

Perfecting the LANGUAGES and TOOLS of science

BY DON MONROE

The Terascale Optimal PDE (Partial Differential Equations) Simulations (TOPS) project focuses on creative algorithms and software that enable scientists to analyze natural phenomena using the highest available computing power. In mid-November 2005, the team unveiled the TOPS Solver Component — a single interface that provides access to a wide variety of tools.

TOPS researchers interface with application scientists to facilitate scientific discovery through high-end computing and high-performance simulations.

Scientific discovery involves understanding the intricacies of nature and replicating them in controlled ways: in calculations, in the laboratory, or, increasingly, in computer simulations or “virtual laboratories.” To exploit the growing power of computers, however, researchers must describe the world in terms of mathematical equations that connect observable quantities with scientific laws. Partial differential equations (PDEs) are a group of such equations that can quantitatively describe many natural phenomena (see sidebars “Different problems, different equations,” p52, and “What is a PDE?,” p53). Efficiently solving such equations is a critical enabler for scientific computations.

One of the key features of the SciDAC program is collaborative research between scientists who specialize in particular disciplines, computational scientists, and applied mathematicians. These scientists are using computer simulations to analyze a host of problems that cannot be probed by experiment and theory alone. Topics range from the scientific challenges of physics, chemistry, and combustion to the evolution of the Earth’s climate and the complexities of biological and astronomical systems.

Since 2001, the SciDAC initiative has been driving the state of the art in advanced scientific computing, and pushing science forward via multiple parallel efforts. Special application projects within the program focus on specific science

areas, in fusion energy science, high-energy and nuclear physics, biological and environmental research, and basic energy sciences. Examples of this work are discussed elsewhere in this issue. Other projects advance enabling technologies and aim to bolster the infrastructure and techniques that facilitate advanced scientific computing. Some of these projects allow users easier and more effective access to computing resources, while others, such as Integrated Software Infrastructure Centers (ISICs), advance the techniques that enable the computations.

The TOPS project

The Terascale Optimal PDE Simulations (TOPS) project is an ISIC that specializes in the solution of PDEs. It provides techniques and algorithms for efficiently solving PDEs to obtain insight into the underlying science. TOPS researchers interface with application scientists to facilitate scientific discovery through high-end computing and high-performance simulations.

The prefix “tera” in “terascale” means trillion, or 10^{12} in the more compact scientific notation. The scale has many meanings, since demands for various types of resources, such as memory, computational speed, and communication bandwidth often increase together. Large-scale problems might require 1 million Mbyte of memory, for example, as well as 1 trillion floating point oper-

An explosion of scientific discovery

ILLUSTRATION BY A. TOREX. SOURCE: J. Phys.: Conf. Ser. 187/02/11

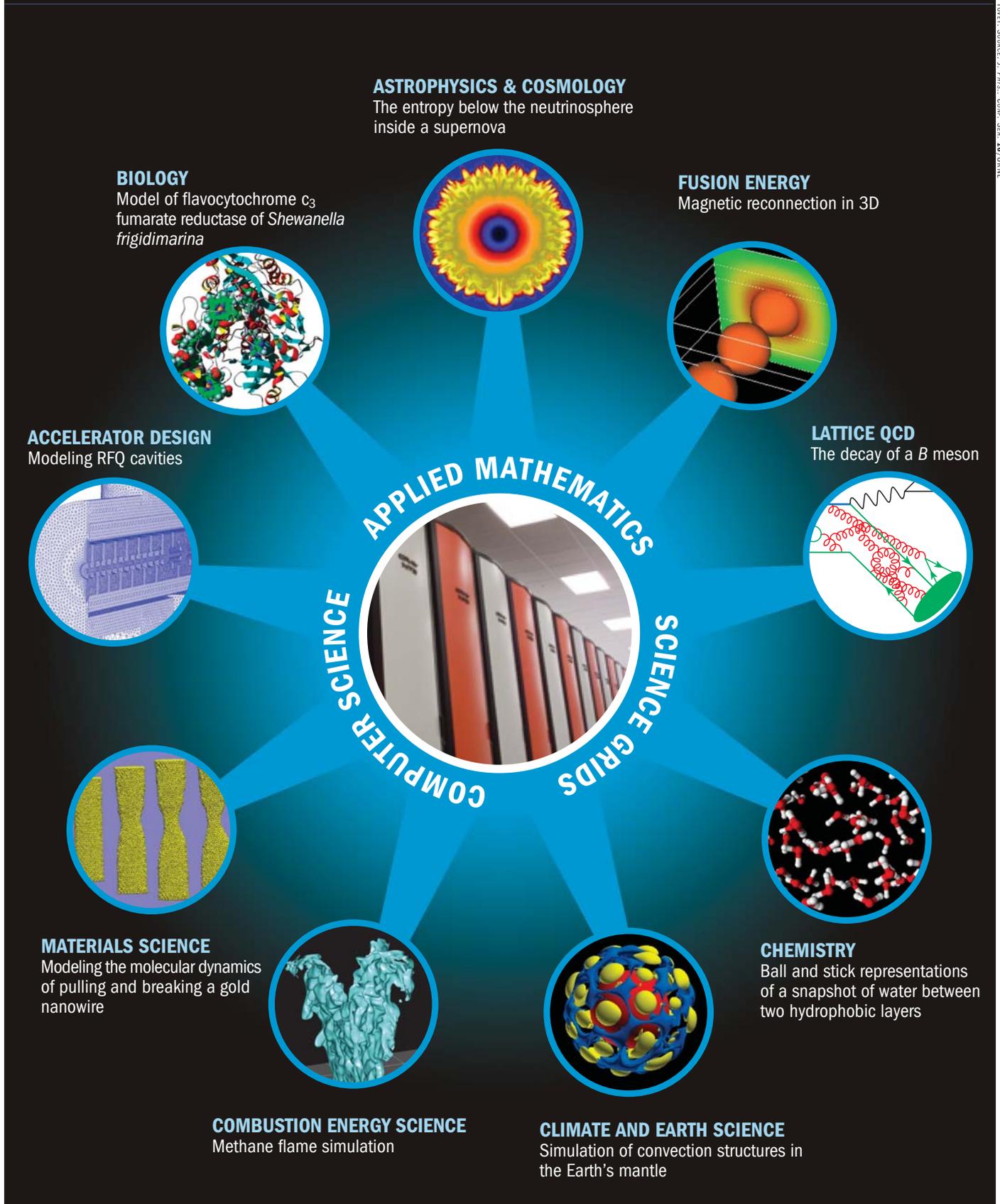


Fig. 1. Software tools that are being developed and enhanced by the TOPS program allow machines like the ORNL Leadership Class Computers (center, and see p38) to simulate ever more complex systems. This allows researchers to resolve a wider range of scales of many phenomena.

