

Multigrid Solvers Do the Math Faster, More Efficiently

TAKE a simulation. A big simulation, in which you want to view a fairly high level of detail. Maybe you want to study inertial confinement fusion as part of the nation's Stockpile Stewardship Program or find out the effects of global climate change in a specific area or explore the mechanism by which stars collapse and then explode in supernovae. To model those events in three dimensions and at high resolution, you'll need immense computer codes. And processing these codes will require the computational speeds and large memories of massively parallel supercomputers, such as those developed for the National Nuclear Security Administration's (NNSA's) Advanced Simulation and Computing (ASC) Program.

Ideally, you would want the computing time to remain the same even though the size of the problem increased. But unfortunately, as simulations become more lifelike and detailed and more processors are added to handle the calculation, the run times to solve a calculation may become even longer.

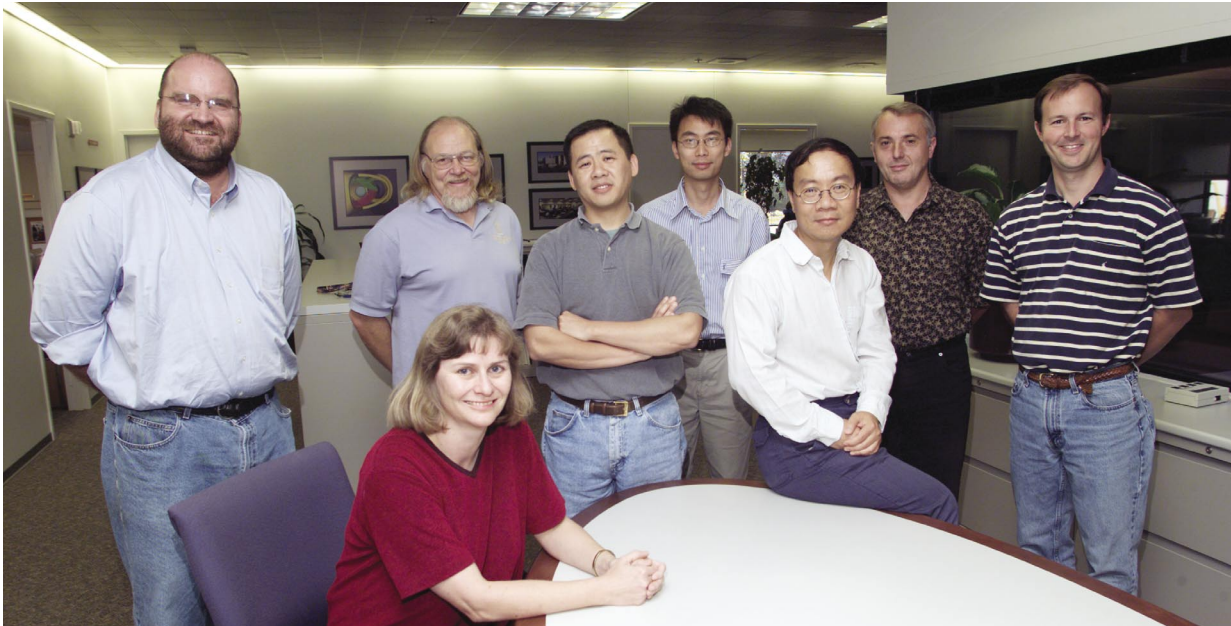
At least, that was the situation until scientists developed software codes called scalable linear solvers. These codes,

including those developed at Lawrence Livermore, help keep simulations running fast as they grow larger and more processors are added to attack the problem.

Flowing from Groundwater Research

According to computational mathematician Rob Falgout, Livermore researchers began to develop scalable linear solvers in the early 1990s as part of a project funded by the Laboratory Directed Research and Development (LDRD) Program. "Steven Ashby and I were working with a team to develop a code called ParFlow, which simulates the flow of groundwater through different kinds of materials underground," says Falgout. "As a part of that LDRD project, we began developing solvers—special algorithms or processes for solving specific problems—to significantly speed up the solution of the mathematical equations generated by ParFlow."

In that particular application, the sites being modeled were several square kilometers, and the calculations had to resolve differences in subsurface media of a few meters. As a result, the



The Scalable Linear Solvers project team: (seated) Ulrike Yang; (standing, left to right) Jim Jones, Van Emden Henson, Barry Lee, Edmond Chow, Charles Tong, Panayot Vassilevski, and Rob Falgout. Not pictured: Jeff Painter and Tom Treadway.

computational grids had up to 100 million spatial zones. For ParFlow to handle such detailed simulations, the researchers had to design the code so it would effectively harness the power of massively parallel processing.

ParFlow did all that and more. It proved to be portable and scalable—that is, it can run on a variety of computing platforms, and it efficiently uses the processors that are added to handle larger problems. It's also very fast, reducing the time for some simulations from 30 minutes to only 13 seconds.

