UCRL--53916

DE91 007188

ENERGY SCIENCES SUPERCOMPUTING 1990

Meeting the Computational Challenge

Predicting global climate change, modeling the basic interactions that occur in nature, and computing the evolution of a fusion plasma—these are some of the national Grand Challenge problems being addressed with the supercomputing facilities at the National Energy Research Supercomputer Center (NERSC).

Our intention with this publication is to demonstrate supercomputing's vital role in solving problems in the energy sciences. To show this, we asked representative users of the NERSC supercomputers to write about how supercomputing affects their areas of research. As you read these articles, you will repeatedly encounter the theme that today's solutions would not have been possible without supercomputers.

Also apparent will be similarities in the computational models and techniques used across the various disciplines that are discussed here. The articles on lattice gauge theory, nuclear interactions, and chemical reactions all describe use of the Monte Carlo approach. Particle simulation methods are cited both in modeling transport in a plasma and in simulating the Superconducting Super Collider. Computing the evolution of a plasma and predicting global warming both use the equations of fluid dynamics. Common to many of these fields are chaotic phenomena, which can be described by their own abstract theory, the elucidation of which often requires the supercomputer.

Virtually all of the applications discussed here are suitable for parallel processing. Both the Monte Carlo and the particle methods involve repetitive operations that may be carried out independently and simultaneously. The parallelism inherent in fluid dynamics derives from the fact that physical space is partitioned into a large number of regions in which quantities of interest are evolved. Parallelism is a recurring theme throughout this publication; one article comprehensively addresses various aspects of parallel processing.

Another recurring theme is the need for even greater computing capability in the future if we are to provide realistic solutions to today's Grand Challenge problems. The degree of spatial and temporal resolution and the sophistication of the computational models are limited by attainable memory size and processing speed. Meeting tomorrow's computational demands will require more advanced parallel processors. In addition to pursuing vector multiprocessors, we recognize the tremendous inroads that massively parallel computers are making in the supercomputing field. Some of the applications discussed here are already running on massively parallel machines and are executing more efficiently in that environment than on conventional supercomputers. As massively

---

supercomputer: the fastest, most advanced computer at a given time.

high-performance computing: the full range of advanced computing technologies, including existing supercomputer systems, special-purpose and experimental systems, and the new generation of large-scale parallel systems.

Grand Challenge: a fundamental problem in science or engineering that has broad economic and scientific impact and that can be advanced by applying high-performance computing resources.

parallel computing continues to evolve rapidly, it is inevitable that more and more applications will demand that resource.

The term "supercomputing" (which is slowly being replaced by "high-performance computing") involves much more than just the supercomputer engine; it refers to the whole computing environment. This point is apparent when examining the scope of the federal government's High Performance Computing Initiative. Producing software commensurate with today's high-performance hardware is one of the biggest challenges facing the computational scientist. It is clear, for example, that software technology will pace advancement in parallel computing.

The rapid development of microprocessors has enabled the use of desktop systems for problems that used to require the supercomputer. It is not atypical for one phase of a problem to require a supercomputer and for another phase to demand the versatility of a desktop system. Such diversified applications make effective use of a distributed computing capability. An ultimate goal is to provide a "seamless" computing environment—that is, one in which the hardware invoked to execute the computational task is transparent to the user.

To interpret the results of three-dimensional simulations requires advanced visualization techniques. Tools in this area are becoming more prevalent, but there is still a great deal of work to be done. Both advanced visualization and distributed computing are successful only to the extent that the network provides the necessary bandwidth. The recent upgrade of the Energy Sciences Network (ESNET) to fiber-optic technology allows data to be transmitted almost 30 times faster than previously, and ESNET's multiprotocol capability adds versatility in servicing different research communities. Equally important is the network's role in enabling scientists at different locations to collaborate effectively.

One vital supercomputing resource of the future is often overlooked: the human component—in particular, today's youth. It is crucial that we seek out tomorrow's potential scientists and introduce them to the world of supercomputing. Programs to accomplish this are described in this publication.

The idea for this publication originated in part with John Killeen, who directed NERSC during its first 15 years. In 1989, John suffered a debilitating stroke, which necessitated his stepping aside. As we search for a new director, I want to take this opportunity to pay special tribute to John and to thank him for his pivotal role in giving NERSC the outstanding reputation that it enjoys today.

In summary, the increased capability of supercomputers has enabled the solution of many important problems in the energy sciences, problems that seemed intractable five years ago. As the supercomputing field continues to grow, we look forward to growing with it and to providing the computational resources needed for solving the Grand Challenge problems of the 1990s.

Robert R. Borchers
Associate Director, Computations
Lawrence Livermore National
Laboratory

Lattice Gauge Theory: Probing the Standard Model

Gregory Kilcup, Ohio State University

Simulations of quantum field theory require four-dimensional lattices and millions of equations, as well as thousands of hours of supercomputer time.

What is the mass of the proton? How long do pions, rhos, and other elementary particles live before they decay into other particles? For decades we have known the answers to these and similar questions from experiments. And for decades elementary particle theorists have despaired of being able to explain those experimentally attained answers on the basis of first-principles calculations. However, the arrival of supercomputers, such as the Crays at the National Energy Research Supercomputer Center (NERSC), is changing all that. Problems that once seemed impossible to solve theoretically are now yielding to numerical attack.

The background for these calculations is two decades of great progress in our understanding of the basic particles and forces. Over time, the particle physics community has developed an elegant and satisfying theory, the "Standard Model," which is believed to be capable of describing all the phenomena that can be produced in today's high-energy accelerators. The ingredients of the Standard Model are matter particles—quarks and leptons—and the three forces through which they interact: electromagnetic, weak, and strong. (Gravity, although not included in the Standard Model, is so much weaker than the other forces that it is irrelevant for experimental particle physics.) The leptons—electrons and their cousins—are susceptible only to the weak and electromagnetic forces, while the quarks are subject to the strong force as well as to the other two. This picture immediately gives a qualitative understanding of the structure of the atom. The strong force binds quarks together into "hadrons," such as protons and neutrons, which stick together inside the atomic nucleus, while the more feebly interacting electrons are free to roam about the whole atom.

The mathematical description of each of the forces is essentially the same. In each case, the matter particles interact by exchanging force particles. The particle that creates the electromagnetic force between charged particles is the photon, while the weak interactions are mediated by the recently discovered W and Z particles. The strong force is created by particles that are whimsically named "gluons," so called because the strong force is "sticky" in comparison to the other forces.

The feature that makes the strong force so different from electromagnetism has to do with the kinds of charge involved. One can state the electric charge of a particle with one number (for example, +1 for a proton), while for the strong force one needs three numbers. In another act of whimsy, and for lack of a more natural name, these three numbers are said to characterize the "color charge" of the quark, analogous to the three primary colors. As a result, the quantum theory of quarks and gluons has been christened quantum *chromo*dynamics, or QCD.

How does one turn this description into solid quantitative predictions for experiments? For processes involving only the weak and electromagnetic interactions, one can do an accurate calculation with pencil and paper (and perhaps a good symbolic algebra program). For most quantities involving the strong interactions, however, the usual techniques are not adequate. The root of the difficulty is the fact that QCD is a nonlinear theory, and the various modes can interact in complicated ways. The situation is somewhat analogous to that in hydrodynamics, where the interactions are very simple at

small scales but can give complicated turbulent solutions on a larger scale. To perform first-principles calculations in QCD, one has to abandon the pencil-and-paper approach and resort to the more brute force approach of large-scale computing. The computational approach goes under the name "lattice QCD" or, more generally, "lattice gauge theory."

The starting point of the computer assault is the Feynman path integral formalism. The quark and gluon degrees of freedom are described by fields [usually labeled $\psi(x)$ and $A(x)$, respectively], which more or less give the probability of finding a quark or a gluon of a particular color at a particular point x of space and time. Then any experimentally measurable quantity can be extracted from expressions of the form

$$\prod_x \int d\psi(x) dA(x) \exp[-S(\psi,A)]\phi(A,\psi) \quad .$$

The function S in the exponential contains the details of the short-distance interactions and is the same for all calculations, while the function ϕ is chosen according to the particular quantity one wants to calculate, whether the mass of the proton or something else. For example, if one took $\phi = \psi(x)\psi(y)$, evaluating the above expression would give the answer to the question: given that a quark existed at a certain place and time x, what is the probability of finding it at some other place and time y?

To evaluate the path integral, one has to perform a sum over all possible configurations of the fields ψ and A, where by field configuration we mean specifying the values of ψ and A at all points of space and time. This is a functional integral, or infinite dimensional integral. To put the expression on a computer, one obviously makes approximations. First, one replaces continuous space-time with a discrete lattice of points. Second, one considers not the whole infinite range of space and time but only a finite chunk of it. In this way, the mathematically intractable expression above is approximated by a large—in fact, *very* large—but finite number of ordinary integrals. Finally, it turns out that the interaction function $S(\psi,A)$ is an expensive nonlocal function to compute, and most calculations today use a well-motivated but uncontrollable approximation for S, the "quenched" approximation. There is no problem of principle in doing unquenched simulations, but they require several orders of magnitude more computer time.

To set the scale of the problem, typical dimensions for a lattice in today's simulations are 16 points in each of the three space

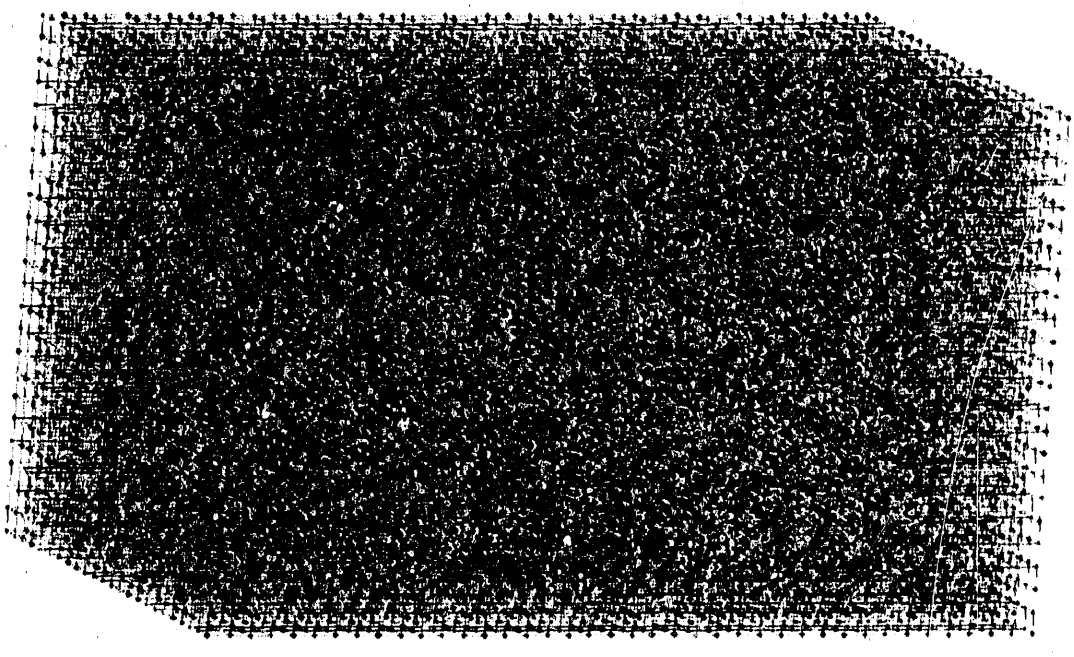


Figure 1. A three-dimensional "slice" (with $24^2 \times 40$ sites) out of a four-dimensional lattice ($24^3 \times 40$ sites) developed for quantum chromodynamics calculations. The dots give the color of the quark field (red, yellow, blue, or mixtures of these) at each of the 23,000 sites shown in this slice.

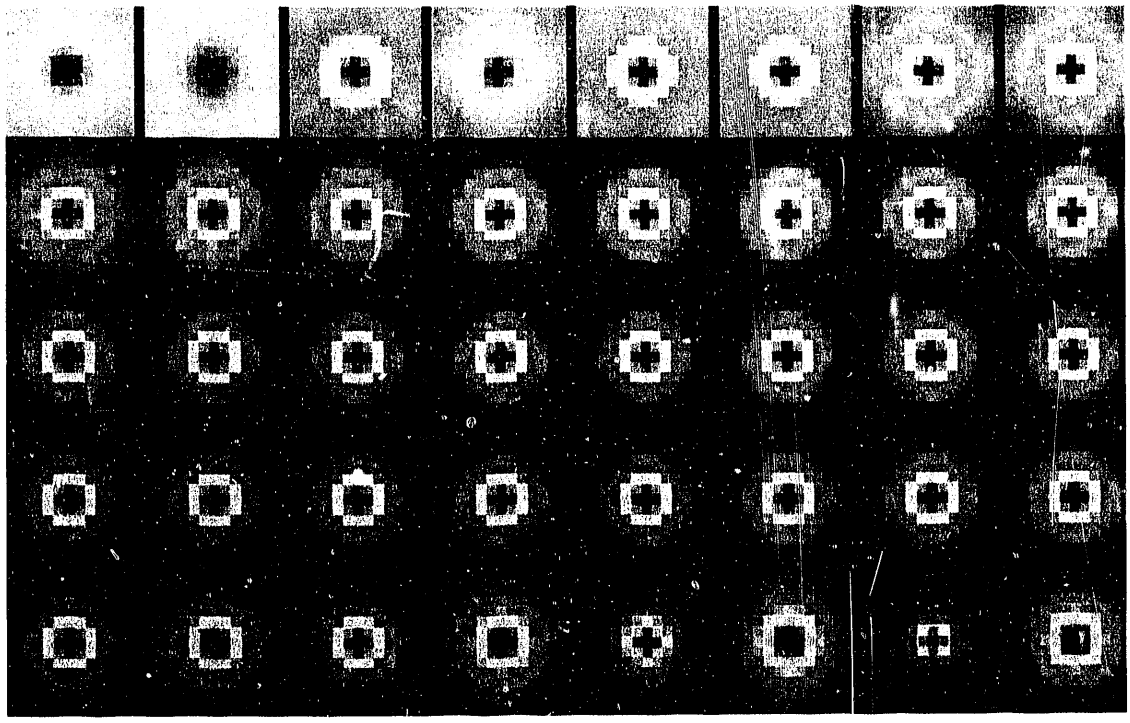


Figure 2. A picture of a pion (a bound state of a quark and an antiquark) as a function of space and time. Each of the 40 squares is a spatial slice (x-y, with the z direction suppressed) of the pion at a given time, with time progressing from left to right and from top to bottom. The pion is created at the first time slice, in the upper left corner. Within each time slice, if the antiquark is located at the central point, what is the probability of finding the quark at a given spatial point? Red denotes the highest probability, dark blue approaches zero probability, and the other colors range in between.

directions and 40 points in time. (For technical reasons it is convenient to have more points in the time direction than in the space directions.) The quark and gluon fields each have several components, and to specify them, one needs 38 numbers at each lattice site. Roughly speaking then, the goal of a typical lattice QCD calculation is to compute an ordinary integral, which happens to live in a space of $16^3 \times 40 \times 38 \approx 6$ million dimensions. Of course, that is not the end of the story—one has to check that the lattice volume is big enough that boundary effects are not important, and that the distance between lattice sites is small enough that the essential physics is not distorted by lattice artifacts. This means having to redo the same calculations on lattices with an ever-increasing number of points. At NERSC we have been able to perform calculations on the largest lattices to date—$24^3 \times 40$ and $32^3 \times 48$—which correspond to about 21 million and 60 million dimensions respectively. To give an impression of the size of such lattices, we show in Figure 1 a typical three-dimensional ($24^2 \times 40$) slice of a four-dimensional $24^3 \times 40$ lattice. The dots give the color of the quark field (red, yellow, blue, or mixtures of these) on each of the 23,000 sites shown in this slice.

Of course, one cannot compute an integral of several million dimensions by naive methods. Even if we discretized the integration region in the crudest possible fashion, putting down only two points per axis, one would have to sum up the result of $2^{1,000,000}$ function evaluations, more than could be done on any imaginable computer. Fortunately, the presence of the exponential factor in the integrand means that almost all field configurations contribute only negligibly to the final answer. Obviously, one wants to evaluate the integrand only in the region that matters.

The technique of choice is a Monte Carlo method. We treat the $\exp[-S]$ as a probability distribution; that is, out of the space of all possible field configurations, we randomly select individual configurations, with probability proportional to $\exp[-S(\psi,A)]$. Then, having collected a whole ensemble of configurations, the integral we want is given by the average value of $\phi(\psi,A)$ in the ensemble. An advantage of this method is that with the same ensemble of configurations one can simultaneously compute the value for many different observables ϕ. On the other hand, the unavoidable disadvantage of a statistical method is that one may require large numbers of samples before some observables emerge from the noise.

In practice, then, there are two phases to a lattice gauge theory calculation: configuration generation and the measurement of observables. In the first phase, the fields are allowed to make a random walk through the space of all possible configurations, making small random changes in the fields in such a way that the probability of hitting a particular configuration (ψ,A) is proportional to $\exp[-S(\psi,A)]$. There are a number of algorithms for performing this random walk, the most popular of them having names such as Metropolis, "heat bath," Langevin, and molecular dynamics. However the configurations are generated, one lets the simulation run for a long time, periodically drawing configurations out of the stream for analysis. On this ensemble of selected configurations, one computes the observables ϕ. Typically this is the most expensive part of the simulation, at least when the observables involve quark fields. To study the way quarks move, one has to invert a differential operator, which depends on each particular field configuration. On the lattice the differential operator becomes a large sparse matrix, and one has to solve a large set of linear equations. At NERSC, a typical number of equations would be a few million. Since the system is sparse, iterative methods are preferred; the most popular are conjugate gradient and its cousins. After the observables on each of the configurations have been computed, the final stage is to compute averages and correlations across the ensemble. The number of configurations and observables is typically in the tens or hundreds, so this stage can be performed on a personal workstation.