Different problems, different equations

A huge variety of scientific problems can be described by PDEs. They often go by the names of their eminent inventors, such as Maxwell or Navier-Stokes in the examples below.

As a rule, individual quantities follow relatively simple rules. Electric and magnetic fields, for instance, obey Maxwell's equations. One of these four PDEs relates the rate of change of the electric field with time to the spatial variation of the magnetic field. A second relates the change of the magnetic field to the variation of the electric field. Combining the two produces the wave equation, whose solutions include propagating electromagnetic radiation ranging from radio waves through light waves to gamma rays. Because Maxwell's equations are linear, these different waves can all pass through each other without interacting.

To describe the motions of a fluid that is subjected to gravity, pressure gradients and viscous forces, scientists use the

Navier-Stokes equations. In contrast to Maxwell's equations, the resulting fluid motions, including complex flows like turbulence, cannot in general be written in a simple mathematical form. Moreover, because the equation is nonlinear, different types of motion strongly influence one another.

Stars and fusion plasmas can be considered to be electrically charged fluids. To describe them, scientists must combine the fluid equations with Maxwell's equations. The resulting magnetohydrodynamic equations have solutions that show a tremendous variety of coupled motions of the fluid and the electromagnetic field.

Even when the underlying equations are simple, when several of them are combined the resulting system can often only be solved using computer simulation. For example, imagine a cubic meter of air outside your window. Any difference in pressure between opposite faces will cause the air to flow, as will various other

forces. Heat will flow if the temperature differs across the cube. Molecules or particles in the air will be heated by the sunlight, and will emit infrared radiation. If the air cools, water may condense, releasing more energy and forming droplets that can scatter sunlight. If the air moves, the cube carries its contents with it, to be replaced by a new one with slightly different properties. On top of all of this, chemical reactions will change the cube's composition.

Although there's a lot going on inside it, the various phenomena are well understood. They and their interactions can be described mathematically, at the scale of a single cube. What is very difficult to predict, though, is how more than 10^{18} such cubes around the globe, pummeled by sunlight from above and interacting with the land and ocean below, give rise to the complexities of hurricanes or global climate change. To even begin to describe these phenomena requires massive computer simulations.

In mid-November 2005, the team unveiled the TOPS Solver Component — a single interface that provides access to a wide variety of tools.

ations per second. The goal of the team is to use the vast resources available efficiently.

Enormous improvements in computational hardware have been made in recent years, but improvements due to better algorithms have been equally significant. On the hardware side, mass-produced microprocessors have enabled new computer architectures, in which tasks are shared by hundreds of thousands of processors. These clusters can attack much larger problems than one processor working alone.

Unless this division of labor is arranged carefully, however, most processors can spend much of their time idle as they wait for others to provide them with data. Therefore, advances in software and algorithms that make it possible to keep all the available resources usefully busy have been critical to maximizing the use of these parallel processing possibilities (see sidebar "Software vs. hardware," p53). The improvements have been particularly dramatic for the PDEs that are at the heart of the TOPS agenda. These equations provide the mathematical link between the science and the computation. Like the science they describe, they are inherently complex.

The TOPS project develops and distributes enabling technology, including sophisticated algorithms and simulation codes. These tools allow the use of modern high-performance computers to solve the PDEs that probe science across

a wide range of applications. By working closely with subject-matter experts and assessing how the software is used, TOPS researchers are endeavoring to make the solvers more user-friendly for application scientists. With this in mind, in mid-November 2005, the team unveiled the TOPS Solver Component — a single interface that provides access to a wide variety of tools.

Multiple scales

A central challenge in the study of large systems is to cope with the fact that structures on many scales interact. Natural phenomena span length scales covering many orders of magnitude, as well as vastly differing time scales. For example, micrometer-sized particles in the atmosphere, though tiny, can strongly affect the large-scale heating of the atmosphere and its chemistry over many kilometers of distance. And an exploding supernova, though very large, develops structure on a very fine scale relative to the star, both because of tiny eddies of fluid flow and because of the shock wave that the explosion generates (see figure 4, p54). Tokamak fusion plasmas involve time scales ranging from picoseconds to many minutes. Biological systems, too, may represent a combination of "multi-scale" and "multi-rate" problems.

In addition to issues of size and rate, some problems researched by SciDAC are also "multi-component" and "multi-physics." Multi-component

Software vs. hardware

The exponential improvements in computational hardware over the last few decades are well known. They are often described in terms of Moore's law, which states that the number of transistors on a chip will double every 18 months. The more densely packed transistors are also faster and consume less energy. Many observers, however, believe that the era of exponential improvement will come to an end over the next decade or so.

Few people realize that improved software algorithms have contributed just as dramatically to the increases in computational power as hardware developments. The chart in figure 2 shows the improvement in the time

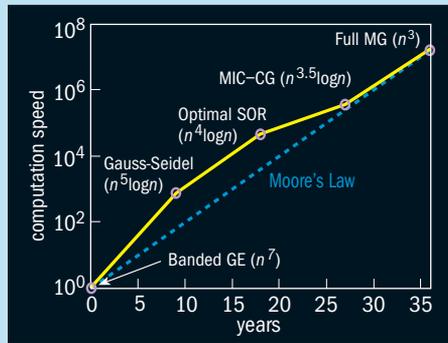


Fig. 2. Improvements owing to new software algorithms are represented by the yellow line. The increase in calculation speed is shown for a 3-D system with $n = 64$ points on each side. Increasing chip sizes are shown in blue.

needed to solve Poisson's equation on an $n \times n \times n$ cubic grid where $n = 64$, which requires solving a sparse matrix. The operations required for straightforward banded Gaussian elimination rise as n^7 .

Successive improvements, including a Gauss-Seidel method, optimal successive over-relaxation (SOR), conjugate gradient, and multigrid methods, successively reduce this demand to n^3 . Thus, for $n = 64$, the full multigrid technique is able to solve this problem 16 million times faster than the banded Gaussian elimination. The memory requirements are also reduced, from n^5 to n^3 . For a larger system, the improvement would be even more dramatic.

What is a PDE?

Much of the complexity in the world reflects the large-scale consequences of simple, local rules. For example, each segment of a guitar string is pulled by the tension of neighboring sections of string. If at a particular instant the string is curved, the pull from each end is not in exactly opposite directions, so a net force acts to set the segment moving. Once in motion, the inertia of the section tends to keep it moving, so that it overshoots the point at which the forces balance. As in a child's swing, a restoring force combined with inertia results in oscillations.

Scientists analyze such situations by using the language of differential equations to describe what is happening at each point. The solution is the complete response of the string, for example, to being plucked. For a simple system like a guitar string, there is an exact mathematical description of the resulting motions. These motions can be broken down into separate sinusoidal oscillations with different numbers of wavelengths along the length of the string. Even for more complex linear systems, finding these "eigenmodes" and their frequencies suffices to describe their behavior.