Part of ParFlow's success sprang from the multigrid approach it used, which solved problems 100 times faster than other solvers. Since that initial success, interest in multigrid linear solvers for parallel computers has grown not only at Livermore but throughout the scientific computing community. As a result, with support from the ASC Program and the Department of Energy Office of Science's Mathematical, Information, and Computational Sciences Program, the Scalable Linear Solvers project was formed at the Laboratory's Center for Applied Scientific Computing. Led by Falgout, the project now employs 10 people and is considered a premier group in this esoteric area where mathematics and computational sciences mesh.

Solvers R Us

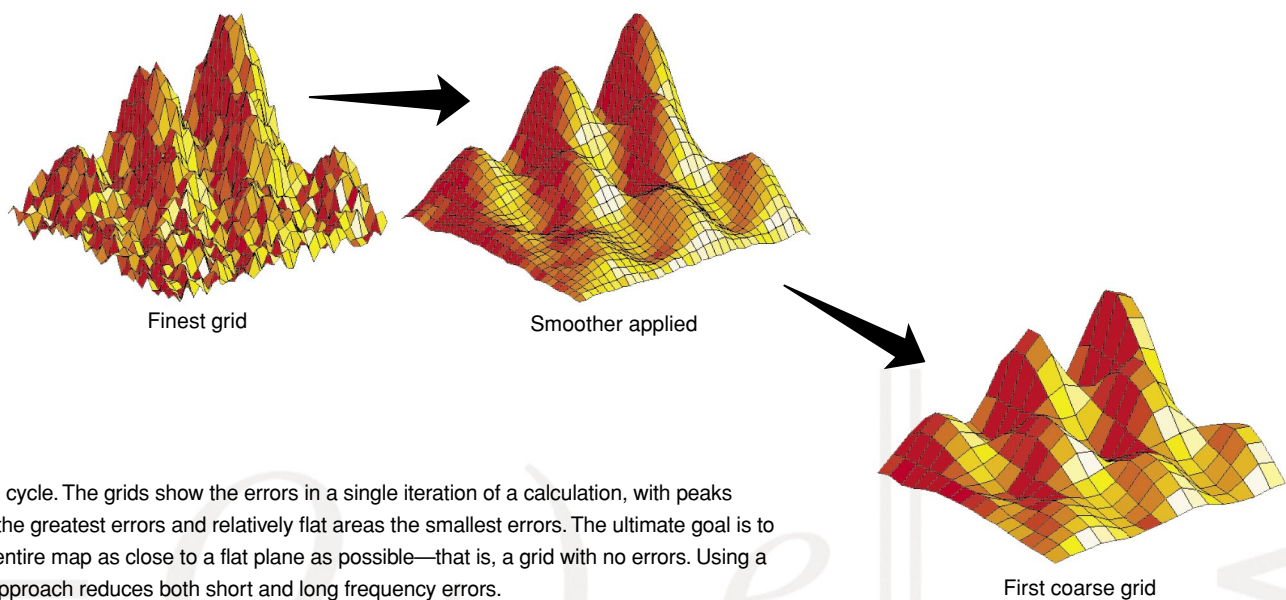
In general, a solver is an algorithm for calculating the solution to a set of mathematical equations. A linear solver calculates the solution to a linear system of equations—a set of equations just like those found in high school algebra, except that the set has millions or billions of equations to solve. The unknowns

in these linear systems of equations can represent a variety of physical quantities, such as the pressure at a particular location underground or the new location of a piece from an automobile frame after a crash. In most simulation codes at Livermore, these unknowns are also associated with points on a grid. For example, in the groundwater-flow application, the grid points represent underground locations.

In many large-scale scientific simulation codes, a large fraction of the overall run time on the computer is spent in linear solvers. Cut the processing time for these solvers, and the total run time shrinks. Thus, says Falgout, much of the research and development in scalable algorithms is aimed at solving these large, linear systems faster and more efficiently on parallel computers.

In the multigrid approach, code developers can reduce this time by making clever use of a sequence of smaller linear systems, each associated with a coarser grid—hence, the term multigrid. Computational mathematician Jim Jones, who also works on the project, says, "Probably the simplest way to think about it is to imagine a two-dimensional problem, where you have some initial guess of what your answer should be. The idea is to generate a new or better guess through some simple procedure or algorithm, then repeat the procedure until your guess converges with the solution of the linear system."

Initially, each unknown in the guess is incorrect, and it may differ significantly from the correct value. When these errors are plotted, the plot has lots of peaks and valleys, as shown on p. 18. The goal is to generate a new guess that matches the correct value, or has zero error. Standard solver algorithms generate new guesses



A multigrid cycle. The grids show the errors in a single iteration of a calculation, with peaks indicating the greatest errors and relatively flat areas the smallest errors. The ultimate goal is to make the entire map as close to a flat plane as possible—that is, a grid with no errors. Using a multigrid approach reduces both short and long frequency errors.

with a “smoother,” a special method that smooths the errors when they are plotted. With the multigrid approach, code developers take advantage of the smoother by recognizing that low-frequency (or long-length-scale) errors can be accurately and efficiently resolved on a coarser, or smaller, grid.