What sort of quantities can be calculated in lattice QCD? A simple quantity one might like to know is the size of an elementary particle. For example, a pion is a bound state of a quark and an antiquark, and one might reasonably ask what is the average distance between the constituents. The answer is shown in Figure 2, which gives a picture of a pion as a function of space and time. Each of the 40 squares is a spatial (x-y) slice through the pion at a given time, with the z direction suppressed. Time runs from left to right and from top to bottom, and there are Dirichlet boundary conditions in time at the ends of the lattice. The spatial boundary conditions are periodic in x, y, and z. At the first time slice, in the upper left corner, a pion is created. We then ask for the probability of finding a quark at a given place in the lattice, having nailed down the antiquark at the central point. Red denotes the highest probability, dark blue approaches zero probability, and the other colors range in between. The distribution depends on time because it takes some time for the pion to settle down to its equilibrium state after being created. Likewise, there are edge effects at early and late time, where the particle is bouncing off the boundary conditions. But in the intermediate region one can see a clear portrait of a pion. To set the scales involved, the smallest squares of color are about 10^{-16} m across, and the time between each successive snapshot is about 5×10^{-25} s.

We can play a similar game with the proton, which is made up of three quarks. Now we nail down two quarks and ask where the third quark likes to stay. Figure 3 shows the history of a proton, where we have fixed two quarks diagonally on either side of the central lattice site within each time slice.

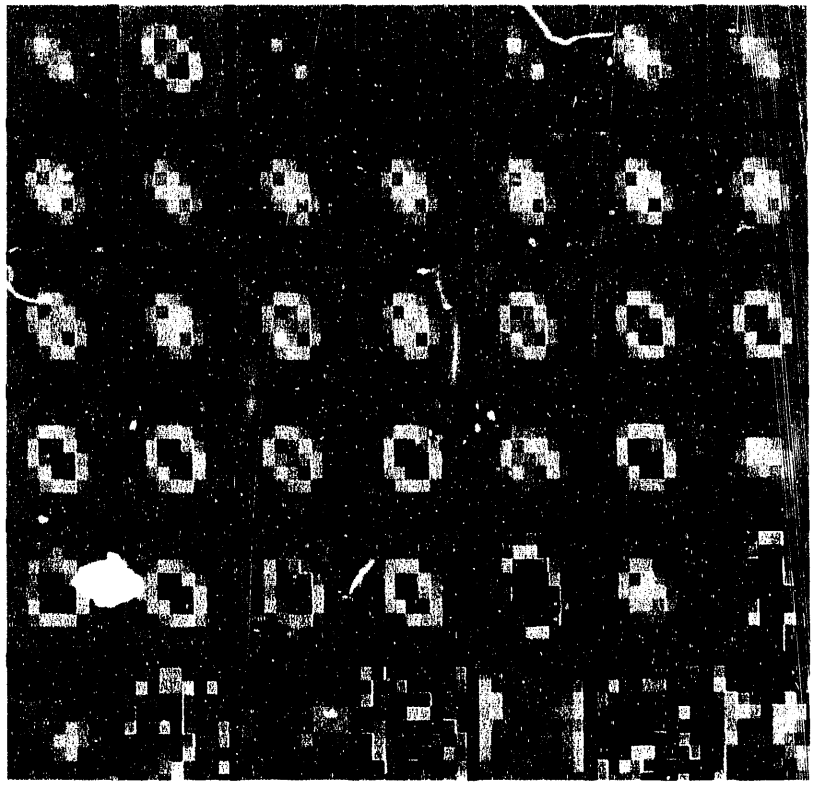


Figure 3. A picture of a proton (composed of three quarks) as a function of space and time. Two of the quarks are fixed diagonally on the central lattice site within each of the 42 time slices. As time progresses (from left to right and top to bottom), where is the third quark most likely to be found? Red denotes the highest probability of being the quark's location, dark blue indicates near-zero probability, and the other colors range somewhere in between. A stable picture emerges in the middle of this lattice (rows 3 and 4), but statistical noise dominates after that.

Again, one sees some sort of stable picture in the middle of the lattice (rows 3 and 4), but this time the picture degenerates into noise soon after, because the statistics for this particular observable weren't good enough to resolve the proton for all times.

Other quantities one can look at in lattice QCD are harder to depict but are of much more theoretical importance. As mentioned above, one of the largest and most interesting calculations has been mounted these past two years at NERSC. As part of the U.S. Department of Energy's "Grand Challenges" program, a group of scientists—including the author; Claude Bernard of the University of California, Los Angeles; Rajan Gupta of Los Alamos National Laboratory; Steve Sharpe of the University of Washington; and Amarjit Soni of Brookhaven National Laboratory—was granted some 16,000 hours of Cray time to study the properties of kaon decay. Like pions, kaons are bound states of a quark and antiquark. They live a short while (about 10^{-10} s) and then decay, most of the time into pions. The decay process involves not only the effects of QCD but also, more importantly, effects from the weak interaction sector of the Standard Model, whose properties are much more poorly known. By doing a very careful study of kaon decay, we can shed light not only on QCD but also on certain aspects of the other forces. For this, the computational approach is proving invaluable, since it is the only reliable and well-understood tool available.

Large-scale computer calculations, such as those for kaon decay, are the theoretical counterparts to the high-energy-accelerator experiments being done at Fermi National Laboratory in Illinois, at CERN in Switzerland, and perhaps sometime soon at the Superconducting Super Collider in Texas. These experiments provide raw numbers, but to extract the implications for the Standard Model, one needs to perform a difficult computation. With detailed-enough experiments and ever-more-powerful computers for lattice QCD calculations, we will be able to fully explore all aspects of the Standard Model of particle physics. Perhaps the most tantalizing prospect is that both the lattice QCD calculations and the experiments will become sensitive enough to detect small discrepancies between the predictions of the Standard Model and the real-world answers. This would be one of the best ways of getting clues about physics beyond the Standard Model—about forces we have yet to detect.

So what is the mass of the proton? To within 20% or so, lattice gauge theory tells us it is what we always knew it to be. But with increasing computer power we expect not only to understand all about protons but also perhaps to get a glimpse of "nonstandard" physics as well. ■

Gregory Kilcup is Assistant Professor in the Physics Department at the Ohio State University. Educated at Yale (B.S., 1981) and Harvard (Ph.D., 1986), Professor Kilcup was a Research Associate at Cornell University and a junior faculty member at Brown University before joining OSU this year.

Supercomputing for the Superconducting Super Collider

Yiton Yan, Superconducting Super Collider Laboratory, Dallas, Texas

Supercomputers are used to simulate and track particle motion for a million turns around the collider rings. These numerical studies will aid in determining the best aperture for the proton beams.

The need to understand the most basic structure of matter requires a major advance in the energy frontier of particle accelerators. The Superconducting Super Collider (SSC), a powerful instrument currently under design, will fill this need.

The SSC will be a proton-proton collider with design luminosity of 10^{33} cm^{-2} s^{-1}. Its purpose will be to accelerate and guide bunches of ultra-high-energy protons into collision. Two tightly focused proton beams, each with an energy of 20 TeV, will move in opposite directions around a racetrack-shaped orbit. As the protons collide, their constituents can interact, thereby releasing enormous energy and revealing a level of detail that has been previously unachievable. Direct evidence about the most fundamental physical forces and entities will be carried by the collision products and may be captured in the sophisticated detectors that will surround the interaction regions. Since the probability of interaction will be comparatively low, the proton beams can be recirculated to collide repetitively for many hours without significant attenuation. Thus the SSC will be constructed as a pair of storage rings capable of holding the tightly confined proton beams on closed paths for a day or more without replenishment. The rings confining the proton beams will be about 54 miles in circumference and will be housed one above the other in an underground tunnel.

A system of superconducting electromagnets will guide the protons around the desired orbit through a beam pipe. This magnetic confinement system will consist of a periodic array of bending (dipole) and focusing (quadrupole) magnets, with the bending magnets establishing the curvature of the orbit and the focusing magnets confining the protons to a narrow region within the vacuum tube. The operating cycle of the SSC will begin with the collider magnets maintained at low current for about 40 minutes while the proton beams are loaded into the collider rings from lower-energy accelerators. With injection complete, the acceleration system powered with radio frequency (rf) waves will be activated. The slow increase in the beam energy will be accompanied by a corresponding increase in the strength of the bending and focusing magnets, thus keeping the position of the beam orbit fixed while also keeping the proton beams synchronized with the accelerating system. This synchronous acceleration will be complete when the protons reach their final energy of 20 TeV. The accelerating system will then be turned down, and the beams will be steered into collision. The resulting reactions can be studied for a day or more before the beams are depleted sufficiently so that the cycle must be repeated. During the collision phase, some of the protons will be lost because of catastrophic nuclear collisions. In addition, the dynamics of the surviving beam particles will be perturbed by the electromagnetic interaction between the two beams at each collision point.

Success of the SSC operating cycle will depend very much on the careful design of what is called the lattice, a detailed description of how the magnets of various sorts and strengths will be placed to form the confinement ring. The lattice encompasses both the physical arrangement and the powering or strength of the magnets. The most fundamental requirement for a good SSC lattice is that the proton beams have adequate lifetimes. Therefore, designing a suitable lattice requires understanding in detail the motion of the protons.

A key issue of proton beam dynamics, and thus of the SSC lattice design, is the "aperture," or the cross-sectional area within which the proton motion is stable. If this cross-sectional area is large (relative to the size of the proton beam emerging from the injector), the proton survival time will be relatively large. Clearly, one would like to have a large aperture to ensure successful operation of the SSC. This requires magnets that can provide a large region of uniform magnetic field, which in turn requires magnets that are relatively large and therefore expensive. Hence, the goal is to achieve the smallest possible magnet aperture (that is, the inside dimension of the vacuum pipe container) in which an adequate space—characterized by the term "dynamic aperture"—can be identified for stable motion of the protons in the beam. Therefore, one must study the dynamic aperture for each of the alternative magnet lattices under consideration. This is done by simulating the motion of the proton beam with numerical codes on supercomputers.

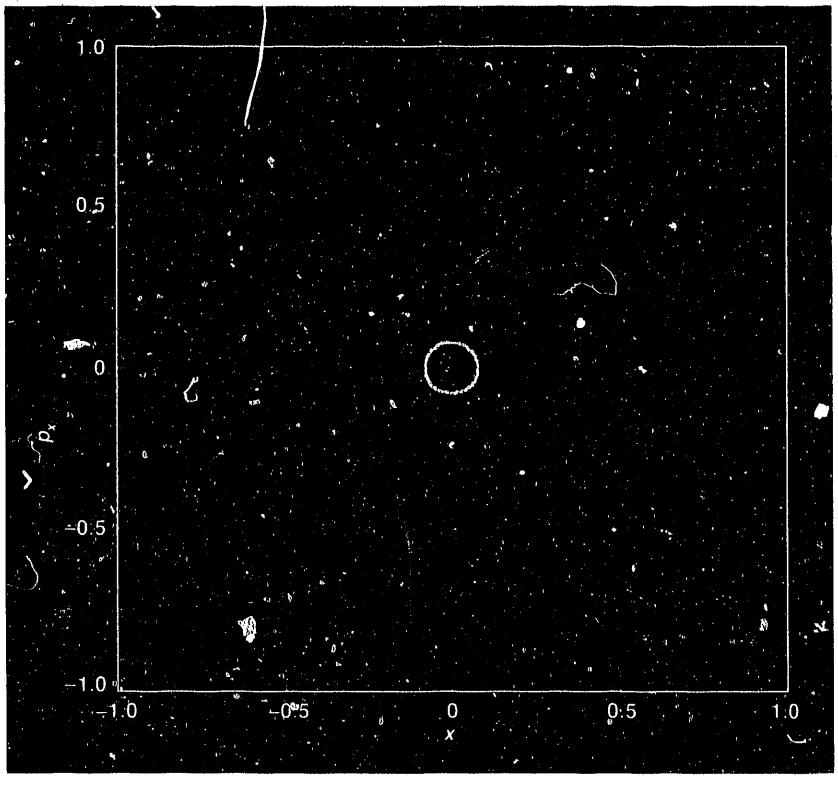


Figure 1. Phase space plot (p_x versus x) for four protons with different initial amplitudes, where x is a Floquet space coordinate and p_x is its corresponding Floquet space momentum; that is, they are normalized such that a proton with linear motion would trace out a circle. The variation in the amplitude traced out by a given proton serves as a diagnostic of accelerator nonlinearity of that proton's motion. For example, the protons here with the smallest initial amplitude (indicated in yellow) show so little nonlinearity that the data points merge into a solid line. However, the protons with the largest initial amplitude (in red) have correspondingly greater nonlinearity, so that the data points are more widely spaced in the circular band.

SSC Aperture Study

The confinement system to guide the protons around the desired orbit will consist basically of a periodic cell of dipole and quadrupole magnets. This structure is called a "linear" lattice if the dipoles and quadrupoles are ideal magnets and are perfectly aligned so that nonlinear beam dynamics is negligible in the proton motion. In a linear lattice, the protons undergo oscillations (betatron oscillation) while circulating around the desired orbit. Analytical techniques would allow an accelerator physicist to design a good linear lattice if things were this simple. However, the protons injected into the SSC rings will not all have the same momentum and so will not all have exactly the same dynamic behavior. To overcome this problem, sextupole magnets will be used in the confining system to adjust the off-momentum proton motion, and this will introduce nonlinearity into the lattice. Furthermore, practical magnets also have systematic and random errors that will induce high-order multipole effects on the proton beam dynamics. Small misalignments are also common and will affect the beam dynamics. All of these unavoidable imperfections make the SSC a nonlinear machine that requires detailed numerical studies.

In these numerical studies, one starts with a well-designed linear lattice and then assigns systematic errors, random errors, and misalignment for the magnets, based on experience and measurement. Correction magnets may also be included. Ideally, protons are then tracked numerically for a limited number of turns to see if the motion is stable. At this stage, adjustment of the correction magnets is usually necessary (somewhat similar to the micro-tuning of a TV or a radio). After the accelerator is well tuned, one can then start short-term tracking (say, 400 turns) to study some

well-defined accelerator physics criteria to predict the behavior of the accelerator.

A typical short-term-tracking phase space plot is shown in Figure 1. The variation in the amplitude traced out by given protons is greater for those protons of larger initial amplitude. Here, as shown in Figure 1, the amplitude is defined as

$$\sqrt{x^2 + p_x^2} \ ,$$

where x is a Floquet space coordinate and p_x is its corresponding Floquet space momentum; that is, they are normalized such that a proton with linear motion would trace out a circle in (x,p_x) phase space. This phenomenon serves as a diagnostic of accelerator nonlinearity. If the amplitude variation is considered too big for a certain desired amplitude, the corresponding accelerator design should be modified.

Generally, to study the long-term stability, one would like to track hundreds of protons (with appropriate initial amplitude distributions) element by element for millions of turns (five minutes of SSC operation will be about a million turns). Using a current scalar computer would require months of central processing unit (CPU) time, since there are more than 10,000 magnet elements in the SSC machine. Fortunately, however, the protons in the beam may be considered to be independent from each other, so that a tracking code can be completely vectorized over the number of particles; a supercomputer is thus ideal for this purpose. One can track many particles (say, 64 protons) simultaneously, saving enormous CPU time over what a scalar machine would require. Indeed, an element-by-element post-Teapot[1] tracking program, "Ztrack,"[2] has recently been developed to take advantage of supercomputer vector processing. The code is vectorized for multiparticle tracking. It either reduces the number of particles tracked as particles are lost or substitutes new particles with new initial conditions for the lost particles to maintain a multiple of 64 particles to take the best advantage of the vector architecture.

Figure 2 shows a survival plot for a million-turn tracking from Ztrack. We studied a 2-TeV injection lattice, with a 4-cm-diameter magnet aperture. The rf cavity system was turned on to maintain bunching. Two hundred forty-seven particles were tracked, with initial horizontal (x-axis) displacement amplitudes distributed between 5 and 12 mm with respect to the closed orbit (the distribution was not equally spaced). The corresponding initial vertical (y-axis) displacement was between 2.07 and 4.98 mm, which was assigned such that the effective vertical displacement amplitude would also be between 5 and 12 mm. (There is a phase difference between horizontal and vertical betatron oscillations.) Thirty-seven particles survived for a million turns. While most of the particles with higher amplitudes were lost in the earlier turns, all of the particles with initial x displacement amplitude lower than 5.3 mm survived for a million turns. The dynamic aperture for a million turns is thus about 5.3 mm in radius for this 2-TeV injection lattice. Note that the dynamic aperture for 100,000 turns is about the same as for a million turns. Such a computation requires about 200 hours of CPU time on a Cray X-MP.

Only a few cases have been carried out to a million turns. Most of our long-term element-by-element tracking effort has

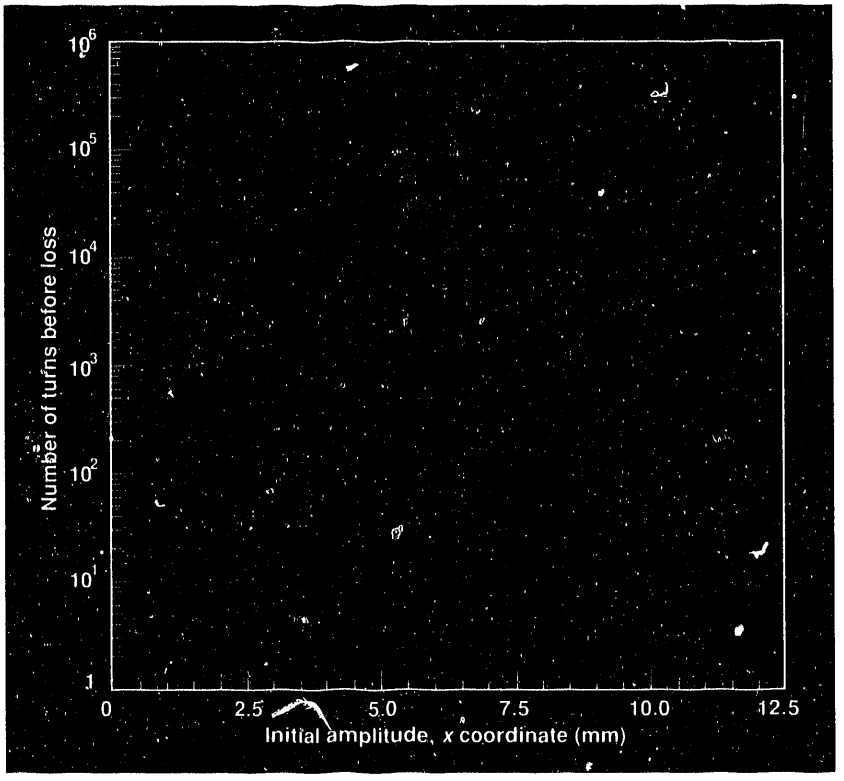


Figure 2. A million-turn survival plot for a 2-TeV, 4-cm-diameter magnet aperture injection lattice, showing how many turns around the collider ring were made by particles of various initial amplitudes. Out of 247 particles (ranging from 5 to 12 mm in the x displacement amplitude), 37 particles survived for a million turns. No protons with initial x displacement amplitude of less than 5.3 mm were lost. Thus the dynamic aperture for a million turns is about 5.3 mm in radius for this 2-TeV injection lattice. (This figure shows only the protons that were lost.)

been for 100,000 turns. Figure 3, a survival plot for up to 100,000 turns, compares the data from Figure 2 (for a 2-TeV lattice with a 4-cm-diameter magnet aperture) with the corresponding data for a lattice with a 5-cm-diameter magnet aperture. (The new data points are shown in red.) None of the particles with initial x displacement amplitude of less than 8.1 mm was lost. The only difference between the two lattices was in the multipole content due to the different size of the magnet aperture. With the increase in magnet aperture, the dynamic aperture for 100,000 turns enlarged from 5.3 to 8.1 mm in radius, because this increases the linearity of the machine.