For example, figure 3(a) shows the tension with which neighboring sections of a guitar string pull on a small piece of length Δx . The curvature of the string causes a net force that accelerates the section to reduce the

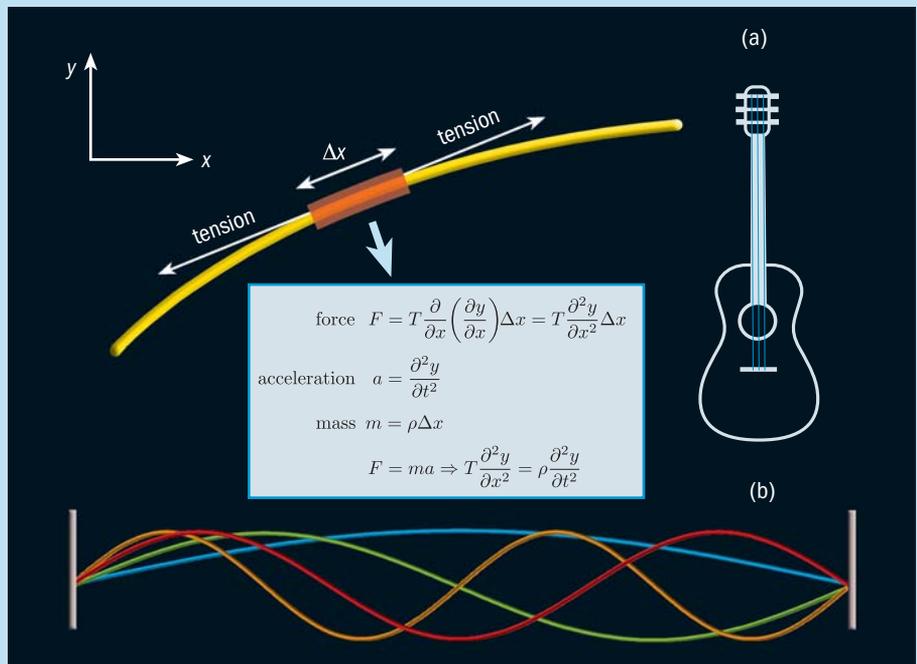


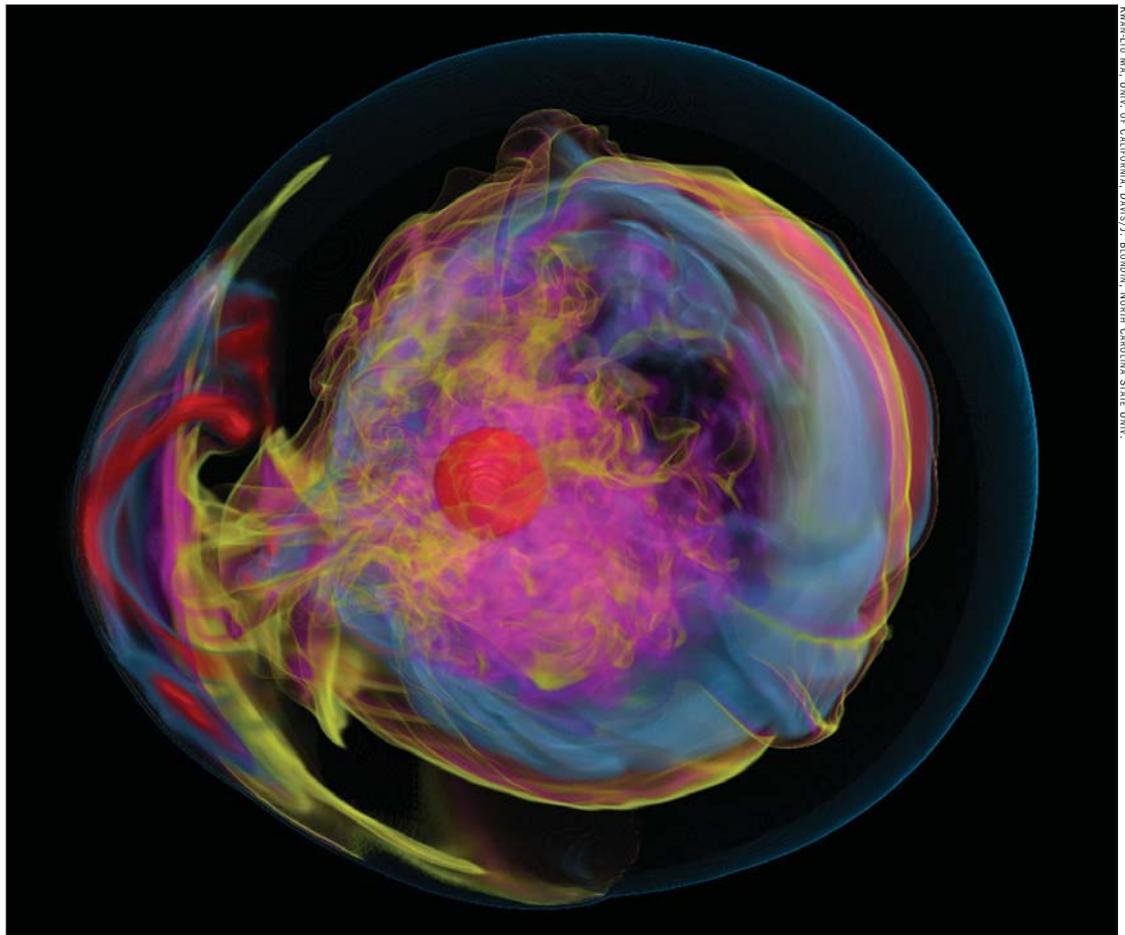
Fig. 3. The tension on a guitar string is affected by a force that can be described using a PDE (in box).

curvature. The displacement y depends on both position x and time t , and is governed by a partial differential equation (PDE). This equation is linear, because multiplying y by a constant doesn't change the equation.

In figure 3(b), the solutions to this simple PDE are sinusoidal waves, the frequencies and wavelengths of which are determined by the

tension T and density ρ of the string. In a guitar, the string is held fixed at either end. Within these boundary conditions, the string can move in a set of simple oscillatory solutions, or eigenmodes, corresponding to integral numbers of wavelengths between the boundaries. All of the possible motions of the string can be described by adding up the eigenmodes.

Even the most complex set of PDEs can be expressed as a balance between terms representing various fluxes and rates that must equal zero.



Kwan-Liu Ma, Univ. of California, Davis / Bonomi, Noah Carolina State Univ.

Fig. 4. A simulation of a core-collapse supernova (see feature “Modeling the first instants of a star’s death” on p26). Simulating the details of the shock wave (the hazy blue halo) and the fine turbulent details of the flow requires a grid spacing that is very small compared to the size of the collapsing star. In addition, to capture the asymmetrical features of the collapse the simulation must be done in three dimensions rather than being idealized to one or two dimensions. These details vastly increase the computational demands.

systems have different parts that overlap in space and interact with one another, while multi-physics problems include diverse physical phenomena, such as electromagnetic waves and fluid flow, in the study of a complex process.

