freshly promoted, and supported for the benefit of the user community.

The efforts defined for TOPS, the co-PIs joining to undertake them, and the alliances proposed with other groups have been chosen to exploit the present opportunity to revolutionize large-scale solver infrastructure, and lift the capabilities of dozens of DOE’s computational science groups as an outcome. The co-PIs’ current software (e.g., hypre, PETSc, ScaLAPACK, SuperLU), though not algorithmically optimal in many cases, and not yet as interoperable as required, is in the hands of thousands of users, and has created a valuable experience base. The co-PIs have extensive networks of previous collaborations with each other and with key co-PIs in other SciDAC groups with which natural development synergisms exist — including complex geometry and adaptive gridding, high performance, and common component architecture. They also have tight collaborations

planned with applications teams. The primary applications collaborators will be allowed to influence TOPS priorities for software development. Since our solver software has many user groups already, we expect considerable additional user interaction beyond the primary SciDAC-supported collaborators.

Just as we expect our user community to drive our research and development work, we expect to significantly impact the scientific priorities of users by emphasizing optimization (inverse problems, optimal control, optimal design) and eigenanalysis as part of our solver toolkit.

Optimization subject to PDE-constraints is a particularly active subfield of optimization because the traditional means of handling constraints in black-box optimization codes — with a call to a PDE solver in the inner loop — is too expensive. We are emphasizing “simultaneous analysis and design” methods in which the cost of doing the optimization is a small multiple of doing the simulation and the simulation data structures are actually part of the optimization data structures.

Likewise, we expect that a convenient software path from PDE analysis to eigenanalysis will impact the scientific approach of users with complex applications. For instance, a PDE analysis can be pipelined into the scientific added-value tasks of stability analysis for small perturbations about a solution and reduced dimension representations (model reduction), with reuse of distributed data structures and solver components.

Our motivating belief is that most PDE simulation is ultimately a part of some larger scientific process that can be hosted by the same data structures and carried out with many of the same optimized kernels as the simulation, itself. We intend to make the connection to such processes explicit and inviting to users, and this will be a prime metric of our success. The assembly of the project team and the organization of the proposal flow directly from this program of “holistic simulation”: *Terascale software for PDEs should extend from the analysis to the scientifically important auxiliary processes of sensitivity analysis, modal analysis and the ultimate “prize” of optimization subject to conservation laws embodied by the PDE system.*

Project Personnel

The TOPS project involves over two dozen senior personnel at nine institutions, plus associated post-doctoral and graduate student members. In the following list (accurate as of July 2001), the names of lead PIs at each institution are italicized.

Argonne National Laboratory Steve Benson, *Jorge Moré*, Todd Munson, *Barry Smith*

Carnegie Mellon University *Omar Ghattas*

Lawrence Berkeley National Laboratory Parry Husbands, Sherry Li, Osni Marques, *Esmond Ng*, Chao Yang

Lawrence Livermore National Laboratory Edmond Chow, *Rob Falgout*, Steve Lee, Radu Serban, Panayot Vassilevski, Carol Woodward

New York University George Biros, *Olof Widlund*

Old Dominion University *David Keyes*, Alex Pothen

University of California *Jim Demmel*

University of Colorado Xiao-Chuan Cai, Tom Manteuffel, *Steve McCormick*

University of Tennessee *Jack Dongarra*, Victor Eijkhout

David E. Keyes, Department of Mathematics & Statistics, Old Dominion University, dkeyes@odu.edu
30 July 2001