The Livermore researchers are creating scalable linear solvers based on this multigrid technology. The main ingredients of a multigrid solver are the smoother, the coarse-grid linear systems, and the procedures for transferring data back and forth between the algorithms. “If all the components are properly defined,” says Jones, “the method will uniformly damp error frequencies, and the computational cost will depend only linearly on the problem size. In other words, multigrid algorithms are scalable.”

The Laboratory’s multigrid solvers are based on two methods: geometric and algebraic. Geometric multigrid methods are used for linear systems defined on rectangular grids or meshes; algebraic multigrid methods are for linear systems defined on unstructured, or nonrectangular, grids. The geometric methods are used more often, but constructing the solver requires geometric information about the physical problem. Algebraic methods require no geometric information but are more difficult to design.

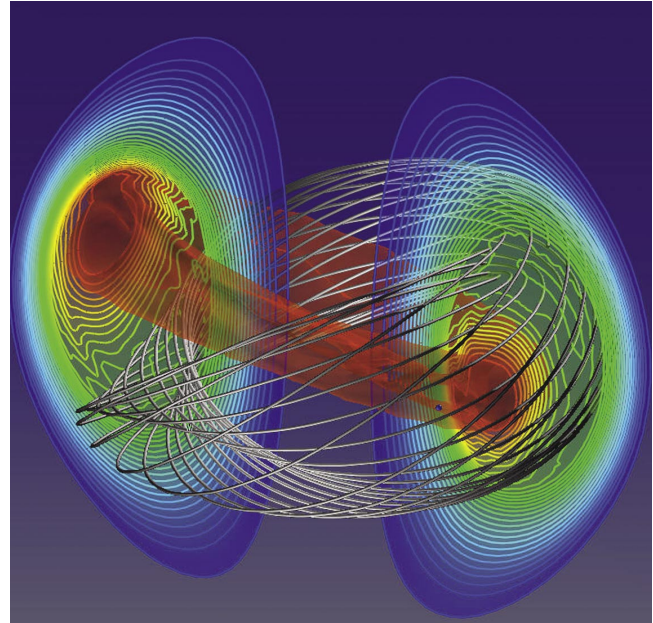
The Livermore team implements its solver algorithms in a software library called *hypre*, which runs on simple laptops and workstations as well as on massively parallel computers such as ASCI White. The solver codes in *hypre*—including the algebraic multigrid code BoomerAMG and the geometric multigrid solvers SMG and PFMG—are maintained by the project team and are available to the scientific community worldwide.

Making the Impossible Possible

Scalable linear solvers are allowing scientists to both pose and answer new questions. For example, consider a simulation at a particular resolution that would take several days to run. Increasing the resolution to make the model more accurate and lifelike means that the simulation will take even longer to run. In the world of simulations, run time is money. Because the Livermore multigrid solvers reduce the run time, scientists can push their simulations to the next level of detail.

At Livermore, for example, researchers integrated a parallel algebraic multigrid code into the hydrodynamics code ALE3D to solve difficult elasticity problems for modeling the deformation of materials. Researchers in Germany have also used *hypre* solvers to predict results from operations to correct facial deformities—another use of elasticity equations, but this time for medical applications.

“For those of us on this project,” says Falgout, “our work is mostly about mathematics. But it’s important to remember that the equations we’re trying to solve are not just abstract symbols and numbers. They describe real physical processes that researchers



The Department of Energy’s Scientific Discovery through Advanced Computing (SciDAC) Program is funding multigrid research and development for various application areas. For example, algebraic multigrid techniques are being applied to tokamak fusion applications in collaboration with the SciDAC Center for Extended MHD Modeling (CEMM). This example from a CEMM simulation is of a tokamak sawtooth instability showing pressure contours and surface with some magnetic-field lines.

are trying to better understand—whether it’s radiation flow in a supernova, the flow of groundwater through the subsurface, or the behavior of a plasma in a complex magnetic field. Today, computer simulations are increasingly important in scientific investigations, aiding or even taking the place of traditional experiments. So any method we can find to make detailed, three-dimensional problems run more quickly and efficiently opens new doors to our understanding of the world.”

—Ann Parker

Key Words: Advanced Simulation and Computing (ASC) Program; algebraic multigrid; ASCI supercomputers; geometric multigrid; *hypre*; Laboratory Directed Research and Development (LDRD) Program; linear equations; Mathematical, Information, and Computational Sciences Program; scalable linear solvers; simulation codes; Scientific Discovery through Advanced Computing (SciDac) Program.

For further information contact Rob Falgout (925) 422-4377 (rfalgout@llnl.gov).