All of these factors add to the difficulty of simulating complex systems efficiently and finding solutions that are valid across these multiple rates, scales, components and types of science. Successful analysis techniques must include relevant science at various levels of detail and utilize computer power to keep track of all the interactions.

Gridding the continuous

Complex physical and biological phenomena usually look smooth at the macroscopic level. Although on the tiny scale of atoms the world is built of discrete particles, on the length scales that are relevant to, say, a fusion reactor or an evolving star, this discreteness can often be ignored. The apparent smoothness is embodied in the PDEs that researchers use to describe such systems. These “continuous” equations deal with the

average properties of many particles, like pressure, temperature, and charge density, and their gradual variation in space and time.

To solve these equations using numerical methods on computers, however, this continuous world must again be broken into discrete parts. These smaller parts are not the actual particles of nature. Rather, the researchers describe their system on a mesh, with each unit of the mesh representing many particles. This replaces a single, continuous equation for the whole system with a large number of discrete ones. For example, in the PDE for a guitar string, the curvature of the string at each point can be estimated from the change in its slope at neighboring points (see figure 3).

Choosing the grid

Choosing the correct grid for the task is crucial to efficiently solving large problems. The simulation must be performed with a very fine grid if internal details are to be captured accurately. However, if the grid is made too fine, the calcu-

Making a continuous problem discrete

Choosing the correct grid for the task is crucial to efficiently solving large problems.

Solving PDE problems numerically requires replacing smoothly varying quantities with a set of discrete values on a grid. The simplest grid is a simple rectangular lattice, for which it is easy to express the derivatives that enter the differential equations in terms of the change in values between grid points.

Such regular grids result in inefficient calculations, however. The points are just as dense in regions where there is little variation as in the regions, for example near the edges of this airfoil, where finer detail is needed. The complex geometry of the grid makes it harder to set up the equations, but in the end the calculations run much faster.

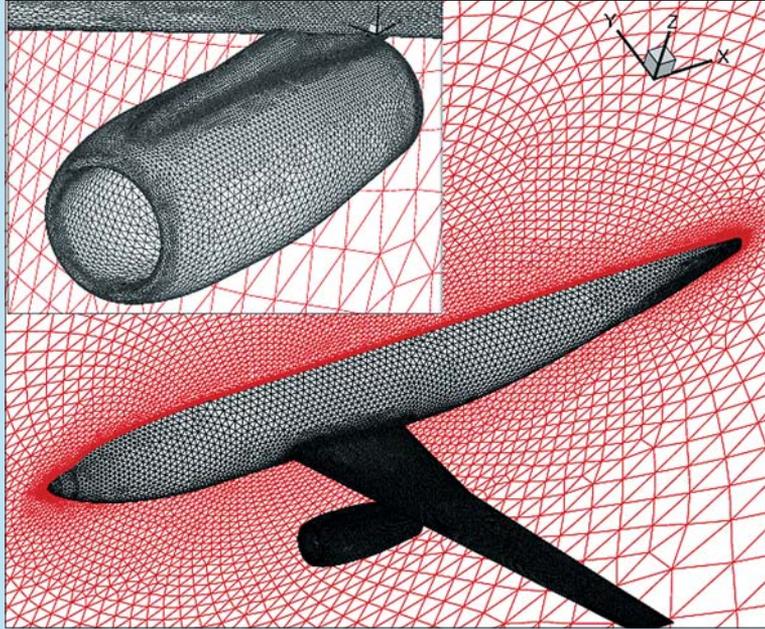


Fig. 5. It is important that the correct amount of detail is used in grids for PDEs. If the grid is made too fine, the calculations may take longer than necessary, and may accumulate errors. Conversely, a grid that is too coarse may blur important details.

lations may take much longer than necessary. They may also be overwhelmed by the accumulation of tiny errors. On the other hand, a grid that is too coarse may blur important details, such as sharp features in the boundaries. In addition, many models develop local structure, such as turbulence or instabilities, that can only be described properly by using a fine mesh that is sensitive to these details.

Use of a fine grid dramatically increases computational demands. Researchers frequently use non-uniform grids, with a finer spacing in regions where more detail is necessary (see sidebar “Making a continuous problem discrete,” above). Such grids complicate the initial formulation of the problem, but the payoff is that the computation runs faster, since time is not wasted on unimportant regions of data.

Even the most complex set of PDEs can be expressed as a balance between terms representing various fluxes and rates that must equal zero. This gives mathematical expression to conservation laws for mass, momentum, energy, charge, chemical species, or other fundamental quantities. For example, Poisson’s equation relating the electric field E to the charge density ρ can be written $\partial^2 E/\partial x^2 + \partial^2 E/\partial y^2 + \partial^2 E/\partial z^2 - 4\pi\rho = 0$. Solving it for a particular spatial distribution of charge is equivalent to finding values for the electric field

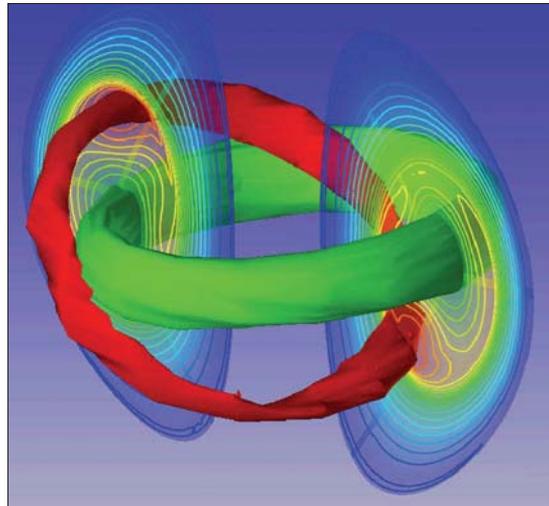
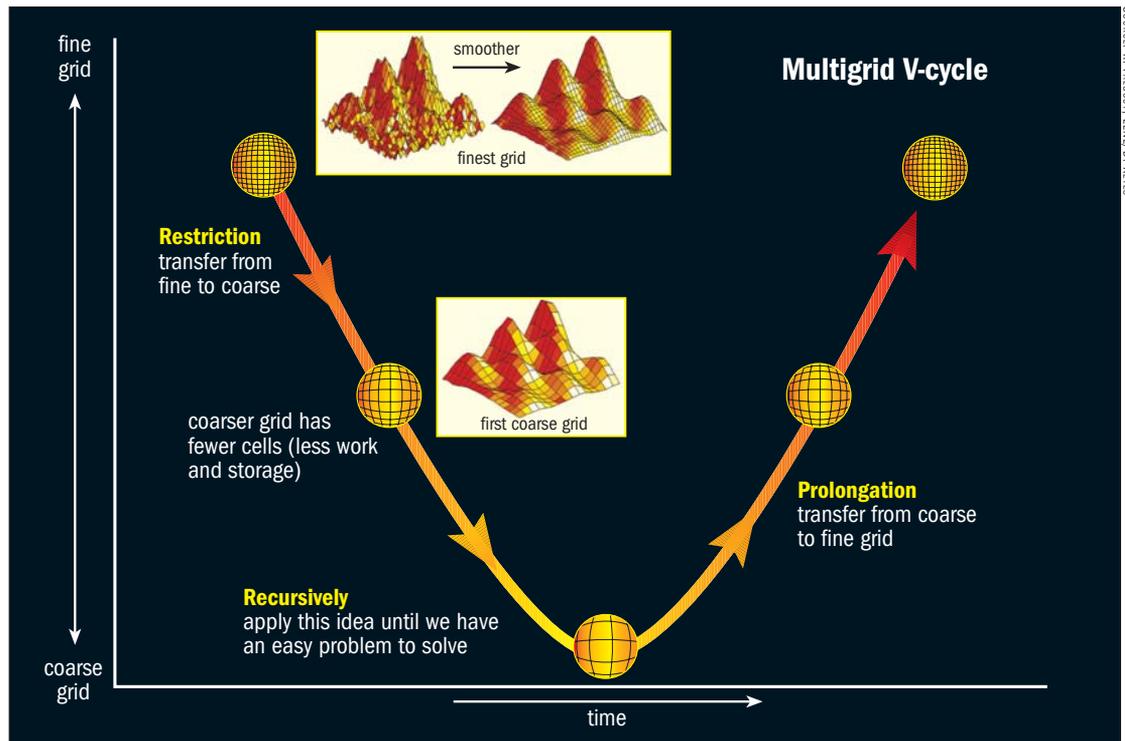