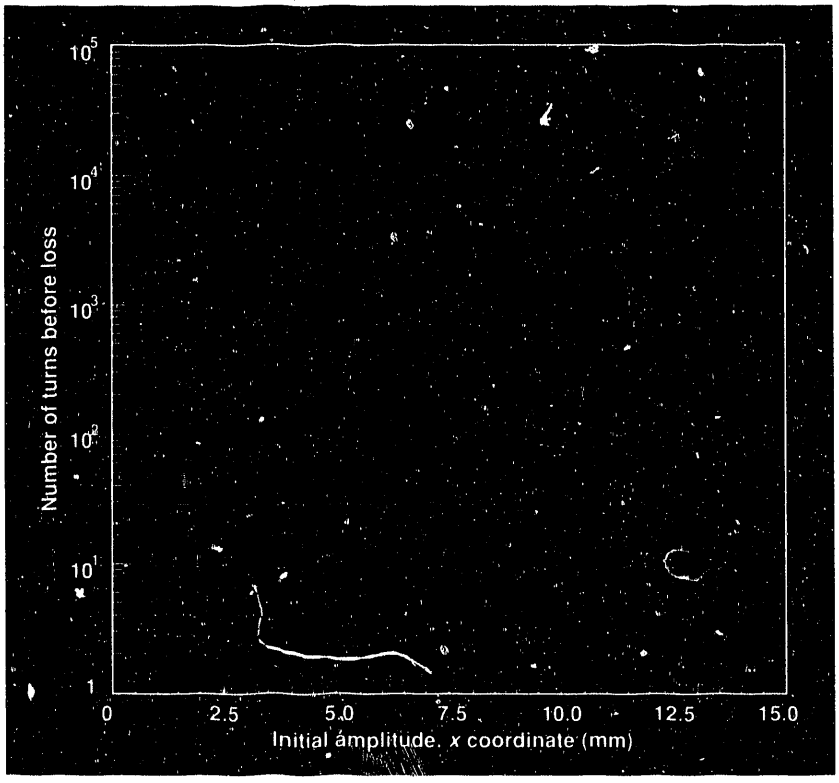


Figure 3. A 100,000-turn survival plot for a 2-TeV injection lattice, comparing the data for a 5-cm-diameter magnet aperture (shown in red) with the data for a 4-cm-diameter magnet aperture (shown in blue and also in Figure 2). With the 5-cm-diameter aperture, no particles with initial x amplitude of less than 8.1 mm were lost. By increasing the magnet aperture, the dynamic aperture for 100,000 turns enlarges in radius from 5.3 to 8.1 mm, which increases the machine's linearity. (Again, this plot shows only the protons that were lost before 100,000 turns were reached.)

Future Supercomputing Needs for the SSC

Although one could use the survival plots in Figures 2 or 3 to qualitatively project the dynamic aperture for longer turns, one would always question the reliability of such an extrapolation. Ultimately one wishes to track hundreds of particles for 10 million turns or more. (The lifetime of the SSC injection lattice will be about 10 million turns, while the lifetime of the collider lattice will be about 100 million turns.) Ztrack, which is a detailed element-by-element tracking code, would not be appropriate for such computations because we know that each round of 10-million-turn tracking would take about 2,000 hours of CPU time on a Cray X-MP.

Two more practical approaches for achieving the SSC lattice lifetime tracking are currently under development. One approach, SSCTRK, uses a simplified lattice with fewer elements to represent the SSC lattice qualitatively. The other approach uses a program called Zmap to extract the truncated power series map for the SSC lattice using a vectorized differential algebra library, the ZLIB,[3] and goes on to generate a factorization kick-map[4,5] for kick-map tracking (Zmaptrk). For the results from SSCTRK or Zmaptrk to be qualitatively comparable with the Ztrack results, it is expected that at least 1,600 super-elements will have to be used in the SSCTRK, or that an eleventh-order map will have to be extracted for Zmaptrk. Both SSCTRK and Zmaptrk are vectorized for multiparticle tracking. Preliminary testing shows that for 64 particles and 10-million-turn tracking, either approach will need about 100 hours of CPU time on the Cray-2 or slightly more time on the Cray X-MP.

In addition to the vector architecture that is helpful to the SSC lattice design, the large memory available in the Cray-2 is also critical in extracting high-order truncated power series maps for the SSC. The ZLIB requires a large number of integer pointers to optimize the truncated power series

multiplication and some other related routines. Zmap will be executed primarily on the Cray-2.

Parallel processing on multi-CPU supercomputers is currently being planned. Just as vector processing can speed up multiparticle tracking, so can parallel processors. Therefore, a multiple-processor supercomputer with an autotasking compiler would be of great value to the simulation effort. We look forward to having available a new supercomputer that has many (say, more than ten) CPUs and that is equipped with an autotasking compiler.

Summary

The success of the Superconducting Super Collider's operation will depend not only on the successful development of the superconducting magnets but also on an appropriate design for the SSC lattice. Obtaining such a lattice design will require extensive nonlinear work, involving huge computations. In particular, to determine the optimum aperture, one has to track hundreds of protons in the SSC lattice for at least 100,000 turns. Such a computer task would be difficult or even impossible without the vector processing available in supercomputers. ■

Acknowledgments

The author thanks L. Schachinger for extracting the SSC lattice file from the Teapot program that serves as the Ztrack input. He also thanks E. Forest for cooperating in developing one of the Z-family programs, the Zmaptrk, and D. Ritson for sharing with him the idea of simplified lattice representation for the SSC (SSCTRK). Valuable comments from G. Bourianoff, A. Chao, D. Edwards, J. Peterson, and, in particular, M. Gilchriese are highly appreciated.

References

1. L. Schachinger and R. Talman, "Teapot: A Thin-Element Accelerator Program for Optics and Tracking," *Particle Accelerators* **22** (1987).
2. L. Schachinger and Y. Yan, *Recent SSC Dynamic Aperture Measurements from Simulation*, Superconducting Super Collider Report No. SSC-N-664 (1989).
3. Y. Yan, *ZLIB: An IMSL-Style Numerical Library for Differential Algebra*, Superconducting Super Collider Report No. SSCL-300 (1990).
4. J. Irwin, *A Multi-Kick Factorization Algorithm for Nonlinear Maps*, Superconducting Super Collider Report No. SSC-228 (1989).
5. A.J. Dragt, "Method for Symplectic Tracking," presented at the Workshop on Nonlinear Problems in Future Particle Accelerators, Capri, Italy, April 1990.

Yiton Yan is an accelerator physicist in the SSC Laboratory. He received his Ph.D. degree from UCLA, where he worked with Professor John Dawson in plasma physics. Dr. Yan received a Director's Fellowship from Los Alamos National Laboratory for postdoctoral study in advanced accelerator concepts upon completion of his Ph.D. thesis in 1986. Two years later, he joined the SSC Central Design Group (which has since merged into the SSC Laboratory), working on long-term beam stability of the Superconducting Super Collider.

An Overview of Ongoing Studies in Climate Model Diagnosis and Intercomparison

W. Lawrence Gates, Gerald L. Potter, and Thomas J. Phillips, Lawrence Livermore National Laboratory; Robert D. Cess, State University of New York, Stony Brook

Predicting climate changes resulting from the "greenhouse effect" poses a problem: Which computational model is the most accurate?

One of today's most critical global problems is the climatic effect of increasing "greenhouse gases" in the atmosphere. These gases act as a blanket to trap the Earth's longwave or infrared radiation, thus raising the Earth's temperature. Carbon dioxide (CO_2) is the primary culprit. The warming from increased CO_2 concentration creates more water vapor in the atmosphere, which in turn adds to the greenhouse effect and causes still more warming and evaporation—a cycle of "positive feedback."

This problem has generated considerable interest in using large-scale computational modeling to predict the global climate changes caused by both the observed and the projected human-caused increases in the concentration of atmospheric CO_2. The most useful tools for this purpose are three-dimensional atmospheric models known as general circulation models (GCMs). Nearly 20 GCMs are currently being used around the world for climate research.

Although the basic design of the GCMs is similar, the various models produce significant differences in their projections of the climate change to be expected from increasing CO_2. The difficulty lies in knowing which GCM is the most accurate. In research supported by the U.S. Department of Energy, we have been using GCMs at the National Energy Research Supercomputer Center (NERSC) to develop and test specific model simulations that are designed to clarify some of the GCM features that may be responsible for these differences in predictions. This work is being performed by the Program for Climate Model Diagnosis and Intercomparison at Lawrence Livermore National Laboratory (LLNL).

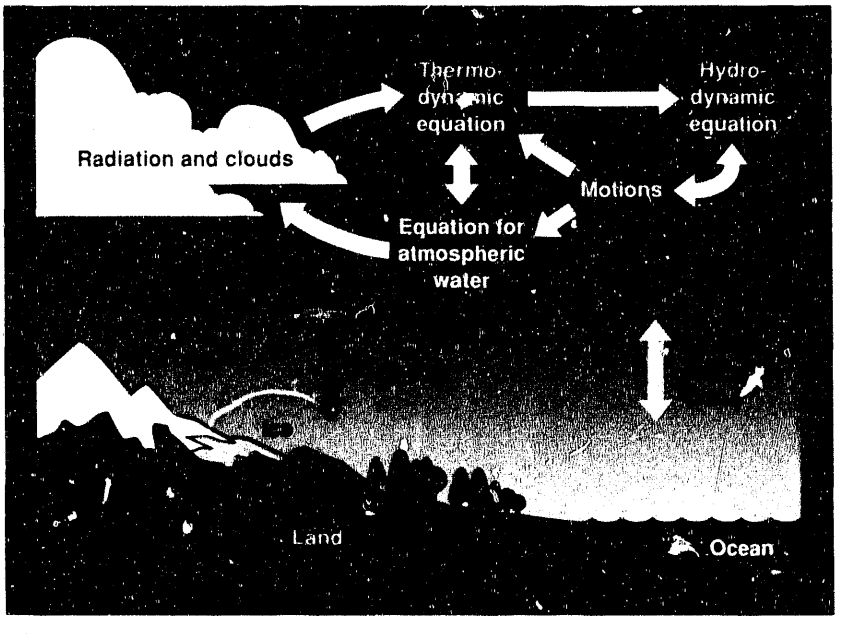


Figure 1. Diagram of the principal interactions between the dynamic aspects of climate models and physical factors such as clouds, radiation, heat fluxes, and surface types.

Overview of Climate Models

Generally, GCMs treat the equations of motion of the atmosphere through interaction with its thermodynamics. The principal interactions are shown in Figure 1. Vertically, the GCM's computational domain extends from the surface of the Earth to as high as 35 km. Horizontally, the computational domain covers the entire globe and is divided into grid cells that represent, in each dimension, hundreds of kilometers on the Earth. The various GCMs have from about 3,000 to more than 50,000 of these grid cells.

Limitations in computer memory and speed have historically restricted the resolution of the models, and only recently have GCMs begun to experiment with horizontal grid sizes smaller than 100 km. Much of the uncertainty and disagreement in the models' projections of climate change derives from the treatment or parameterization of processes that occur on scales smaller than the grid size.

A Climate Experiment

We have recently shown that cloud-radiative feedback is one of the most important reasons for the differences in model response.[1] This conclusion is the result of our effort to compare the various GCMs in use today and is based on preliminary tests and recommendations developed both at LLNL and at the State University of New York, Stony Brook, as part of an international cooperative project. The initial prototype simulations were performed at NERSC and resulted in a strategy for the intercomparison of climate models.

Clouds have a significant influence on the radiation budget at the top of the Earth's atmosphere.[2] For example, clouds act to cool the Earth by reflecting solar radiation back to space, and at the same time they warm the Earth by acting as a blanket to hold in the Earth's longwave radiation. The combination of these two effects is referred to as cloud radiative forcing. To illustrate, of the approximate 340 W/m^2 of solar radiation that on the average reaches the Earth from the sun, clouds cool the Earth by reflection by about 50 W/m^2, and the longwave blanketing (or greenhouse) effect warms the Earth by about 25 W/m^2. Thus the net cloud radiative forcing in this case would be a cooling of about 25 W/m^2, which is much larger than the 4 W/m^2 warming expected to occur as a result of CO_2 doubling. Changes in cloud radiative forcing by as little as 15% can therefore mask increased CO_2 effects.

Unfortunately, however, clouds present one of the most difficult phenomena to model. The problem is primarily one of resolution, since clouds normally occur on very small scales. Even with today's state-of-the-art supercomputers, it is impossible to include individual clouds and their microphysics in a global model.

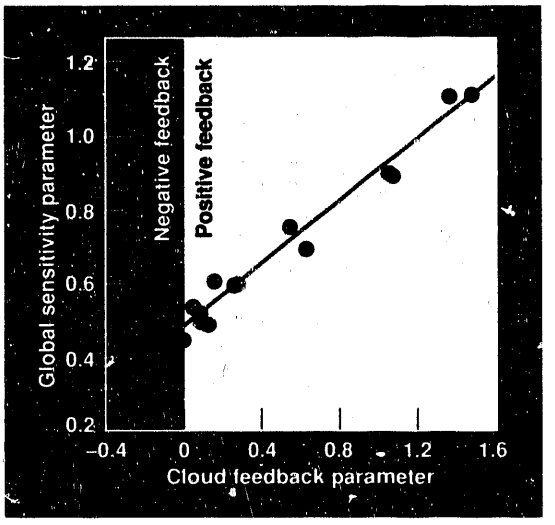


Figure 2. In a common simulation, the various global climate models produced widely different estimates of the radiative impact of clouds. As indicated by this graph, three of the models predicted negative feedback from clouds, one predicted no effect from cloud feedback, and the remaining models predicted varying degrees of positive feedback.

One of the first steps in unraveling the differences in various models' predictions of the clouds' effects has been to perform a common simulation designed to emphasize the differences in the models' treatment of atmospheric processes. In a collaborative program, we suggested that the various GCM modeling groups use the same surrogate for climate change (as discussed, for example, in References 1 and 3). Each modeling group subtracted 2 kelvin (K) at every ocean grid point from a standard sea-surface temperature distribution that we had furnished, and ran their model to statistical equilibrium with a fixed July solar declination. Another run was then made with 2 K added to the ocean temperature. These runs amounted to a −2 K and a +2 K simulated climate change. The advantage of this strategy is that other feedbacks are minimized, such as those with snow and ice.

The results, summarized in Reference 1, demonstrate that cloud radiative properties are responsible for the bulk of the differences in model response. In examining only the clear-sky fluxes of both longwave and shortwave radiation, the models showed close agreement. However, for the total response (including cloud radiative processes), the models varied in sensitivity by a factor of 3. This important result is shown in Figure 2, taken from Reference 1.

Comparison Against Observed Data

Now that we have determined that cloud-radiative feedback causes many of the differences in model response, the next question we face is to find out which, if any, of the various models is correct. To this end, we hope to determine cloud radiative forcing under the most realistic conditions possible and then to calibrate the computational results against satellite observations of the Earth's radiation budget.

A new set of global data on observed cloud radiative properties can be used for this purpose. The Earth Radiation Budget Experiment (ERBE), sponsored by the National Aeronautics and Space Administration, has produced an observational data set. These data have only recently been processed, and just a few months of observations have been released to date. The value of these new data is that they contain the same radiative balance information as simulated by the models. The ERBE satellite scanners measure Earth's radiation under clear, partly cloudy, mostly cloudy, and overcast conditions, with a resolution of about 35 km at nadir.

Ideally, each GCM's results would be compared with the observed data—a project that would require computational resources beyond current capabilities. As a start, we are cooperating with the European

Studies with a Coupled Atmosphere-Upper Ocean Model

As part of our effort to more fully understand CO_2-induced climate change, we have used an atmospheric model coupled with the upper layers of the ocean. This coupled atmosphere-ocean model has now been used in a 50-year simulation of how the climate would respond to a doubling of atmospheric CO_2. The model was developed jointly by Lawrence Livermore National Laboratory and Oregon State University.*

The ocean model simulates currents and sea ice, along with an internally determined sea-surface

Centre for Medium Range Weather Forecasts (ECMWF) in Reading, England, in the use of their model for climate simulation. This model is readily adaptable to extremely high resolution and was available for cooperative use.

The ECMWF Model

The ECMWF model is vectorized and multitasked and now runs on one of NERSC's Cray-2 machines at a speed of about two computer resource units (CRUs) per simulated day at a global resolution of about 100 km. In the model, the fields of atmospheric variables such as wind, pressure, temperature, and humidity are represented as coefficients of a truncated series of spherical harmonics, the eigenfunctions of Laplace's equation in spherical coordinates. The time evolution of these spectral coefficients is determined by numerical integration of a coupled set of partial differential equations that are evaluated at 19 vertical levels in the model atmosphere.