Fig. 6. A simulation of the trajectory of a packet of charge in the toroidal volume of a tokamak for fusion applications.

at every point in space, so that this expression is zero everywhere.

An effective method for solving many such equations is to start by guessing a solution, and evaluate the left side of the equation everywhere. The deviations from zero are then used to modify the trial solution, and the process is repeated. This iterative procedure will eventually settle on a field distribution that satisfies the equation everywhere to any desired level of accuracy. A

Modern multigrid techniques reduce the computational effort so that it rises only in direct proportion to the number of points on the finest grid. This scalability lets researchers tackle much larger problems than otherwise would have been possible.



SOURCE: R. FALGOUT, LIN/D. KEYES

Fig. 7. Most of the computational effort in a simulation is expended at the finest level of detail. Unfortunately, these expensive calculations must be repeated many times to correctly describe the variation on long-length scales. Researchers employ multigrid techniques to pass information back and forth between coarse and fine grids. This smoothes out noise in the simulation quickly, allowing for scalable computations for which the required resources only grow as rapidly as the size of the problem.

The need for scalability

Modern algorithms strive for scalability, meaning that the required computational resources grow only as fast as the size of the problem to be solved. In many other situations (and for older PDE algorithms), the resource demands increase much faster than the size of the problem itself as the problem gets bigger.

For PDEs, explains TOPS project Principal Investigator Dr David Keyes of Columbia University, “the motivation for scalability is that

you’re trying to approach a continuum.” If the scientists want to explore finer details of the system, they can distribute discrete pieces of the calculation to different computers, which do their work in parallel. By contrast, in simulating the structure of a material, for example, once the computation reaches the level of atoms, no further partitioning is possible.

With more efficient simulations, larger problems can be investigated,

with as many as tens of billions of unknown values to be determined. Potentially misleading simplifying assumptions, such as using one- or two-dimensional slices in place of a true three-dimensional problem, can be avoided. Also, faster calculations make it easier to check whether the results are sensitive to particular assumptions or details of the model. This sensitivity analysis can then help the researchers to reverse the problem and optimize the results.

similar technique can be used for systems of coupled differential equations — for example, if the charged density ρ responds to other quantities.

This kind of algorithm very quickly finds approximate solutions that are valid at short-length scales, comparable to the grid spacing. However, errors that spread over large distances require many iterations to be corrected, and as they evolve, the fine-scale details must change as well. Because it takes longer for a large grid to con-

verge to a solution, the computational demands of this kind of algorithm grow much more rapidly than the number of points in the grid.

The multigrid solution

A solution to this challenge, first proposed in the mid-1970s, is a class of algorithms called multigrid (see figure 7, above). The idea is to solve the problem on both fine and coarse grids, passing information back and forth between them. The

Tools for TOPS

TOPS researchers support a variety of tools for solving different kinds of problems. These “solvers” can be divided into five categories:

- **Linear solvers** analyze large systems of equations in which the response is simply proportional to the driving forces or source terms. This method will always be valid for looking at small perturbations of the system, and often provides a useful starting point even in more complex cases.
- **Eigensolvers** build on the linear solvers to identify the natural modes of a complete system. In a linear system, these modes don't affect one another, so researchers can describe the complete behavior by specifying the degree to which each one is present.
- **Nonlinear solvers** deal with the vastly more difficult situation that arises when the response

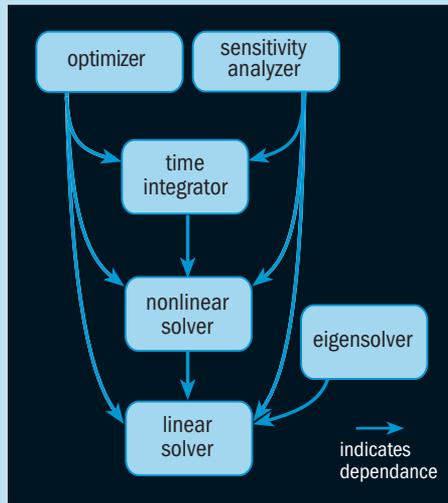


Fig. 8. The relationship between different solvers used in the TOPS program.

is not proportional to the driving forces or source terms, so that excitation of one mode spreads into others.

- **Time integrators** take a description of how the system changes with time and track its evolution over extended periods.
- **Optimizers** change the problem parameters to try to obtain a specified response. This response could be a numerical performance goal, or it could be an indicator of the overall agreement with an experimental result.

In addition to these five categories, **sensitivity analyzers** are built into each of the first four types of solver to indicate how much their behavior varies in response to a change in the inputs. This information then helps the optimizers to determine the best way of changing their inputs to achieve a change in output.

coarse grid can quickly reduce the long-length-scale errors. Indeed, since there are far fewer points in this grid, it can be solved much more quickly. This coarse grid solution is passed to successively finer grids using a “prolongation” scheme to interpolate between the coarse grid points. The solution on the finest scale is passed back to coarser levels, first using a smoother to remove noise, and then a “restriction” algorithm to reduce the number of grid points.

Researchers successively estimate solutions on different levels, while passing information between coarser and finer levels. They use various rules for moving up and down the levels. These multigrid methods are much more complex than a direct solution at a single scale, but they vastly reduce the number of times that the problem must be solved on the finest grid, which requires the most resources.

Modern multigrid techniques reduce the computational effort required so that it rises only in direct proportion to the number of points on the finest grid. This scalability lets researchers tackle much larger problems than otherwise would have been possible (see sidebar “The need for scalability,” p56). Researchers are also extending the multigrid techniques to deal effectively with complex grid geometries. Some of these extensions generate coarse grids from the geometry of the fine grid. The most advanced, “algebraic” multigrid techniques, however, create the coarse grid based on the numerical properties of the scientific problem to be solved.

Important phenomena occur over many orders

of magnitude in time, as well as distance. As discussed earlier, the motion of an electron in the magnetic field of a plasma tokamak varies on a time scale of picoseconds, while the pulse length may extend over several minutes.

Some researchers employ a further enhancement: an “adaptive” grid, the structure of which changes depending on the computational demands. For example, turbulent fluid flow generally introduces structure on many length scales, including very fine ones. An adaptive grid can introduce more points only at times and places where fine structure appears. Such adaptive grids are being actively investigated by, for example, SciDAC’s Terascale Simulation Tools and Technologies (TSTT) ISIC.

Unfortunately, aggressive use of adaptive grids sometimes poses challenges for parallel computation. For one thing, ensuring that the computational load is equally distributed between processors requires frequent shifting of the load as the grid changes, which can slow things down. Since TOPS aims to investigate large problems with very high parallelism, the researchers have not made extensive use of grids that change during the computation.

Types of solvers

TOPS researchers are building five types of solvers (see sidebar “Tools for TOPS,” above). Of these, the linear solver is the most fundamental, designed to solve problems of the general form $Ax = b$. This is “our meat and potatoes,” says Dr David Keyes, Principal Investigator. The apparent

A large part of the SciDAC effort is devoted to understanding how a complex system will behave under prescribed circumstances.

What's all the fuss about $Ax=b$?

Much effort in scientific computing aims to solve equations that can be written in the form $Ax=b$. This simple form belies great underlying complexity. The quantity x , for a start, consists of a series of values, one at each of the N points on the grid. For the guitar string in figure 3, for example, the elements of x are the vertical displacements of the string at positions along the string. The slope of the string at any point is computed from the difference in the displacement at neighboring sites. The curvature, which determines the restoring force on a particular segment of string, is determined from the change in slope between neighboring points. This can be expressed in terms of the difference of the displacement

of that segment and the average displacement of its neighbors on each side.

The forces on each of the N segments are computed as a weighted sum of the displacements of all segments of the string. In general, the N computations can be written compactly as a matrix product Ax . The $N \times N$ elements of the matrix A each express the weight with which a particular segment contributes to the force on another segment. Problems that are linear can always be expressed as such a weighted average.

When researchers try to solve larger problems by increasing the number of points in the grid, the matrix A can become enormous. If there are 1 million points on the grid,

for example, A could have 1 trillion elements. Fortunately, as in the guitar string example, each element is often affected by only a few others, such as nearby points on the grid. For this reason, researchers in scientific computation have worked hard to solve problems that involve such sparse matrices.

Linear solvers are used to solve problems of the form $Ax=b$, in which b represents an external condition (such as the plucking of the string). Eigensolvers build on the linear techniques to find the set of eigenmodes that solve the related problem $Ax=\lambda x$. An extraordinarily powerful feature of linear problems is that any solution can be expressed as a weighted sum of such eigenmodes.

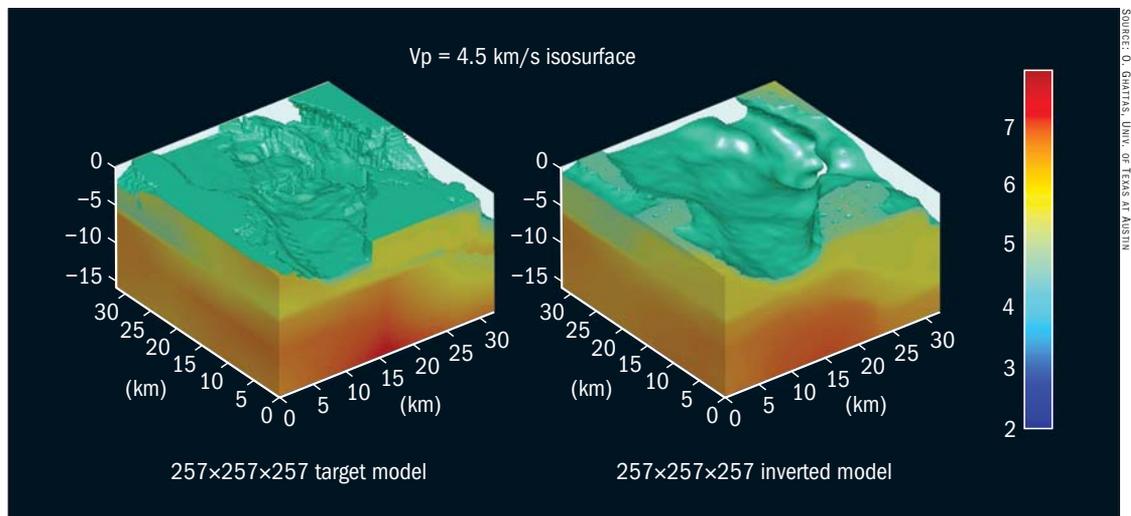


Fig. 9. Many important problems require researchers to solve a PDE system repeatedly. In the example above, which won the Gordon Bell prize at SC2003, the long-term goal is deducing the geological structure under the Los Angeles basin using observations of seismic waves made during earthquakes. The researchers validated their method by first simulating the observations that would arise from a standard model of the basin (left). They then used optimization algorithms to reconstruct the structure from the data (right).

simplicity of this equation masks a deep complexity (see sidebar “What’s all the fuss about $Ax=b$,” above). Linear solvers — and their close cousins, eigensolvers — efficiently find solutions for such matrix problems.

Solving linear equations consumes the vast majority of the computational resources for most problems. In most cases the original PDEs are nonlinear. Nonlinearity means that, unlike the

vibrations of the guitar string, different solutions can’t be combined to get a new solution. The complete solution of nonlinear systems is much more complex than for linear systems.

The TOPS program has also adapted techniques for solutions evolving in time.

The highest components on the chain are sensitivity analysis and optimization. Sensitivity analysis lets researchers explore the neighbor-

hood of a single solution with nearby solutions from nearby initial and boundary conditions and parameters, to understand not merely a single output, but its dependence on all of the inputs. Optimization builds further on sensitivities and is discussed in the next section.

TOPS researchers are also working to provide common software interfaces so that the most effective tools can easily be adapted to different problems and different hardware environments. The TOPS Solver Component is the latest development in this aspect of their work.