At each model timestep, the linear contributions to the equations of atmospheric motion (such as the diffusion of heat and momentum) are computed exclusively in spectral space, while the nonlinear products of atmospheric variables are first computed on a latitude/longitude grid and then transformed to spectral space by the successive application of a fast Fourier transform (FFT)

temperature. Although the model displays systematic errors, this simulation indicates that the polar oceans would become ice-free during the summer with doubled CO_2, while notable warming would occur over continental interiors.

Here we show the model's simulated changes of surface air temperatures produced as a result of doubled CO_2 for (a) January and (b) July, averaged over a 10-year period. The color bars indicate the amount of warming, in kelvins.

—W.L.G. and G.L.P.

* W.L. Gates and G.L. Potter, "Simulation of the Climatic Effects of Increased CO_2 with a Coupled Ocean-Atmosphere Model," to be submitted to *Climate Dynamics*.

and Gaussian numerical quadrature. The grid spacing, which is slightly more than one degree latitude by one degree longitude, is fine enough to allow this spectral transformation to be done without loss of useful information through "aliasing." The model's numerical time integration scheme combines the "explicit" evaluation of modes of atmospheric motion that are slowly varying and the "implicit" computation of more rapidly varying modes. This "semi-implicit" scheme has the advantage of permitting longer timesteps than would be the case if all terms were evaluated explicitly.

Further Climate Simulations

We hope to complete several sets of climate simulations with the ECMWF model in perpetual seasonal modes, a task that we estimate will take about 2,000 CRUs and that is expected to be completed by the end of fiscal year 1990. This will be the first time that a GCM has been run for long simulations at such high resolution, and the results should generate considerable interest in the climate modeling community.

A next step will involve using the ECMWF model to produce multiyear simulations with four different resolutions: about 500, 300, 200, and 100 km. (The 100-km resolution is the same resolution used for the cloud radiative studies in perpetual seasonal mode.) This set of runs will require an additional 2,000 or more CRUs on one of NERSC's Cray-2 computers during fiscal year 1990.

In the future, we expect to use between 5,000 and 10,000 CRUs annually as the work of the Program for Climate Model Diagnosis and Intercomparison reaches full stride with the involvement of the international modeling community in a major intercomparison effort. ■

References

1. R.D. Cess et al., "Interpretation of Cloud-Climate Feedback as Produced by 14 Atmospheric General Circulation Models," *Science* **245**, 513 (1989).
2. T.P. Charlock and V. Ramanathan, "The Albedo Field and Cloud Radiative Forcing Produced by a General Circulation Model with Internally Generated Cloud Optics," *J. Atmos. Sci.* **42**, 1408 (1985).
3. R.D. Cess and G.L. Potter, "A Methodology for Understanding and Intercomparing Atmospheric Climate Feedback Processes in General Circulation Models," *J. Geophys. Res.* **93**, 8305 (1988).
4. R.D. Cess et al., "Intercomparison and Interpretation of Climate Feedback Processes in Nineteen Atmospheric General Circulation Models," *J. Geophys. Res.* (1990) (in press).

W. Lawrence Gates is Program Leader of the Program for Climate Model Diagnosis and Intercomparison at LLNL. He was previously Chairman of the Department of Atmospheric Sciences at Oregon State University.

Gerald L. Potter is Deputy Program Leader of the Program for Climate Model Diagnosis and Intercomparison at LLNL.

Thomas J. Phillips is a staff scientist in the Program for Climate Model Diagnosis and Intercomparison at LLNL.

Robert D. Cess is Leading Professor of Atmospheric Sciences at the State University of New York, Stony Brook.

MHD Simulation of the Fueling of a Tokamak Fusion Reactor Through the Injection of Compact Toroids

A.A. Mirin and D.E. Shumaker, Lawrence Livermore National Laboratory

A three-dimensional magnetohydrodynamics code is used to model a new concept for fueling a magnetic fusion reactor.

Magnetic confinement fusion is one of several possibilities for meeting the world's long-term power needs. The concept involves injecting fuel into a container surrounded by magnetic coils. Ionization of the fuel will occur, and the resulting charged particles—or plasma—will follow the direction of the magnetic field, which is oriented to keep the particles from escaping from the container. Heating the fuel mixture to an extremely high temperature will result in fusion of the particles, releasing great amounts of energy.[1]

The long-range goal is to develop a magnetic fusion reactor that produces sufficient energy for the economical generation of electricity. To be successful, the fusion power produced must substantially exceed the power put into the system. For this to occur, the plasma must reach a stable equilibrium that is both hot enough and sufficiently well confined. If we denote the plasma density by n, the plasma temperature by T, and the plasma lifetime by τ, the product $n\tau$ must exceed a critical minimum, and T must also be high enough. At present, either one or the other of these conditions is achievable, but not both simultaneously.

The high cost (hundreds of millions of dollars) and long lead times (up to 10 years) for constructing contemporary magnetic fusion devices make a comprehensive theoretical effort absolutely essential. The leading approach in magnetic fusion research is to confine the fuel in a doughnut-shaped device called a tokamak. Because the equations governing tokamak dynamics are much too complicated to be solved by analytic means, a strong computational effort is required. Moreover, the range of timescales and spatial scales is so broad that it will probably never be technologically possible to completely model a fusion reactor with a single "supercode." For example, the typical plasma lifetime of a contemporary fusion device is on the order of seconds, whereas important electron microinstabilities operate in the picosecond-to-nanosecond range. A tokamak reactor is expected to be several meters in circumference, whereas one of the most important spatial scales, the Debye length, is in the millimeter range. To do a complete simulation taking all phenomena into account could require billions of timesteps and perhaps trillions of meshpoints. It is thus apparent that magnetic fusion modeling taxes the capability of today's supercomputers and will continue to do so for a long time to come.

One of the most commonly used representations of a plasma is that of a fluid, and the physical model that governs the evolution of the plasma and the associated electric and magnetic fields is known as magnetohydrodynamics (MHD). MHD codes typically operate on the microsecond timescale and thus cannot be used to simulate the duration of an experiment. Such an endeavor would require further assumptions and would result in a set of transport equations. On the other hand, the fluid theory is itself an assumption, and relaxing that to model kinetic phenomena, for example, would require either a Vlasov or a particle-in-cell approach. (See the article by W. W. Lee, page 25.)

The problem addressed here, a rather novel approach to fueling a tokamak reactor, is well suited to an MHD representation. It is a fully three-dimensional problem that would not have been tractable before the advent of the Cray supercomputers.

How to adequately and efficiently fuel a tokamak reactor is not a trivial question. Present tokamak devices are fueled by the injection of pellets. However, there is fear that in a tokamak reactor, due to its larger size, this technique will not succeed because of limits on how far pellets can penetrate the plasma before ablating and dumping their fuel. Lack of deep-penetration fueling will not only cause particles to be confined for less time (thereby reducing their probability of undergoing fusion), but it is also likely to result in a less stable density profile. Hence, compact torus injection has been suggested as an alternative fueling method.[2]

A compact torus (CT) is merely a magnetically confined ball of plasma. To date, CTs have served mainly as the centerpiece of alternative-concept magnetic fusion devices. CTs can be accelerated to very high velocities. The idea is to continuously inject CTs from a plasma gun into the tokamak. The CT will decelerate and stop at the optimal location in the tokamak, and its magnetic field lines will then "reconnect" or merge with those of the tokamak, allowing the entrapped fuel and magnetic flux to be deposited and mixed with the tokamak plasma.

An added benefit of CT injection is the possibility it provides of continuous (rather than pulsed) operation. For a tokamak to confine a stable plasma, there must be a current flow in the toroidal (that is, the long-way-around) direction. This current has traditionally been provided by a large pulse transformer. However, the magnetic flux convected with the injected CTs can serve to provide a continuous current. Such a mode of operation is much more desirable from an engineering standpoint.

To evaluate the effectiveness of this technique, one must ascertain whether the CT will stop in the desired location, how long it will stay there, and where it will dump its fuel. The last issue is determined by how long it takes for the magnetic field of the CT to merge with that of the tokamak.

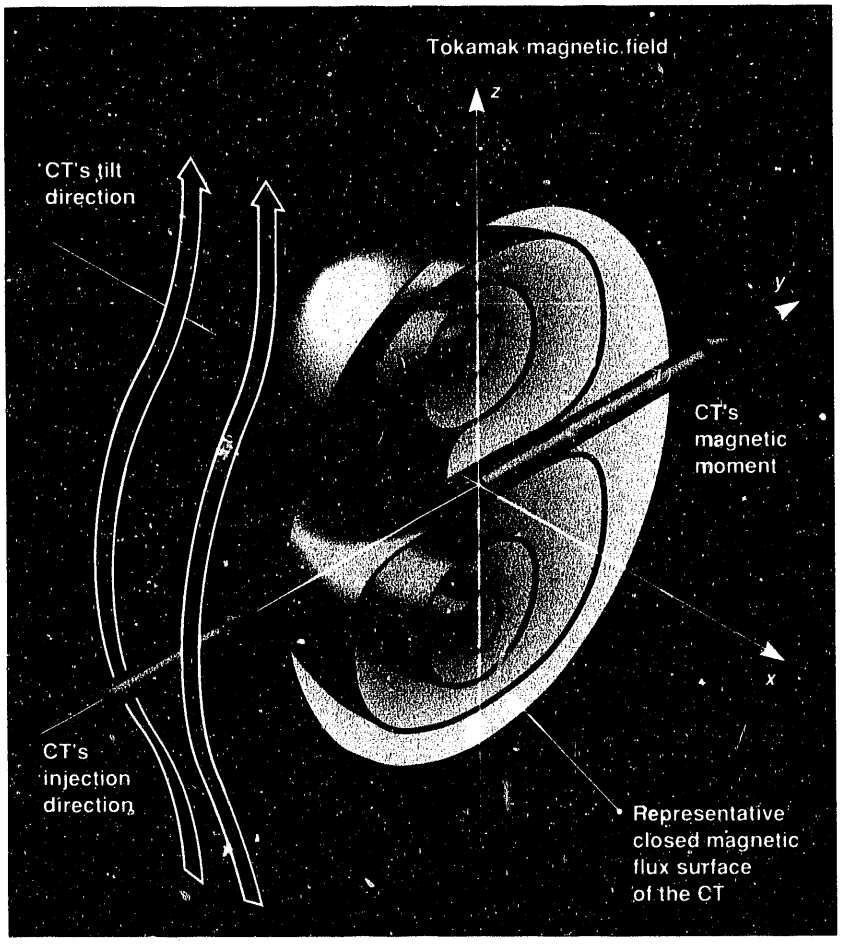


Figure 1. Depiction of a compact torus moving across the tokamak magnetic field. The CT is moving in the *y* direction; the tokamak magnetic field is in the *z* direction.

Computational Model

The time evolution of the system of equations is modeled with the three-dimensional code TEMCO,[3] which solves the primitive, single-fluid equations of compressible magnetohydrodynamics. TEMCO was developed at the National Energy Research Supercomputer Center during the 1980s and has been used to model various types of magnetic confinement devices.[4,5] TEMCO has also served as an example of multitasking.[3]

The physics model of TEMCO includes resistivity, viscosity, and thermal conductivity (all isotropic); Ohm's law as applied here contains Hall terms. The code uses cylindrical coordinates (r, ϕ, z) in toroidal geometry, where ϕ is the toroidal angle and r, z are the coordinates for the poloidal (that is, the short-way-around) planes. Coupled evolutionary partial differential equations for the density, temperature, velocity, and magnetic fields are integrated in time. The pressure, electric field, and current density are expressed in terms of

the primary dependent variables through the equation of state, Ohm's law, and Ampere's law, respectively.

Finite difference approximations on a variably spaced mesh are used in the r and z directions, and a Fourier expansion is used in ϕ. Fourier convolutions are performed using a pseudospectral technique.[6] This is a method in which variables are freely transformed between configuration space and wave number space; multiplication is carried out in the former and differentiation in the latter. The discretized equations are time-integrated using an explicit (leapfrog) algorithm either with operator splitting or with implicitization of the diffusive terms.[7] To allow larger timesteps, a semi-implicit algorithm may be used.[8]

The explicit portion of the time advance is accomplished as follows. Both the current density and pressure are computed everywhere. The z fluxes and their derivatives are then computed at all meshpoints. The dependent variables are advanced one z line at a time for all r, ϕ; and all other coefficients are computed as needed. Vectorization is generally performed in the r direction.

The semi-implicit time advance is designed so that the various ϕ harmonics decouple (taking the viscosity and resistivity to be functions of, at most, r and z), and the z components decouple from the r and ϕ components. Fourier transforming is used in the z direction, and the resulting tridiagonal systems (either scalar or 2-by-2-block) are solved using standard techniques. The decomposed system matrices are recomputed only when necessary.

Most of the main time-integration loop is multitasked.[3] In undertaking such a strategy, one must decide where to place synchronization points—that is, the locations at which the code will wait for all outstanding tasks to be completed. It turns out that five such synchronization points are needed, resulting in a division of the main time-integration loop into six multitasked sections. The work within each multitasked section is then partitioned into N tasks of approximately equal duration, where N is the number of central processing units. Details of this analysis are presented in Reference 3.

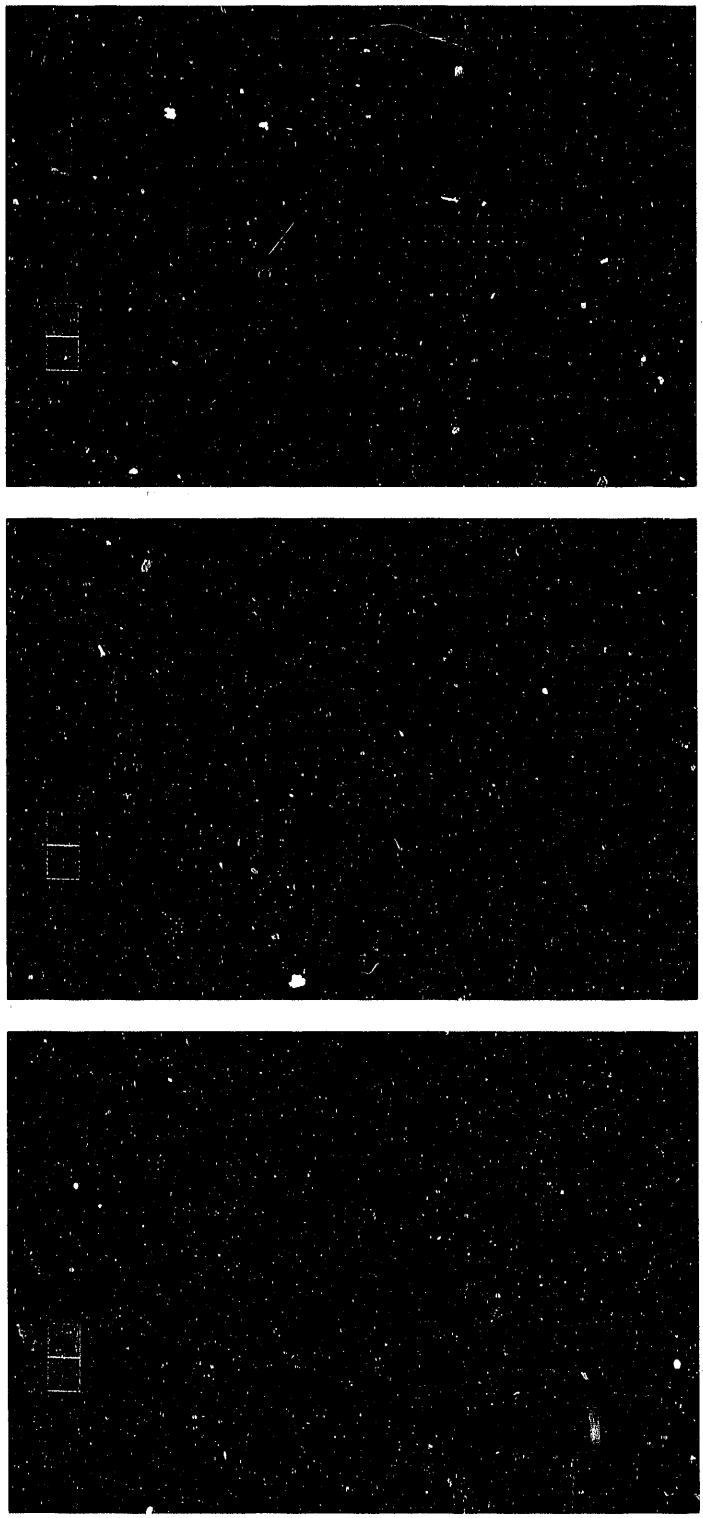


Figure 2. Vector plots of the magnetic field in the $x = 0$ plane, taken at (top) $t = 0$, (middle) $t = 2.5$, and (bottom) $t = 5.0$. (One time unit t equals one internal CT Alfvén time.) The red arrows correspond to a strong magnetic field in the $+x$ direction, the blue arrows correspond to a strong magnetic field in the $-x$ direction, and the other colors correspond to intermediate values of magnetic field. The CT radius is approximately 10 units. By $t = 5.0$, a tilting of 45 degrees is discernible as the CT begins to align its magnetic moment with the tokamak's magnetic field.