Optimization problems

A large part of the SciDAC effort is devoted to understanding how a complex but precisely defined system will behave. Frequently, however, this forward simulation problem is only the beginning. Researchers are often seeking control parameters, boundary conditions, or design values to produce a desired outcome. This outcome might be the best possible match to measured data, or the best performance of a new design. Improving the efficiency of the forward simulation makes this faster, but the inverse problem needs to be explicitly addressed.

Say, for example, that astrophysicists have observed the time course of neutrino flux from a supernova, and would like to know what kind of star collapsed to give such a signature. The most direct approach would be to run repeated “forward” problems, tweaking the parameters each time to improve the match between the simulation results and the experiment. More sophisticated optimization algorithms automatically search the parameter space to find the optimum match in a relatively small number of iterations. Despite the efficiency of these optimizers, inverse problems tend to be much more computationally intensive than the corresponding forward problem. In fact, says Dr Omar Ghattas of the University of Texas at Austin, “Optimization of terascale problems is often a petascale problem [1000 times more demanding].”

Even if a configuration is found that produces predictions that match the desired result, researchers need to know how unique that configuration is. Indeed, for the most challenging problems like supernovae, researchers have barely begun to address such inverse problems, since the original problem is so difficult.

In more modest cases, however, TOPS researchers are applying terascale computations to inverse problems, together with colleagues in the TSTT project and elsewhere. To make this easier, they insert “hooks” into the original code to let them monitor how sensitive the end results are to changes in the initial parameters. They do this

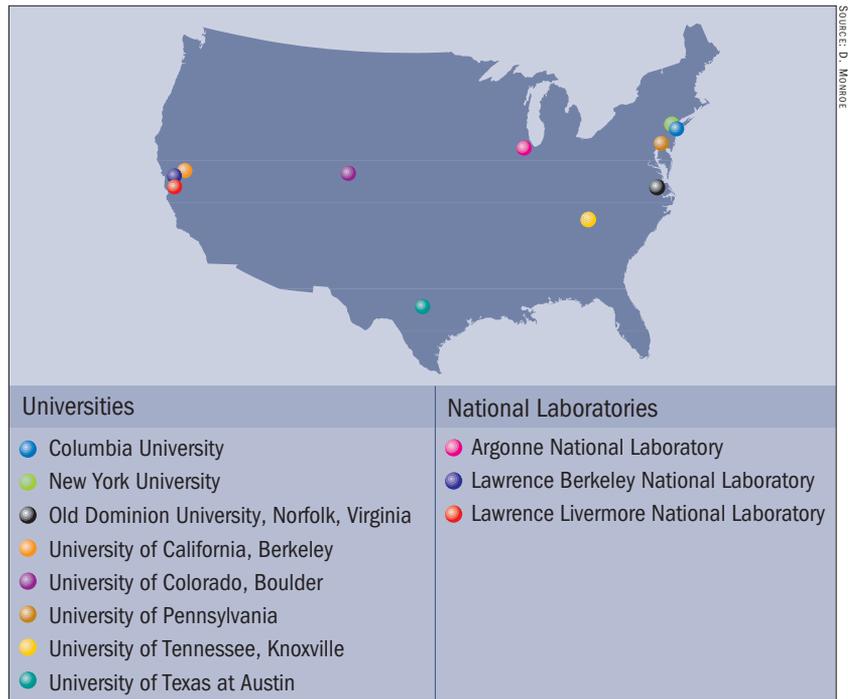


Fig. 10. Researchers from across the US are contributing to the TOPS research program.

by including an additional, spatially varying field to the simulation. Although this increases the time needed to perform the initial computation, they can use this adjoint field to calculate the derivatives immediately. In this way they can better assess how to change the parameters to make the results approach those desired.

To compute the overall sensitivity to changes, the researchers need to know the adjoint field at each grid point. For less complex cases, this can be written as a simple mathematical expression, and can be found by mathematical differentiation. In many cases, however, the sensitivity cannot be expressed analytically. TOPS researchers have developed automatic differentiation techniques that remove the mathematical burden of calculating a derivative.

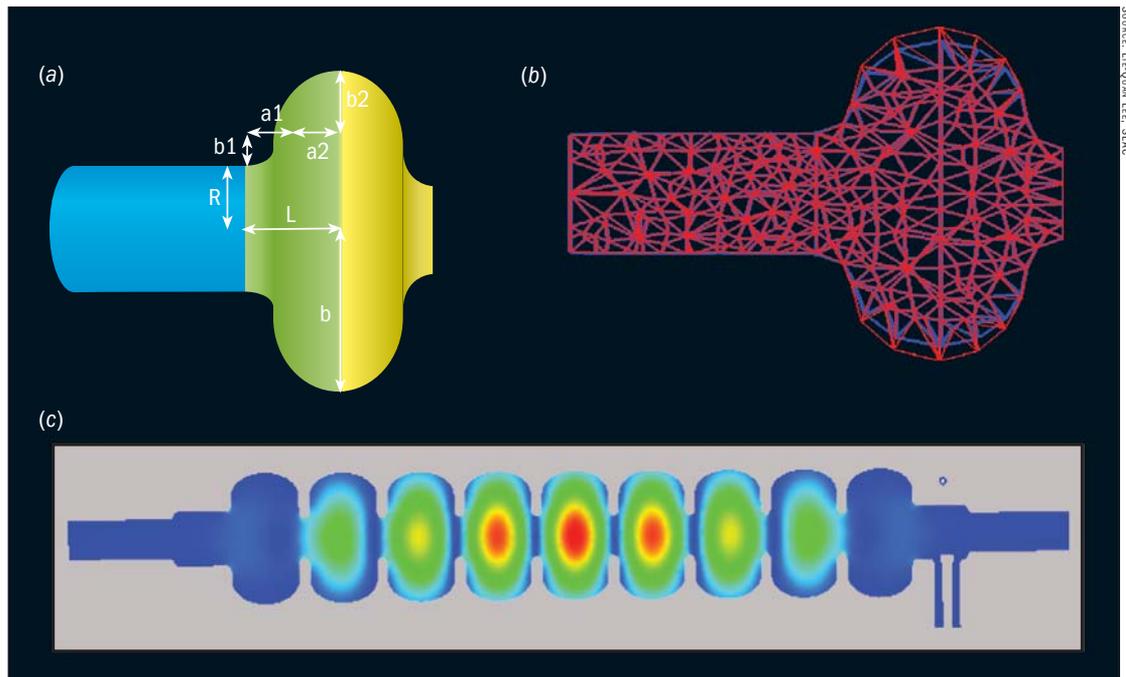
One important class of inverse problem is the determination of the underlying structure that might give rise to signals observed. One such study, funded by the National Science Foundation and the Department of Energy, was awarded the 2003 Gordon Bell prize (see figure 9, p58). Based on seismic data, researchers sought to deduce the geological structure under the Los Angeles basin. As an initial validation of the technique, they used the Southern California community velocity model for the seismic properties of the basin, and simulated the seismic waves expected in response to earthquakes. As shown in figure 9, they were able to recover a slightly blurred version of the original structure. Later they expect to use actual data from a variety of earthquakes to determine the struc-

“Optimization of terascale problems is often a petascale problem.”

DR OMAR GHATTAS
University of Texas at Austin

“SciDAC’s number-one product is communication. TOPS strives to understand the algorithmic needs of physicists and to serve as a point of contact for them for solvers within the applied mathematics and computer science community.”

DR DAVID KEYES
Principal Investigator



SOURCE: LEE-QUAN LEE, SLAC

Fig. 11. To design the next generation of particle accelerators, physicists will rely more than ever on advanced simulation tools that not only let them evaluate a proposed design, but make it easy to optimize the design for better performance. **Fig. 11(a).** The detailed design of the end cell of an early proposal for the new International Linear Collider includes many dimensions that can affect its performance. **Fig. 11(b).** When researchers change the geometry of the cell, they must create a new grid on which to perform the simulation. This grid differs only at the boundaries. **Fig. 11(c).** The forward simulation shows the magnetic field in the operating accelerator. To optimize the choice of the various dimensions, this simulation must be repeated for each new option, but the decision can be made easier by anticipating this need in the original simulation. (See article on p12.)

ture of the basin. Similar problems arise in medical and other imaging techniques.

Another important example of an inverse problem is the optimization of apparatus design. This is illustrated by an early proposal for the next-generation International Linear Collider (ILC) done in collaboration with the Advanced Computations Department at the Stanford Linear Accelerator Center and members of the TSTT project. In this case, rather than trying to match the simulation results to an experiment, the researchers are trying to maximize a measure of the magnetic field within the cavity of the accelerating structure, while keeping the resonant frequency of the cavity tuned to the value that drives the electrons most efficiently. The externally-controlled parameters that can be modified include parameters describing the shape of the cavity (see figure 11).

Like the seismic problem, solving this design problem is made much easier by including sensitivity analysis in the original solver. However, in design problems, the effect of the parameters on the output can be much more indirect, making the details of the solution rather different. In addition, changing the dimensions of the cavity requires that the grid be altered as the parameters are changed. So far, the researchers have explored small changes in the dimensions of the cavity, which can be

accommodated by changing the dimensions of cells on the periphery of the simulation (see figure 11(b)). Larger changes that would require complete regriding are an area for ongoing research.

Outreach

A major goal of the TOPS program is to help users to identify the best tools for the job. “SciDAC’s number-one product is communication,” says Dr Keyes, adding that the task of TOPS is not primarily to develop new algorithms, but to make it easy for users to adopt the best procedures known. Otherwise, even if users know, for example, that multigrid is the most efficient technique, the programming complexity may deter them from using it. Users vary significantly in their computational sophistication, so ideally a tool will be easy for new users to get started with, but will allow more advanced users to employ it in a more sophisticated way.

Dr Keyes describes the three ideal stages of collaboration between TOPS tools and their scientific users as follows. In the first stage, users adopt the basic linear tools. Although these tools can only digest linear problems, the scientists have generally spent large amounts of effort casting their problems into a linear form. Such linear calculations typically consume 90% of the

An important future direction is in climate-change research.

Our favorite things

TOPS project leader Dr David Keyes adapted the Rodgers and Hammerstein song “My Favorite Things” to include many of the issues the team faces.

F-cycle multigrid, Krylov subspaces,
Saddlepoint solvers, reduced-order bases,
Cache-friendly, optimized “node code” that sings –
These are a few of our favorite things.

High-order schemes with flux-surface alignment,
Low-order schemes with adaptive refinement,
Users leave happy, whatever they bring –
These are a few of our favorite things.

*When the code hangs,
When the queue’s full,
When support is sad,*

*We simply remember our favorite things...
...and then we don’t feel so bad!*

Unstructured meshes and multi-rate physics,
PETSc* and Hypr† on farms running Linux,
Massively parallel flow modeling –
These are a few of our favorite things.

Multiple disciplines working together,
Tracking neutrinos, predicting the weather,
Probing stability of plasma rings –
These are a few of our favorite things.

*When AG‡ drops,
When the plane’s late,
When meetings drive us mad,
We simply remember our favorite things...
...and then we don’t feel so bad!*

* PETSc: the Portable, Extensible Toolkit for Scientific Computing, pronounced PET-see.

† Hypr: the High-Performance Preconditioner Library

‡ AG: the Access Grid, an Internet-enabled collaboration medium

computational budget, so using more efficient algorithms has a big impact.

In a second stage of collaboration, the computer scientists work with the scientists to recast the nonlinear problems. “We can use their existing solver as a preconditioner,” Dr Keyes says. The preconditioner doesn’t solve the problem, but modifies elements of the matrix so that the solver’s job is easier. The efficient nonlinear tools developed in the TOPS program can then have an even greater impact.

Finally, the two groups of researchers can work to improve the discretization of the fundamental problem. In these latter stages, close collaboration between the researchers who know the science and those who know the algorithms is essential.

Looking to the future

Modern supercomputer platforms consist of clusters of powerful machines, each having only a small part of the data stored locally. This distributed memory strongly constrains the sorts of algorithms that can efficiently run on these clusters. Fortunately for TOPS, this preferred architecture has been rather stable since the program began, allowing researchers to concentrate on using the machines effectively.

Some aspects of this platform could be changing soon, however. In particular, computer scientists are exploring the use of “vector” machines,

which perform many similar operations simultaneously. Even if no such paradigm shift occurs, the researchers will be working hard to tune their code for steadily improving hardware.

TOPS researchers also work closely with researchers exploring applications. As they enter new fields, they may need to develop new types of algorithms. An important future direction is in climate-change research. Huge amounts of effort have gone into the existing codes for climate, taking advantage of the very short vertical dimension of the atmosphere and oceans relative to their horizontal extent. So far, the researchers have a huge investment in specialized codes, so adapting the general-purpose codes that TOPS specializes in would not be of immediate benefit to them. But as ever more detailed simulations — and greater confidence — are required, Dr Keyes believes that collaboration on this important subject is almost inevitable. ●

Further reading

The TOPS project www-unix.mcs.anl.gov/scidac-tops.

V. Akcelik, G. Biros, O. Ghattas, D. Keyes, K. Ko, L. Q. Lee, and E. Ng 2005 Adjoint methods for electromagnetic shape optimization of the low-loss cavity for the International Linear Collider *J. Phys.: Conf. Ser.* **16** 435–445.

V. Akcelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, and B. van Bloeman Waanders 2006 Parallel PDE constrained optimization *Frontiers of Parallel Computing* eds M. Heroux, P. Raghaven, and H. Simon (SIAM, to appear).