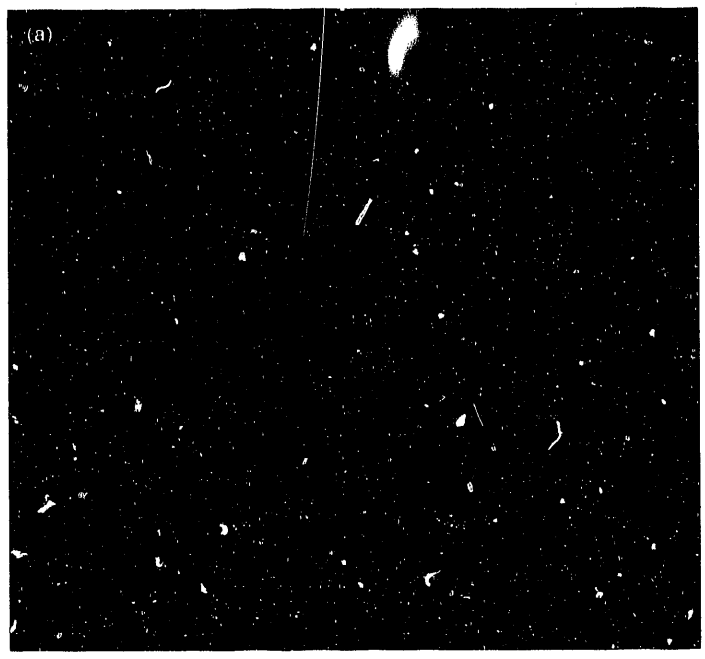


Figure 3. Intersection of magnetic field lines with different z-planes, with (a) showing $t = 0$, (b) showing $t = 2.5$, and (c) showing $t = 5$. The yellow dots represent confined field lines, and the blue dots represent unconfined field lines. At $t = 0$, the toroidal structure is easily discernible, extending vertically just beyond $z = +5$ and $z = -5$. By $t = 5$, the confined magnetic flux has all but evaporated.

Application

To simplify the calculation, the CT is taken to be initially at rest in a uniform magnetic field (which represents the tokamak field) that has been modified to exclude the CT. The neglect of spatial dependence of the surrounding field is justified on the grounds that the CT is much smaller than the tokamak. The assumption that the CT is at rest is valid provided the deceleration time of the CT is faster than other timescales of interest (this point is currently under study). The magnetic field inside the CT itself is an analytic representation typical of such configurations.[9] The initial density and temperature inside the CT and at the computational boundary are specified, with an analytic smoothing formula for locations in between. The initial plasma velocity is taken to be zero. The boundary values (which in actuality represent the tokamak parameters) are fixed in time. The computational domain extends 25 units in the r direction and 50 units in the z direction. The CT has a radius of approximately 10 units. The finite difference mesh is uniformly fine in the CT and becomes gradually coarser (in geometric fashion) as the boundary is approached. The explicit difference method is used for the time integration (the semi-implicit algorithm turns out to be unsuitable for this problem).

Figure 1 shows the initial orientation of the CT. (The coordinate system x, y, z for the output diagnostics is different from that of the computation itself.) Note that the CT's magnetic moment is aligned orthogonally to the direction of the tokamak magnetic field.

Our representative case has a Lundquist number (ratio of resistive to Alfvén times) of 4000 and is carried out on a 41 (r) by 16 (ϕ) by 81 (z) mesh; to assure de-aliasing, only nine ϕ modes are retained.[6] The resulting magnetic field is postprocessed using the field-line-tracing code TUBE.[10]

Figure 2 shows vector plots of the magnetic field in the $x = 0$ plane at times $t = 0$, $t = 2.5$, and $t = 5$. (One time unit is roughly equivalent to an internal CT Alfvén time). The arrow color indicates the direction and magnitude of the magnetic field. Red corresponds to a strong magnetic field in the $+x$ direction (out of the picture), blue corresponds to a strong magnetic field in the $-x$ direction (into the picture), and the other colors correspond to intermediate values of the magnetic field. By $t = 5$, a tilting of about 45 degrees is discernible, as the CT is attempting to align its magnetic moment with the tokamak's magnetic field.

Figure 3 shows intersections of magnetic field lines with different z-planes. The yellow dots represent confined field lines, and the blue dots represent unconfined field lines. At $t = 0$, it is relatively easy to discern

the toroidal structure, which extends vertically just beyond $z = +5$ and $z = -5$. By $t = 5$, though, the confined magnetic flux has all but evaporated. The careful observer will detect a slight tilt in that time—much less than indicated in Figure 2. This shows that the CT is not tilting as a rigid body.

Figure 4 is a plot of the confined magnetic flux versus time. It can be seen that the decay is approximately exponential. By measuring the slope of the straight-line fit, one obtains a time constant of 3.72. When this case is rerun with twice the mesh resolution (in each direction), the decay time changes by a mere 2%.

Toward the Future

Although the above calculation gives numerically accurate results, extrapolation to larger Lundquist numbers is necessary for the proper assessment of the rate of decay of the confined magnetic flux. Unfortunately, as the Lundquist number increases, the spatial scale length associated with the resistivity decreases, and greater mesh resolution is required. This makes the calculations all the more computationally demanding.

A reasonably accurate theoretical assessment of the CT fueling process can be made once we have ascertained the flux decay rate (for this scenario in which the CT is initially at rest) and the CT deceleration rate (provided the latter is fast enough). This should be of great aid in establishing the optimal parameters for CT injection experiments.

Evaluation of advanced concepts, such as this one on the use of CTs, is invaluable to the magnetic confinement effort. Because of the high cost and long lead time for experimental projects, state-of-the-art supercomputer utilization is—and will continue to be—indispensable to the magnetic fusion energy program. ■

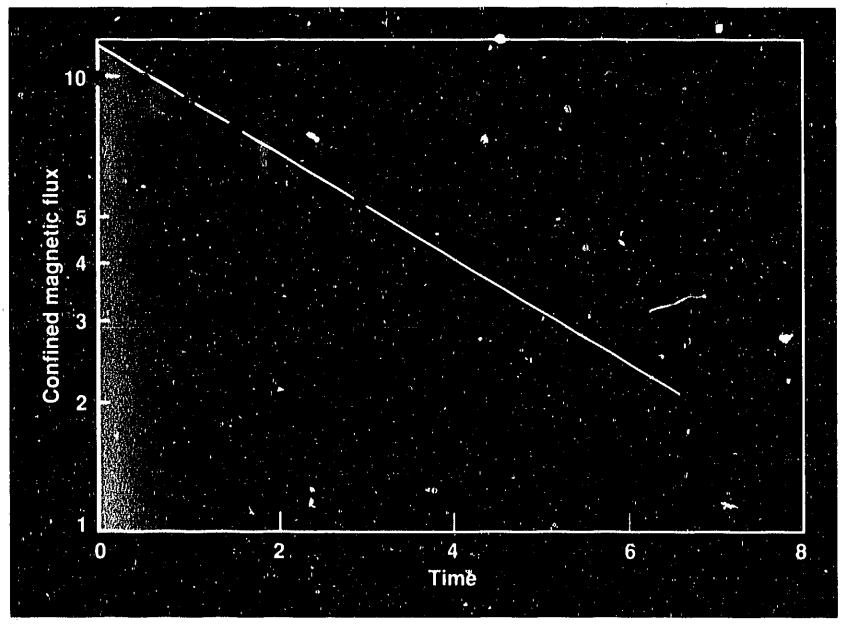


Figure 4. A plot of the confined magnetic flux versus time, showing that the decay is approximately exponential. Measuring the slope of the straight-line fit gives a time constant of 3.72.

Acknowledgment

We acknowledge J.H. Hammer of LLNL for many helpful discussions.

References

1. K.I. Thomassen, "Progress and Directions in Magnetic Fusion Energy," *Ann. Rev. Energy* **9**, 281 (1984).
2. L.J. Perkins, S.K. Ho, and J.H. Hammer, "Deep Penetration Fueling of Reactor-Grade Tokamak Plasmas with Accelerated Compact Toroids," *Nucl. Fusion* **28**, 1365 (1988).
3. A.A. Mirin, "Predicting Multiprocessing Efficiency on the Cray Multiprocessors in a (CTSS) Time-Sharing Environment: Application to a 3D Magnetohydrodynamics Code," *Computers in Physics* **2**, 62 (1988).
4. A.G. Sgro, A.A. Mirin, and G. Marklin, "The Evolution of a Low Beta Decaying Spheromak," *Phys. Fluids* **30**, 3219 (1987).
5. A.Y. Aydemir, D.C. Barnes, E.J. Caramana, A.A. Mirin, R.A. Nebel, D.D. Schnack, and A.G. Sgro, "Compressibility as a Feature of Field Reversal Maintenance in the Reversed Field Pinch," *Phys. Fluids* **28**, 898 (1985).
6. S.A. Orszag, "Numerical Simulation of Incompressible Flows Within Simple Boundaries. 1. Galerkin (Spectral) Representations," *Stud. Appl. Math.*, Vol. L, 293 (1971).
7. R.D. Richtmyer and K.W. Morton, *Difference Methods for Initial Value Problems*, 2nd ed. (Wiley, Interscience, New York, 1967).
8. A.A. Mirin, "The Semi-Implicit Method," *NMFECC Buffer*, August 1989.
9. M.N. Rosenbluth and M.N. Bussac, "MHD Stability of Spheromak," *Nucl. Fusion* **19**, 489 (1979).
10. A.A. Mirin et al., "TUBE88: A Code Which Computes Magnetic Field Lines," *Comput. Phys. Commun.* **54**, 183 (1989).

A.A. Mirin is the leader of the Computational Physics Group at the National Energy Research Supercomputer Center at Lawrence Livermore National Laboratory. His main area of interest is scientific computing. Dr. Mirin received his Ph.D. in mathematics from the University of California, Berkeley, in 1974.

D.E. Shumaker is a member of the Computational Physics Group at the National Energy Research Supercomputer Center at Lawrence Livermore National Laboratory. His main area of interest is computational plasma physics. Dr. Shumaker received his Ph.D. in applied science from the University of California, Davis, in 1976.

Gyrokinetic Particle Simulation of Tokamak Plasmas

W.W. Lee, Plasma Physics Laboratory, Princeton University

The gyrokinetic approach to particle simulation holds promise for a better understanding of magnetically confined plasmas.

For the past quarter of a century, particle simulation has gradually developed into a useful tool for understanding the highly complicated interactions of charged particles in laboratory and space plasmas.[1,2] It has made contributions to magnetic fusion research in the areas of heating and transport studies. The basic idea is to use the computer to calculate the position and velocity of the individual plasma particles (electrons and ions) whose trajectories are given by Newton's law. The particles interact with externally applied electric and magnetic fields, as well as with those fields generated by the particles themselves. The latter, commonly known as self-consistent (or collective) fields, are given by Maxwell's equations. Since the purpose is to reproduce numerically the behavior of a real plasma, we can consider particle simulation as an experiment performed via computer. Mathematically, we are actually solving a set of Vlasov-Maxwell equations using the method of characteristics and the Klimontovich representation for particle distribution in the phase (configuration and velocity) space.[1-3]

There is one hitch, however. The typical density of a laboratory plasma ranges from 10^{12} to 10^{14} particles/cm^3, and it is not feasible to use that many particles in the simulation, even with present-day supercomputers. Fortunately, this is not necessary, because of the existence of Debye shielding—a unique property of the plasma. Debye shielding occurs through the Coulomb interaction between the plasma particles. It can be shown that a "test" particle, when introduced into a plasma in equilibrium, acquires a shielding cloud made up of an excessive charge with the opposite sign. This property is manifested through the resulting Coulomb potential of the test particle, which is now modified from $1/r$ to $(1/r)\exp(-r/\lambda_D)$, where r is the distance from the particle and λ_D is the size of the shielding cloud. Thus, for $r > \lambda_D$, the effect of the test particle is neutralized because of the shielding.

Since, collectively, the Coulomb potential for each individual particle in the plasma still retains its original $1/r$ dependence, the presence of Debye clouds does not alter the physics governing the long-range interactions for which the wavelengths are longer than λ_D. On the other hand, the existence of Debye clouds enables us to make each particle in the simulation the size of the Debye cloud and to use a much reduced number of particles for investigating these collective phenomena. In this process, the charge for the original atomic-point-size particle is now smeared to a sphere with a radius λ_D. This is the well-known finite-size particle-simulation model.[1,2] Thus, particle simulation does not attempt to describe the phase space dynamics in full gory detail; instead, it is simply a sampling technique for capturing the long-range collective interactions.

However, there is a slight problem: although the Coulomb potential remains unchanged outside the sphere with radius λ_D, inside the sphere it is modified and becomes proportional to r. Modifying the inside potential renders the simulation plasma essentially collisionless. To simulate accurately the "collisional effects" arising from the particle interactions inside the shielding cloud, one needs an enormous number of point-size particles (with $1/r$ for the potential). Fortunately, these interactions are fairly well understood theoretically. If and when the collisional effects become important, one can account for them by simply using (non-self-consistent) Monte Carlo calculations in the simulation. (See, for example, Reference 4.)

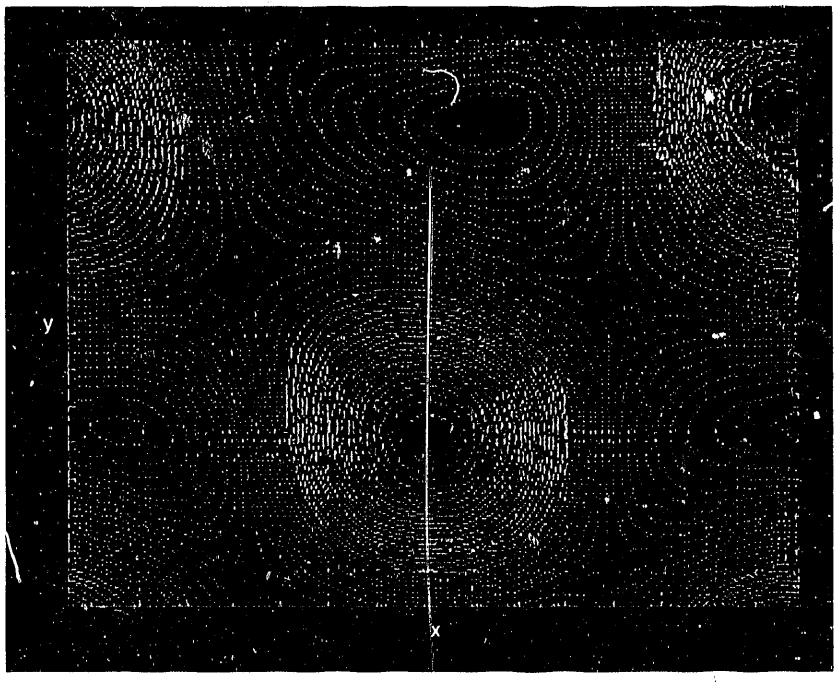


Figure 1. Electron trapping and equipotential contours of the electric field for a simple simulation of drift waves. The electron guiding centers are shown in red, and the yellow lines are the equipotential contours for the electric field (or the waves). In this saturated stage, trapping of the electrons is clearly visible.

Applying particle simulation to investigate low-frequency phenomena in tokamaks has been plagued with numerical difficulties.[3] The dominant computational workhorse in magnetic fusion research so far has been magnetohydrodynamics (MHD) simulation, which represents the plasma as a fluid. (See the article by A.A. Mirin and D.E. Shumaker, page 19.) Since MHD simulation is based on a reduced description of the plasma, the numerical schemes to solve these fluid-type equations are well established and relatively easy. However, the reduced description does not give all the physics we need to understand plasma behavior in tokamaks. Most notably, it cannot treat the physics of the anomalous transport that arises from the microinstabilities driven by the spatial inhomogeneity of the confined plasma.

"Anomalous transport" refers to the complex and not-yet-understood phenomena associated with the particle and heat loss observed in tokamak experiments. Transport, through the radially outward movement of the particles, dilutes the density and energy at the center of the confined plasma. It cannot be explained by the standard neoclassical transport theories and is believed to be caused by highly nonlinear plasma interactions (turbulence). The U.S. fusion community recently inaugurated a Transport Initiative to coordinate and focus research on this important issue.

In an attempt to understand anomalous transport, various versions of the two-fluid equations (instead of the one-fluid MHD model) have been used. Nevertheless, with the advent of the tokamak experiments in recent years, it becomes increasingly hard to justify the use of the fluid model to describe the collisionless physics associated with high-temperature plasmas.

The search for a better particle-simulation technique for studying tokamak physics was initiated in the early 1980s at the Princeton Plasma Physics Laboratory. Our gyrokinetic approach was analytical in nature, in contrast to the numerical approach taken by our contemporaries, who were alert to the implicit algorithms[2] and who had somewhat different applications in mind. Nonetheless, our purpose was the same: to increase both the timestep ($\omega_{pe}\Delta t >> 1$, where ω_{pe} is the plasma frequency) and the grid spacing or cloud size ($\Delta x / \lambda_D >> 1$) in the simulation, and also to decrease the numerical noise for using a finite number of particles.

The derivation of the governing equations for gyrokinetic simulation was based on the well-known gyrokinetic ordering, which assumes that the ratios of the following quantities are all small: (1) the frequencies of interest versus the gyrofrequencies, (2) the wavelengths parallel to the ambient magnetic field versus the wavelengths that are perpendicular to the field, (3) the fluctuation potential energy versus the particle kinetic energy, and (4) the gyroradii versus the scale lengths of the background plasma and field inhomogeneities. Applying the gyrokinetic ordering to the original Vlasov-Maxwell equations and taking the gyrophase average, we obtained a set of reduced equations, known as the nonlinear gyrokinetic equations.[5] The numerical properties of these equations satisfy all the requirements mentioned in the previous paragraph.[3] The reason is that the high-frequency oscillations, such as plasma waves (ω_{pe}) and lower hybrid waves (ω_{LH}), associated with the space-charge phenomena and particle gyromotion are eliminated from this new set of

equations. However, unlike the MHD equations, the gyrokinetic equations retain the all-important finite gyroradius effects, as well as the wave-particle interactions. These physical processes are believed to be the most important ingredients responsible for anomalous transport in tokamaks.

In gyrokinetic particle simulation,[3] each individual particle is transformed into a charged ring, and its gyrocenter is pushed with the appropriate forces at every timestep in the computer. The size of the ring is given by its gyroradius, which in turn is determined by the magnetic field and the perpendicular thermal velocity of the particle. In the limit of zero gyroradius, the gyrocenters become the usual guiding centers, and their motion is described by the well-known drift kinetic equation. Another unique feature of the gyrokinetic model is that the gyrokinetic Maxwell's equations (actually the gyrokinetic Poisson equation) include the density response due to ion-polarization effects. This important piece is missing in the usual drift kinetic (or guiding center) approximation and is an essential component for the microinstabilities as well as for the shear-Alfvén waves. The origin of the ion-polarization effects can be traced to the polarization drift of the gyrocenter in the presence of a spatially varying electric field. The appearance of these effects in the gyrokinetic Poisson equation enables us to use a grid of the size $\Delta x/\rho_s \cong 1$, where ρ_s ($>> \lambda_D$) is the ion gyroradius measured with the electron temperature. In other words, in the gyrokinetic simulation the shielding cloud becomes much larger.[3] When magnetic perturbations are introduced to the simulation, the size of the shielding cloud becomes even larger because the electron response to the shear-Alfvén waves is nearly adiabatic.[6]

These equations, which contain all the vital physics for low-frequency microinstabilities, are related in the fluid limit to the well-known "reduced MHD equations."[6] Most interestingly, according to the gyrokinetic formulation, the equilibrium MHD equations are actually a part of the gyrokinetic Maxwell's equations (Ampere's law). Thus, nonlinear gyrokinetics enables us to describe, with a single set of equations, the small-scale fluctuations associated with kinetic effects, as well as those arising from global MHD behavior. Since the numerical requirements are the same for these two types of phenomena, a single code will be sufficient to describe all the low-frequency phenomena for the tokamak discharge.

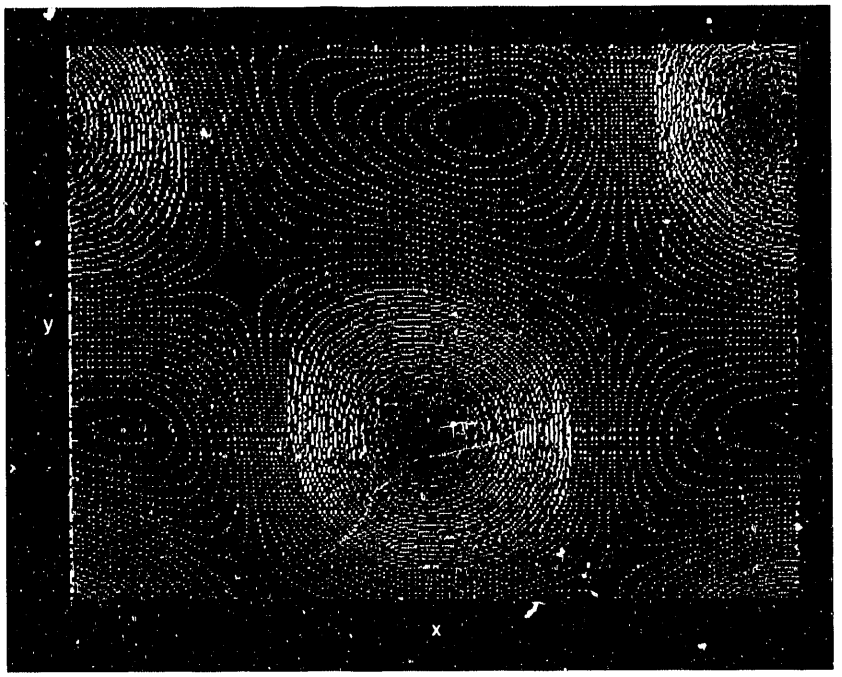


Figure 2. Ion trapping for the same drift-wave simulation described in Figure 1. The ion gyrocenters are shown in red, and again the equipotential contours for the electric field are indicated by yellow lines. In comparing the two figures, it is clear that significantly more ions than electrons are trapped.

The current plan, assuming adequate funding, is to have such a code developed within two to three years. Preliminary estimates indicate, for example, that we need about 75 hours on a Cray-2, using four central processing units (CPUs), to simulate a 10-ms discharge using one million particles per species for a tokamak with $a < 100\rho_s$, where a is the minor radius of the plasma. This is a costly but manageable exercise, especially if one views the simulation as a tokamak experiment in the computer. (It is much cheaper than the actual experiments!) Initial results using a massively parallel Connection Machine (CM2), with 65,536 CPUs, are also very encouraging. For the same discharge, we need only about 25 CPU hours. The improvement comes from the fact that particle simulations, which spend most of the time in gather/scatter operations, are most amenable to the parallel architecture.[7]

Simulating the whole machine is not the only way to study tokamak physics. At the present time, using the National Energy

Research Supercomputer Center's Crays, we have been studying the microinstabilities driven by the density and temperature inhomogeneities, for just a small slice of the tokamak. For this, we have been using the existing gyrokinetic particle codes in the electrostatic limit (that is, with no magnetic perturbations) in two- and three-dimensional slab geometries. One simple example, shown in Figures 1 and 2, is a two-dimensional (x, y) simulation of drift waves. The size of the simulation is $8\rho_s \times 8\rho_s$. The external magnetic field is aligned in the z direction but with a small tilt toward y, and the density inhomogeneity is in the x direction, for which the higher-density region is at the left end of the simulation box.

The drift wave instability comes from those electrons moving along the field lines in the y–z plane and having the same speed as the phase velocity of the electrostatic waves. Through the process known as "inverse Landau damping," the free energy associated with the plasma inhomogeneity is given to the waves. Those same "resonant" electrons are the medium of this exchange. When the electrons give their kinetic energy to the waves, the amplitude of the waves (that is, the electrostatic potential ϕ) grows to a large magnitude. As that amplitude becomes large enough, the resonant electrons moving parallel to the magnetic field also become trapped in the potential maximum of the waves, due to the motion known as the $\mathbf{E} \times \mathbf{B}$ advection, which describes the guiding center (or gyrocenter) movement perpendicular to both the electric and the magnetic fields. (Note that in this case the resonance trapping by the electrostatic field parallel to the magnetic field is much smaller than the $\mathbf{E} \times \mathbf{B}$ trapping by the electric field in the perpendicular direction of the magnetic field.) In turn, this $\mathbf{E} \times \mathbf{B}$ motion shuts off the energy exchange between the electrons and the waves. The evolution then reaches a saturation stage, in which the waves cease to grow but the plasma particles continue to move collectively to the right toward the low-density region. In other words, although individual particles can move chaotically in every possible direction, on the average they move steadily to the right, giving rise to the infamous anomalous transport.

This saturated stage is shown in Figure 1, where the red dots are the electron guiding centers and the yellow lines are the equipotential contours for the electric field (or the waves). Trapping of the electrons in the potential maximum is clearly visible. However, at this stage of the development, the electrons undergo continuous trapping and detrapping by the waves. The reason is that the trapped electrons, which move with the wave and give rise to the diffusion (transport), lose their momentum (or velocity) in the process and become detrapped and move away from the potential maximum. However, after moving along constant potential contours, they can be recaptured by another potential maximum and contribute again to diffusion. In addition, nonresonant electrons can also be trapped by the waves through collisions with the ions and/or through scattering by the waves. Reference 4 gives a detailed account of this process.

The ions also diffuse to the right toward the low-density region, and the diffusion rate is the same as for the electrons (commonly referred to as the ambipolarity condition). However, the physical process is different.[4] As shown in Figure 2, the number of ion gyrocenters (red dots) trapped in the potential maximum is significantly higher. Since the ions are much more massive, they are practically stationary in relation to the wave motion in the y direction at the onset of the instability. However, when the amplitude of the wave grows large, the ions are suddenly $\mathbf{E} \times \mathbf{B}$ trapped by the waves in significant numbers. Once trapped, they cannot easily become detrapped as the electrons do. Therefore, they move with the wave in the x direction and give rise to the diffusion.

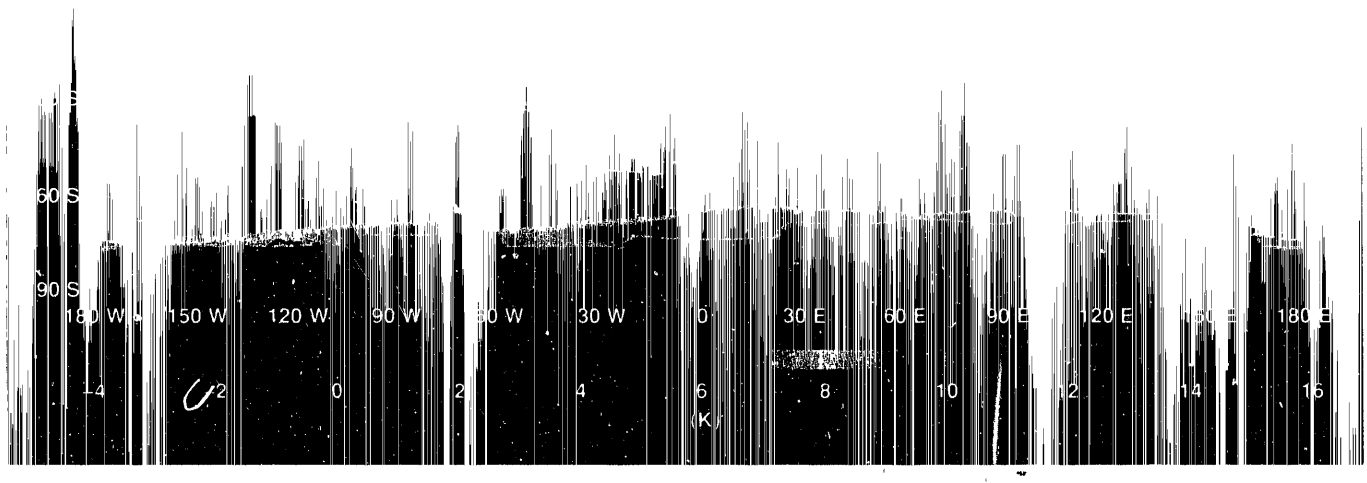
This detailed account of the different behavior of electrons and ions highlights the uniqueness and versatility of gyrokinetic particle simulation. Conventional fluid simulation and particle codes cannot easily give us this type of physical insight. Unfortunately, even for this simple example, a

theoretical description of the wave motion in the x direction, which is responsible for the diffusion, is still not available. However, it is at least gratifying to know that the motion, which is always directed toward the low-density region in the simulation, satisfies the second law of thermodynamics. On the other hand, this example shows how little we know about anomalous transport and how important it is to develop the gyrokinetic particle simulation capability. With the inauguration of the Transport Initiative in the magnetic fusion community and the availability of present-day supercomputers, the time is ripe for us to embark upon this worthwhile undertaking. ■

References

1. J.M. Dawson, "Particle Simulation of Plasmas," *Rev. of Mod. Phys.* **55**, 403 (1983).
2. C.K. Birdsall and A.B. Langdon, *Plasma Physics via Computer Simulation* (McGraw-Hill, New York, 1985).
3. W.W. Lee, "Gyrokinetic Particle Simulation Model," *J. Comput. Phys.* **72**, 243 (1987).
4. A.M. Dimits and W.W. Lee, "Nonlinear Mechanisms for Drift Wave Saturation and Induced Particle Transport," Princeton Plasma Physics Report PPPL-2659 (1989).
5. W.W. Lee, "Gyrokinetic Approach in Particle Simulation," *Phys. Fluids* **26**, 556 (1983).
6. W.W. Lee et al., "Numerical Properties of a Finite-β Gyrokinetic Plasma," in preparation.
7. J.V.W. Reynders and W.W. Lee, "Comparison of a 2D Finite-β Gyrokinetic Simulation Code on Serial and Parallel Computers," 13th Conf. on Numerical Simulation of Plasmas, Paper-PMB8 (1989).

W.W. Lee received his education at National Taiwan University and Northwestern University. He worked as an accelerator physicist at Fermi National Accelerator Laboratory from 1970 until 1974. Since then, Dr. Lee has been associated with the Princeton Plasma Physics Laboratory, where he is currently a Principal Research Physicist. His research interest includes microinstabilities in tokamaks and space-charge effects in particle beams.



Applied Mathematical Sciences

Analyzing Chaos: A Visual Essay in Nonlinear Dynamics

Celso Grebogi, Edward Ott, Frank Varosi, and James A. Yorke, University of Maryland

Even relatively simple mathematical systems can behave in surprisingly complex and erratic ways, with the smallest changes to the system's initial conditions leading to unpredictable behavior. The study of nonlinear dynamics—and specifically of chaotic dynamics—attempts to analyze the chaos inherent in some mathematical systems. The overall thrust is to increase the ability to anticipate how a dissipative nonlinear system will unfold in the future from a given initial state.

Although first hinted at in the late 19th century by the French mathematician Henri Poincaré, chaotic dynamics has developed as a field of study only in the last two decades. Advances in computer technology are providing the computational tools needed to understand the vastly complex behavior of nonlinear systems. Our research at the University of Maryland—where the term *chaos* was first used to describe this field of work—focuses on discovering basic chaotic phenomena and on

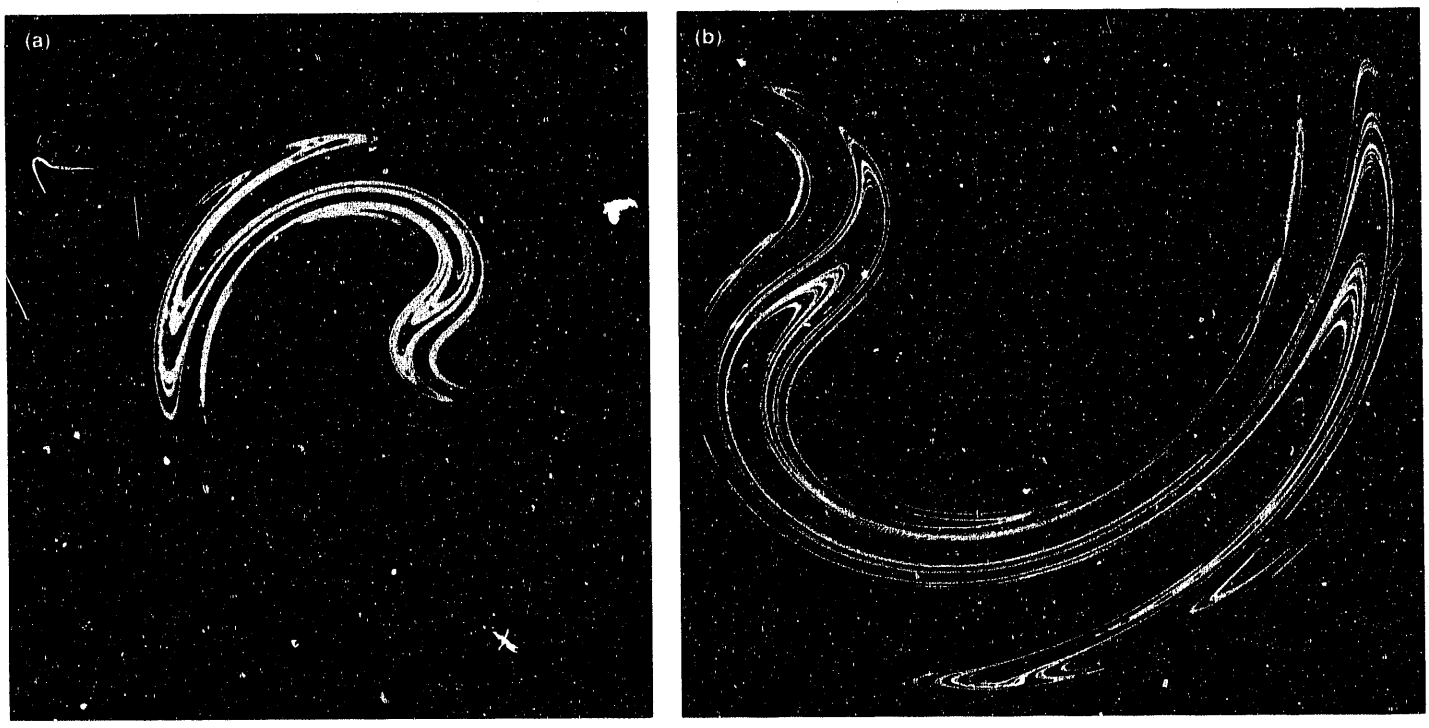


Figure 1. These pictures show a chaotic attractor for an idealized model of an optical switch for a laser system with fixed parameter value. The points of the attractor represent the electromagnetic field (phase and amplitude) in the optical cavity. For this set of parameters, the chaotic attractor has just experienced a "crisis," which results in greatly increasing the size of the attractor. Prior to this sudden transition, the attractor was restricted to the bright central region. The pictures illustrate how the orbit of the model map breaks free of the bright region when a crisis occurs. The model map describing the optical switch is

$$z_{n+1} = A + B\, z_n \exp\{ik - ip/(1 + |z_n|^2)\} \ .$$

In this example, $A = 0.85$, $B = 0.9$, $k = 0.4$, and the control parameter p is the amplitude of the laser pulse entering the optical switch, a mathematical number. A crisis occurs when $p = 7.26994894$ (the crisis value). If p is less than the crisis value, the attractor is restricted to the bright central region. View (b) is an enlarged version of view (a).

Acknowledgment

This work was supported by the U. S. Department of Energy.

References

1. J. Carlson, V.R. Pandharipande, and R.B. Wiringa, *Nucl. Phys.* **A401**, 59 (1983).
7. R.B. Wiringa, R.A. Smith, and T.L. Ainsworth, *Phys. Rev.* **C29**, 1207 (1984).
8. R. Schiavilla, V.R. Pandharipande, and D.O. Riska, *Phys. Rev.* **C40**, 2294 (1989) and CEBAF preprint (1989).
9. D.H. Beck, *Phys. Rev. Lett.* **64**, 268 (1990).

Joseph A. Carlson is a staff scientist in the medium-energy nuclear theory group of the

developing nonspecific mathematical models that can apply to many scientific areas. The potential applications are wide-ranging: from ecology to engineering, from physics to fluid mechanics. Understanding nonlinear dynamics could lead to more precise weather forecasting, better aerodynamic engineering, more efficient combustion chambers, more accurate predictions about insect populations—the list of possibilities is long.

The three figures included here derived from our work. Each set of graphics colorfully depicts what happens when a specific differential equation is computed through a great many (perhaps a billion) iterations. A typical way of evaluating a system's long-term behavior is to observe its asymptotic behavior in phase space, which often requires looking at smaller and smaller regions of phase space on a finer and finer scale. This is where supercomputers come in. Typically one has to integrate more than one million initial conditions to produce a single picture.

In developing these mathematical models, there are two kinds of chaotic sets that commonly emerge: *attractors* and *repellers*. An attractor (Figure 1) is a complex mathematical set in which the trajectory remains confined within a given region, although it may bounce around within that region. On the other hand, a repeller (Figure 2) is a mathematical set that winds up pushing away neighboring points from the given region.

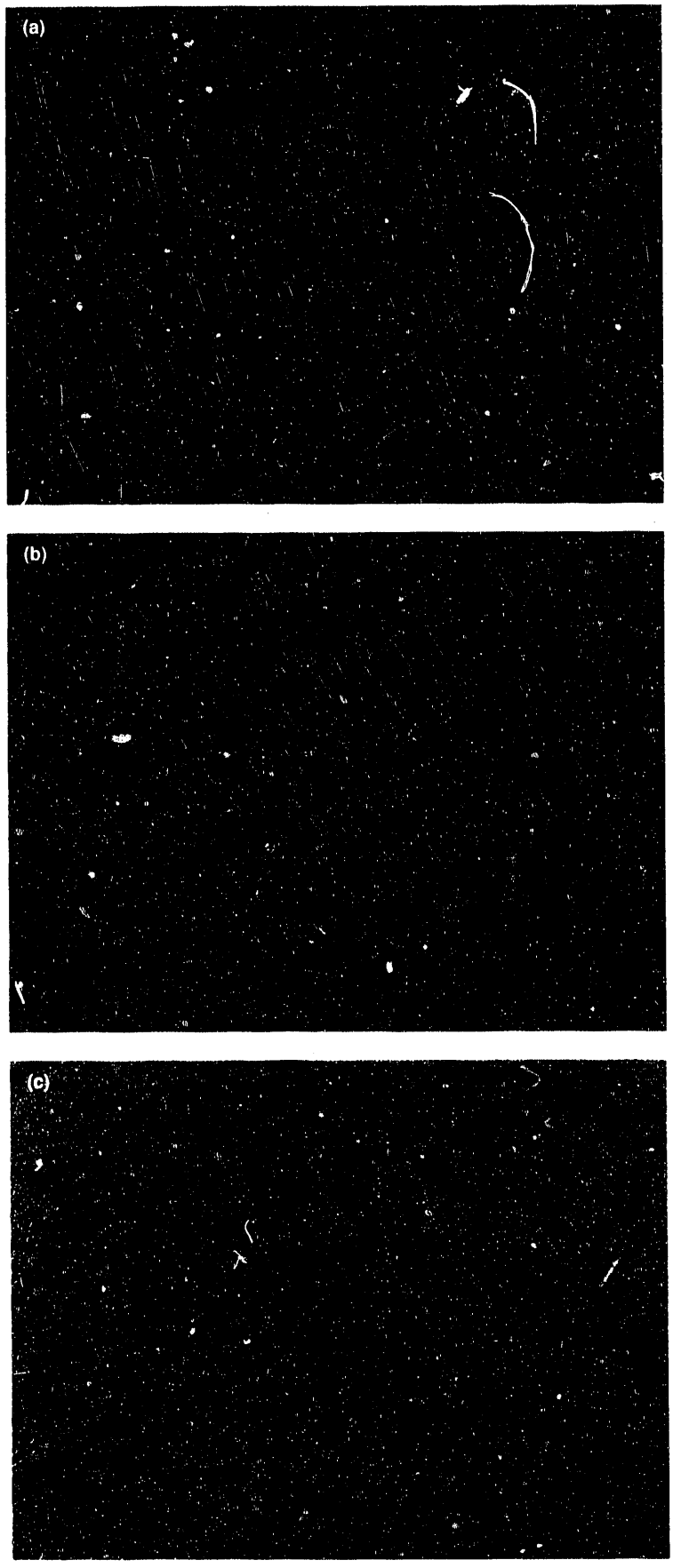
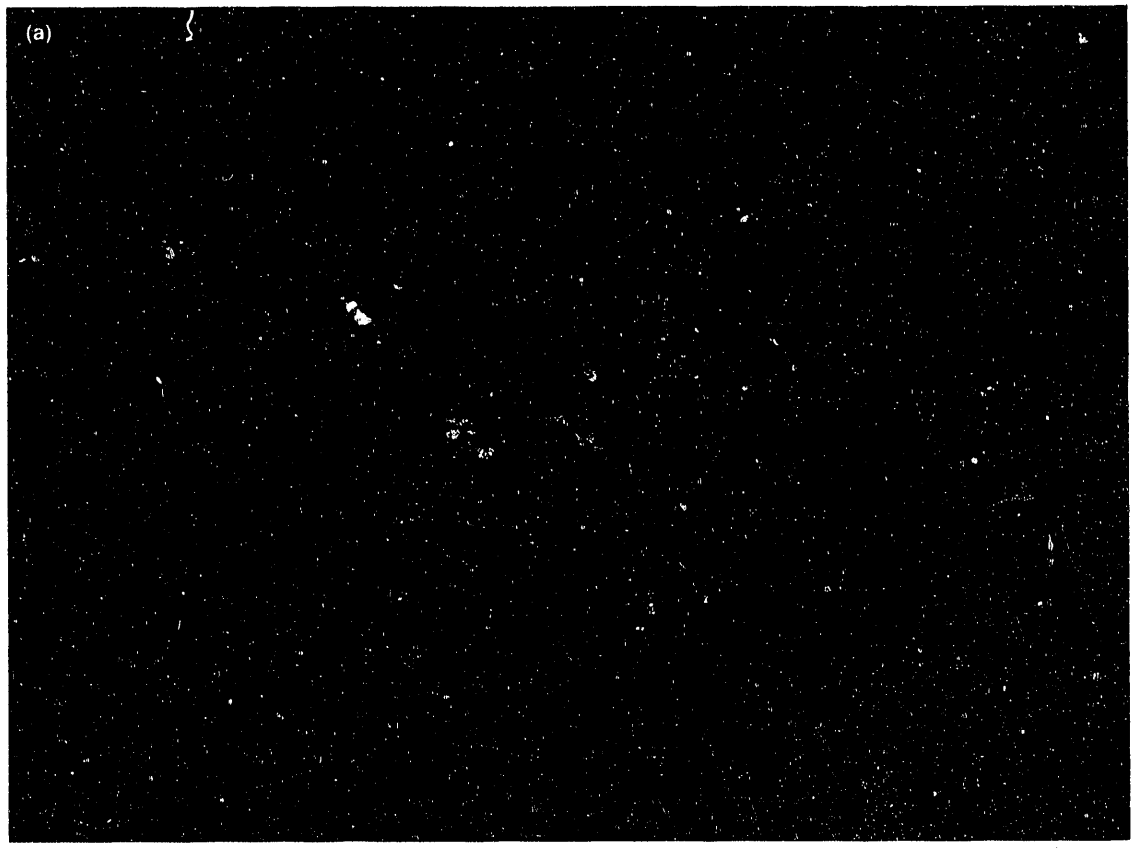


Figure 2. Even systems as simple as a periodically forced damped pendulum can have complex behavior. These images show initial pendulum positions (measured horizontally) and velocities (measured vertically). Orbits starting at points in the red region eventually settle into one type of periodic motion, while orbits starting in the blue region yield a different type of periodic motion. The boundary between these regions is fractal. The brighter the shade of red or blue, the longer it takes to settle into the corresponding motion. The differential equation for the pendulum used to generate these pictures is

$$\frac{d^2x}{dt^2} + \frac{1}{5}\frac{dx}{dt} + \sin x = 2\cos t \quad .$$

View (b) magnifies the upper-left-hand corner of view (a), characterized by a pair of reddish swirls; view (c) magnifies still further one of those swirls.



Attractors versus repellers. The terms themselves neatly convey logic and coherence, but these mathematical sets are in fact immensely complicated. Nowhere is this more apparent than in the concept of a *fractal basin boundary*. (The dimension of a fractal boundary is not an integer; for example, it may be 1.26 or perhaps 1.73.) With fractal boundaries, complexity in fine-scale structure remains, no matter how close you zoom in. The successive magnifications in Figures 2 and 3 illustrate the intricate mixture of colors of fractal basin boundaries.

Although there have been many recent advances in the study of chaotic nonlinear systems, a host of more challenging problems remains to be solved. Most concepts developed to date have arisen through the study of low-dimensional systems. The study of higher-dimensional systems is expected to involve a whole new phenomenology, requiring orders of magnitude more computer time. One question that interests us is the relationship between the dimension of a chaotic attractor and the minimum dimension of the phase space necessary to describe its dynamics—an issue that should keep the supercomputers busy for years to come. ■

Celso Grebogi is a professor in the Department of Mathematics, in the Laboratory for Plasma Research, and at the Institute for Physical Science and Technology at the University of Maryland. His main research interest is the dynamics of dissipative systems. In particular, he has done extensive work on computational methods to investigate chaotic processes.

Edward Ott is a professor in the Departments of Electrical Engineering and of Physics and Astronomy at the University of Maryland. His interest in chaotic dynamics originally arose from his work in plasma physics.

Frank Varosi is a consultant to NASA's Goddard Space Flight Center.

James A. Yorke, a mathematician, is director of the Institute for Physical Science and Technology at the University of Maryland. He was one of the pioneers in the study of chaotic dynamics and originally applied the term chaos *to describe this field.*

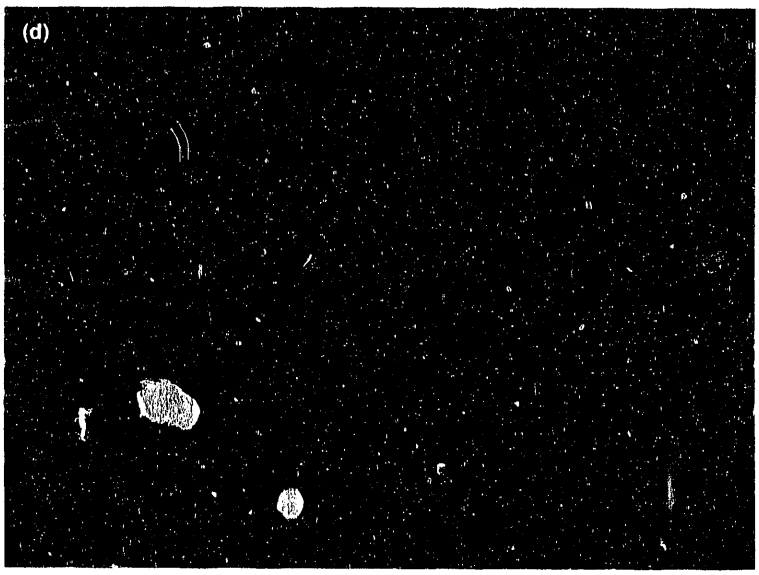


Figure 3. These pictures, which represent a pendulum allowed to swing through 360 degrees, show initial pendulum positions (measured horizontally) and velocities (measured vertically). Orbits starting at points in the red region are attracted to a periodic orbit that winds clockwise. The blue represents initial points that have the opposite asymptotic behavior, winding counterclockwise. The green and yellow regions designate initial points that are respectively attracted to two other types of periodic motion. Hence, in this case there are four possible states that the system can eventually settle into, depending on the initial condition of the system. For each one of these regions, the darker the shade of the color, the longer it takes to settle into the corresponding motion. The intricately mixed patterns of colors are due to the fractal nature of the boundary separating the regions. The pendulum differential equation used to generate these pictures is

$$\frac{d^2x}{dt^2} + \frac{1}{10}\frac{dx}{dt} + \sin x = \frac{7}{4}\cos t \quad .$$

This is a series of progressive magnifications. View (b) enlarges the yellow streak at the extreme right center of (a); view (c) magnifies the brown and green swirls in the upper-right-hand portion of (b); and view (d) magnifies (c) still further.

Supercomputing and Research in Theoretical Chemistry

William A. Lester, Jr., University of California, Berkeley
Lawrence Berkeley Laboratory

Supercomputers play a crucial role in understanding the electronic structure of isolated atoms and molecules, the interactions between molecular species that govern reaction pathways, and the dynamics of molecular collisions.

Supercomputing is an essential ingredient for a number of theoretical chemistry research projects my group is pursuing at the University of California, Berkeley (UCB) and at Lawrence Berkeley Laboratory (LBL). We focus primarily on *quantum chemistry* (using the laws of quantum mechanics to determine the properties of atoms and molecules) and on *collision dynamics* (determining what happens when atoms and molecules collide). Ultimately, our goal is to be able to predict more accurately the stability of individual molecules, the forces between molecules, and the probabilities of a chemical reaction, of energy exchange between colliding atoms and molecules, and of energy flow in an individual molecule.

The three areas of active research discussed here include:

(1) Developing and applying the highly accurate quantum Monte Carlo method for determining electronic structure, as an alternative to the methods typically used in chemistry for computing potential energy surfaces and other properties of molecular systems.

(2) Studying reaction pathways—that is, calculating the energetics and geometries of stable reagent and product species, transient intermediates, and transition states for reactions that are primarily related to combustion.

(3) Determining collisional energy transfer—that is, determining the detailed mechanisms and probabilities of energy transfer between translational and internal (rotational, vibrational, and electronic) degrees of freedom.

Quantum Monte Carlo Method

Our major effort is directed toward developing the quantum Monte Carlo (QMC) method, a stochastic approach that solves the Schrödinger equation, which describes the quantum mechanics of molecules. The Schrödinger equation can be solved exactly only for the very simplest systems, and we are looking for ways to obtain high accuracy for complex systems. The QMC method has generated considerable interest in the theoretical chemistry community because of the high accuracy achieved with it in calculating the properties of atoms and small molecules.

In the QMC approach, a computer "experiment" is performed in which an ensemble of random walks (the coordinates of which, at any given time, represent a configuration of the electrons) evolves to an equilibrium distribution, and properties (such as the energy) are measured. At any time after equilibrium has been reached, the ensemble of configurations is a random sample drawn from the probability distribution $f(\mathbf{R}) = \Psi_T(\mathbf{R})\Phi(\mathbf{R})$, where the coordinate-vector $\mathbf{R}$ is the multidimensional vector describing the full many-electron system. Here $\Psi_T(\mathbf{R})$ is a simple trial wave function used for importance sampling. The function $\Phi(\mathbf{R})$ is the lowest-energy eigenfunction of the Schrödinger equation that is not orthogonal to Ψ_T. Convergence to the lowest-energy state results from an essential feature of the mapping of the Schrödinger equation into its diffusion equation analog: namely, that time in these two equations differs by a factor of i. Thus, when a time-dependent molecular-state vector is expanded in energy eigenfunctions multiplied by $\exp(-iEt/\hbar)$, in

imaginary time one obtains a series in which only the lowest-energy term (that is, Φ) survives at large times t. If Ψ_T is orthogonal to the exact lowest-energy state, one projects out the ground state, and convergence will be to the next-lowest energy.

Although neither b nor f is known analytically, one can nevertheless sample desired quantities from the equilibrium distribution f. Averages taken with respect to f are known as mixed averages. For example, sampling a quantity A in equilibrium gives (in the limit of large N) the average

$$\langle A\rangle_f = \langle \Psi_T |A| \Phi\rangle \quad , \qquad (1)$$

where the Dirac notation used here implies normalized functions. The correct expectation value of A for a state Φ is $\langle\Phi|A|\Phi\rangle$; however, in computing any property for which Φ is an eigenstate, there is no difference between these two averages. This follows since the eigenvalue can be taken out of the integral in Equation 1. In particular, to compute the energy, one samples the quantity $E_L(\mathbf{R}) = \Psi_T^{-1}(\mathbf{R})H\Psi_T(\mathbf{R})$. Then

$$\langle E\rangle_f = \langle \Phi|H|\Psi_T\rangle = E_0 \quad , \qquad (2)$$

where E_0 is defined by $H\Phi = E_0\Phi$.

In its comparatively brief history, QMC has demonstrated the capability for yielding energies of atomic and molecular systems that are typically better than the best available results from *ab initio* (basis set expansion) methods. Because of its statistical character, QMC provides estimates of physical quantities with computed statistical uncertainty. For this reason, statistical deviations need to be small compared to the quantity of interest to obtain useful results. For all Monte Carlo methods, reducing statistical uncertainty by a factor of 10 requires 100 times more computing, which leads to the need for supercomputers. QMC's high potential for contributing to the understanding of molecular properties and processes has made this approach an area of intense investigation with supercomputers. These machines have enabled the rapid testing and exploration of new ideas on timescales that would not otherwise have been possible, as well as the highly accurate determination of quantities of chemical interest.

Reaction Pathways

Determining reaction pathways is of primary importance in chemistry. One example of a reaction pathway study is the investigation, with Sheng-yu Huang and Brian L. Hammond, of the reaction of ground-state triplet oxygen (O) atoms with allene (propadiene-C_3H_4). Small unsaturated hydrocarbons play an important role in combustion processes. The reaction of allene with $O(^3P)$ has long been accepted to proceed by the addition of O to the central carbon atom (CCA), followed by a spin-flip (triplet-singlet) transition to form vibrationally excited cyclopropanone (C_3H_4O). The latter then dissociates to form the products CO and ethylene C_2H_4.

However, in recent molecular beam experiments, Nobelist Yuan T. Lee of LBL and UCB observed the final products allenyloxy (formyl-vinyl H_2CCCHO) radical and hydrogen (H) atom formed from O-atom attack on a terminal carbon atom (TCA). Lee was thus confronted with alternative products from the reaction of allene with $O(^3P)$, implying a new reaction pathway. To understand theoretically the mechanisms that lead to these two sets of products, we undertook first principles (*ab initio*) quantum mechanical calculations of the barrier heights encountered in both the TCA and the CCA pathways. In addition, we performed computational studies to characterize the geometry and energetics of the diradical precursor to the allenyloxy radical formed in the TCA. In an earlier study of this type for the O + ethylene system, we were successful in predicting the pathways leading to similar structures not encountered before the experiments by Lee and his collaborators.

Our computer program uses an algorithm that computes, by an analytical procedure, the gradients and second derivatives of the energy with respect to nuclear coordinates. This method makes possible the efficient determination of equilibrium and transition-state geometries and energies for the low-lying electronic states of each of the species of interest in the reaction. One of these, C_3H_4O with four first-row atoms, is considered relatively large by current standards of

ab initio computations. To carry out quantitative estimates of these quantities for $O + C_3H_4$ took approximately 60 hours of central processing unit (CPU) time on the Cray X-MP at the National Energy Research Supercomputer Center (NERSC) and required the maximum accessible computer memory and disk storage on that system. The results of our study are represented in Figure 1, a correlation diagram showing relative energies of reactants, transition states, and products.

Our study confirmed that the allenyloxy radical could be formed directly under the conditions of the experiment. Other reaction products have been suggested from the analysis of recent experiments, and these will be the subject of future computational studies to ascertain their relative importance and mode of formation. At our present level of computation, we have confirmed that our absolute estimates of barrier heights are too high. However, our O-ethylene study supports the usefulness of the relative energy differences shown in Figure 1 for the TCA and CCA transition states—that is, for H_2CCCH_2O and H_2CC-OCH_2. To improve these estimates would require a large increase in CPU time, disk space, and memory.

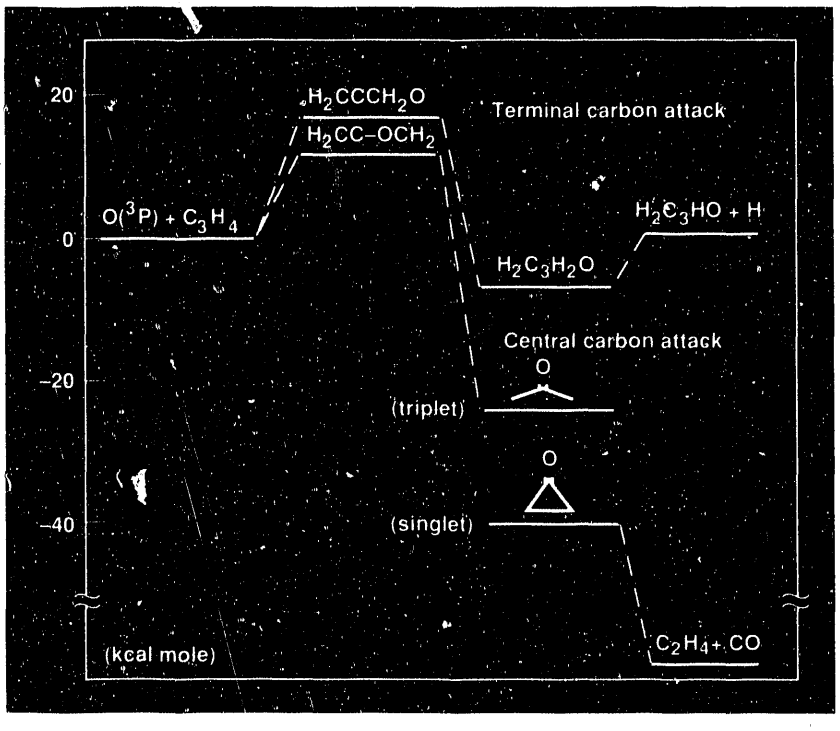


Figure 1. A correlation diagram for the reaction pathways of ground-state oxygen $O(^3P)$ and allene (C_3H_4). This figure provides information on the energies of reagents, products, transient species, and stable intermediates of the reaction. At the center are the relative energy levels of the transition states H_2CCCH_2O and H_2CC–OCH_2. A transition state is a theoretically important unstable structure at the maximum energy along the minimum energy path connecting reactants to products. The energy scale is at left; the zero of energy is arbitrary.

Collisional Energy Transfer

The quantum Monte Carlo and expansion methods described above provide the potential energy surface (or surfaces) governing the dynamics of nuclear motion. We have also recently focused on theoretical studies of collisional energy transfer. Specifically, the process of interest is the transfer of relative translational energy to vibrational and electronic degrees of freedom for the H_2 molecule in its first excited stable state, caused by the impact of helium (He). Pascal Pernot, a visiting postdoctoral researcher from Paris, carried out time-dependent wave-packet calculations of the inelastic processes, using potential energy surfaces and couplings computed using similar *ab initio* methods. A significant benefit of time-dependent methods such as this—not possible with time-independent approaches—is the physical insight made possible by visually following the time evolution of the system.

Figure 2 presents snapshots of the temporal behavior of a wave packet for the system, taken from a video constructed from computed results. The wave packet initially is localized on the excited-state surface. The color coding is as follows: white indicates maximum probability, blue and magenta indicate the lowest probability, and yellow designates zero probability. As time proceeds, the wave packet spreads and makes transitions to the ground-state surface. Figure 2 shows clearly the dominance of the atomization channel (that is, the formation of H + H + He) over the creation of bound ground-state H_2 and He at the energy (0.5 eV) of the calculation. This finding is indicated by the absence of probability density (that is, of any color other than yellow) for large He–H_2 distances in the frame at lower right (ground state at time = 3500). ■

Related Readings

1. Peter J. Reynolds, David M. Ceperley, Berni J. Alder, and William A. Lester, Jr., "Fixed-Node Quantum Monte Carlo for Molecules," *J. Chem. Phys.* **77**, 5593 (1982).
2. Peter J. Reynolds, Robert N. Barnett, Brian L. Hammond, and William A. Lester, Jr., "Molecular Physics and Chemistry Applications of Quantum Monte Carlo," *J. Stat. Phys.* **43**, 1017 (1986).
3. Brian L. Hammond, Peter J. Reynolds, and William A. Lester, Jr., "Damped-Core Quantum Monte Carlo Method: Effective Treatment for Large-Z Systems," *Phys. Rev. Lett.* **61**, 2312 (1988).
4. P. Pernot, R.M. Grimes, W.A. Lester, Jr., and C. Cerjan, "Quantum Time-Dependent Study of the Scattering of He by $H_2^+(B^1\Sigma_u^+)$," *Chem. Phys. Letts.* **163**, 297 (1989).
5. Sheng-yu Huang, Zhiwei Sun, and William A. Lester, Jr., "Optimized Trial Functions for Quantum Monte Carlo," *J. Chem. Phys.* **92**, 597 (1990).
6. William A. Lester, Jr., and Brian L. Hammond, "Quantum Monte Carlo for the Electronic Structure of Atoms and Molecules," to be published in *Annual Reviews of Physical Chemistry* **41** (1991).
7. B.L. Hammond, S.-Y. Huang, W.A. Lester, Jr., and M. Dupuis, "Theoretical Study of the $O(^3P)$ + Allene Reaction," to be published in *J. Phys. Chem.*

William A. Lester, Jr., is Professor of Chemistry at the University of California, Berkeley, and Faculty Senior Scientist at Lawrence Berkeley Laboratory. He holds B.A./B.S. (1958) and M.S. (1959) degrees from the University of Chicago and a Ph.D. (1964) from the Catholic University of America. He previously held positions in government (National Bureau of Standards) and in industry (IBM Research Division), as well as in academia (Theoretical Chemistry Institute, University of Wisconsin).

Figure 2. Probability plots of the time evolution of a wave packet for collision of helium atoms with hydrogen molecules in the first excited singlet state ($B^1\Sigma_u^+$). The color coding reflects the quantitative (computed) magnitude of the probability, with white indicating the highest probability, magenta indicating the lowest probability, and yellow designating zero probability. These snapshots were taken from a video constructed from computed results and are quantitative.

Monte Carlo Simulations of Light Nuclei

Joseph A. Carlson, Los Alamos National Laboratory

New computational methods test existing models of interactions within an atomic nucleus.

Light nuclei are an important testing ground for nuclear physics. They are simple enough to allow reliable microscopic calculations with realistic interaction models, yet they offer the opportunity to study many intriguing questions in nuclear physics. These calculations are aimed toward a better understanding of the interactions between neutrons and protons within an atomic nucleus, and also toward developing a fuller understanding of the static and dynamic properties of nuclei. These topics are also of great interest experimentally, current (Bates, Saclay) and future (CEBAF) electron accelerators measure elastic and inelastic response of these nuclei.

A primary concern in all such calculations is testing the existing theoretical models of nuclear interactions. Accurate nucleon-nucleon interaction models have been developed that fit the two-body (deuteron and nucleon-nucleon scattering data) properties, but only recently has it been feasible to solve for even small nuclei ($A > 2$) because of the highly nonperturbative nature of the problem. Recently we have concentrated on exact alpha-particle calculations with Green's function Monte Carlo methods, and especially on tests of three-body interactions in this system. Three-body forces are those terms in the interaction that cannot be written as a sum over pairs of particles. These terms arise from suppression of some degrees of freedom in the nucleus. Although these three-body forces are weak in comparison to the two-body terms, they are nevertheless important in obtaining quantitative agreement with experimental results.

Our starting point is a nonrelativistic Hamiltonian with only nucleon degrees of freedom:

$$H = -\frac{\hbar^2}{2m}\sum_i \nabla_i^2 + \sum_{i<j} V_{ij} + \sum_{i<j<k} V_{ijk} + \ldots \quad , \quad (1)$$

for which we attempt to solve the Schrödinger equation,

$$H|\Psi\rangle = E|\Psi\rangle \quad (2)$$

for the quantum-mechanical states $|\Psi\rangle$, and to determine the static and dynamic properties of the nucleus. In this simplified picture, nucleons are point particles containing only two internal degrees of freedom: the spin (up or down) and the isospin (proton or neutron). We have suppressed mesonic degrees of freedom and any internal excitations of the nucleons; both simplifications have important consequences.

The nucleon-nucleon interaction V_{ij} is determined by fits to two-body scattering data and properties of the deuteron. At long distances, all such models reduce to a one-pion-exchange potential arising from the diagram in Figure 1(a), and they often represent a sum of heavier meson exchanges at shorter distances. All nucleon-nucleon (or two-body) interaction models depend very strongly on the internal degrees of freedom, the spins (σ_i) and isospins (τ_i) of the nucleons. The simplest models may be written:

$$V_{ij} = \sum_{m} V^{m}(r_{ij}) O_{ij}^{m}$$

$$O_{ij}^{m} = [1, \sigma_i \cdot \sigma_j, S_{ij}, L \cdot S_{ij}] \otimes [1, \tau_i \cdot \tau_j] \quad , \quad (3)$$

where S_{ij} is the spin tensor operator and $L \cdot S_{ij}$ is the spin-orbit term.

This strong spin-dependence necessitates solving $2^A A!/Z!(A-Z)!$ coupled differential equations in $3A$ dimensions for a nucleus of A nucleons (that is, of Z protons plus $A-Z$ neutrons).

The Hamiltonian in Equation 1 also contains a three-nucleon interaction (TNI). Models of the two-nucleon interaction V_{ij} that fit available two-body data fail to describe three- and four-body nuclei adequately. Unless three-body forces (V_{ijk}) are included, calculations indicate that these nuclei are underbound by roughly 1 and 4 MeV, respectively, out of a total of 8 and 28 MeV. A long-range contribution to the TNI [Figure 1(b)] arises from two pion exchanges, the first exchange exciting one nucleon to a delta resonance which then decays, emitting a pion that is absorbed on a third nucleon. A three-body force clearly results from suppressing the internal structure of the nucleons, but there are many other diagrams besides the one shown in Figure 1(b) that will contribute to the TNI. It is difficult to determine the full three-nucleon interaction in any fundamental way, due to the complexities of the strong interaction. For this reason, we take the spin-isospin dependence of the long-range part of the TNI from the diagram in Figure 1(b) and add a short-range phenomenological repulsion.[1] The strength of these terms is adjusted to fit the binding energies of three- and four-body nuclei.

Algorithms

We have used Monte Carlo methods, both variational Monte Carlo (VMC)[1,2] and Green's function Monte Carlo (GFMC),[3–5] to study the ground state of the alpha particle. VMC methods determine the best approximate solution of a given form, while GFMC provides an exact solution, subject only to statistical errors.

VMC methods employ the Metropolis algorithm[1,2,6] to determine the ground-state properties of the nucleus. Given a trial wave function Ψ_T containing a set of parameters $\{\alpha\}$, the variational principle can be employed to optimize the set $\{\alpha\}$ and produce a good approximation to the ground-state wave function. To minimize $\langle H \rangle$, one must compute:

$$\langle H \rangle = \frac{\int d\mathbf{R}\, \langle \Psi_T^\dagger(\mathbf{R}) H \Psi_T(\mathbf{R}) \rangle}{\int d\mathbf{R}\, \langle \Psi_T^\dagger(\mathbf{R}) \Psi_T(\mathbf{R}) \rangle} , \quad (4)$$

where the integrals run over the coordinates of all particles, and the angled brackets indicate sums over all the spin and isospin states of the nucleons. The sums are performed explicitly, and the spatial integrals are performed using Metropolis Monte Carlo methods.

We choose

$$|\Psi_T\rangle = S \prod_{i<j} F_{ij} |\Phi\rangle , \quad (5)$$

where $|\Phi\rangle$ is a Slater determinant of one-body states, the F_{ij} are pair correlation operators, and S is a symmetrization operator indicating a sum over all orders of pair correlations. The operators F contain the same state-dependence as the interaction and are obtained by solving a set of two-body differential equations that incorporate the variational parameters $\{\alpha\}$.

These problems require tremendous computer resources because of the interaction's strong state dependence. The operators O_{ij} (see Equation 3) are sparse in the full spin-isospin space, but even so the time required to compute the wave function grows as

$$\frac{A(A-1)}{2} 2^A \frac{A!}{Z!(A-Z)} .$$

For this reason, we are limited to studying only light nuclei; to date only systems with

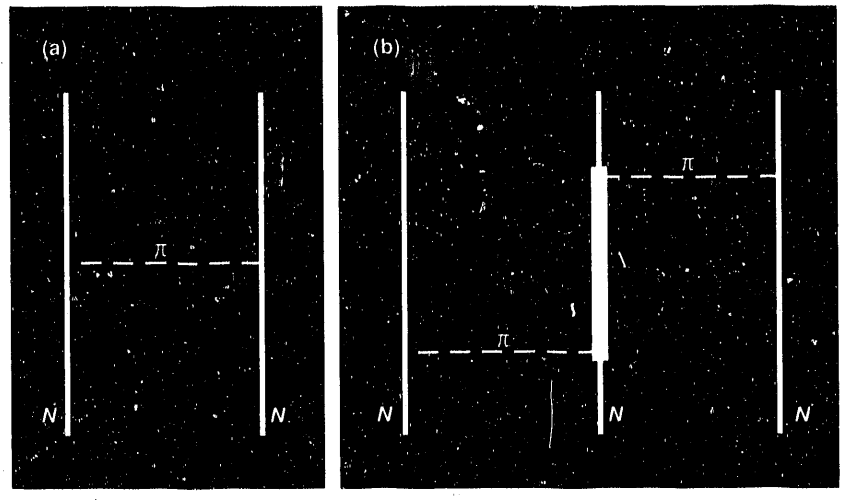


Figure 1. (a) The one-pion-exchange diagram, which gives the longest-range contribution to the nucleon-nucleon interaction. (b) The two-pion-exchange, three-nucleon-exchange diagram involves the excitation of a nucleon to a delta resonance. This diagram gives the longest-range contribution to the three-nucleon interaction.

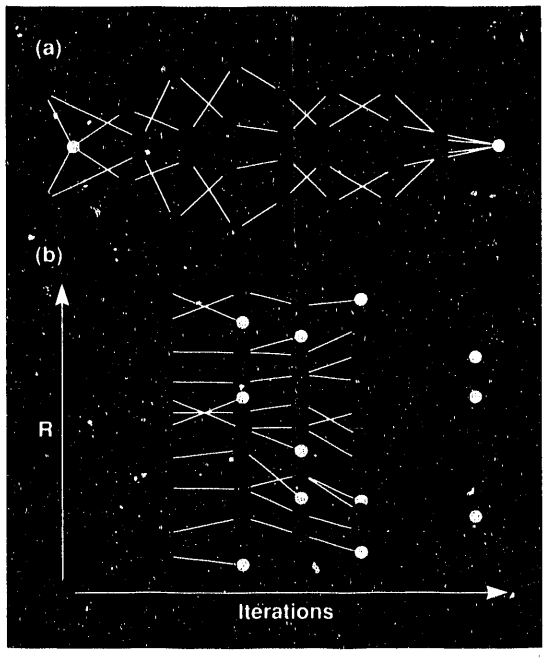


Figure 2. (a) Sampling of internal paths to determine the Green's function (Equation 9), with the yellow circles representing the end points and the red circles representing intermediate sampled points. A set of paths is sampled for each particle. (b) Iteration of Equation 7, with each circle representing a set of coordinates and amplitudes for all particles and the lines representing a set of sampled paths, as above. Configurations diffuse in the $3A$ dimensional space and are replicated (blue circles) or eliminated (yellow circles), depending on the value of the Green's function.

$A \leq 5$ have been studied. With increasing computer power and modest algorithm developments, we should be able to extend these methods to $A = 8$, allowing us to study very neutron-rich nuclei. This isospin dependence is very important in microscopic calculations of the properties of neutron stars.

Variational calculations, although valuable, should be subjected to tests from exact algorithms. This is especially important when trying to calculate the effect of three-nucleon interactions, where even small errors in variational calculations can be significant. GFMC methods, developed over the last few years, in principle allow one to calculate exactly the ground states of quantum systems, even when the interactions are strongly state dependent.[4,5]

The ground state of a quantum system can be determined through the projection

$$|\Psi_0\rangle = \lim_{\tau\to\infty} \exp(-H\tau)\,|\Psi_T\rangle \quad , \qquad (6)$$

where $|\Psi_T\rangle$ is an initial trial state, typically a trial wave function used in a variational calculation. The exponential in imaginary time τ damps all excitations present in the approximate solution. Calculating the full propagator explicitly is not feasible, of course, since it requires a knowledge of all the eigenstates of the system.

However, the full propagator over time τ can be split into many timesteps, each of duration $\Delta\tau$:

$$\exp(-H\tau) = \prod_{1}^{n} \exp(-H\Delta\tau)$$

$$= \int G_{\Delta\tau}(\mathbf{R}_n,\mathbf{R}_{n-1}) \ldots G_{\Delta\tau}(\mathbf{R}_1,\mathbf{R}_0) \quad . \qquad (7)$$

The short-time propagator $G_{\Delta\tau}$ can be evaluated explicitly, and Monte Carlo methods can be used to sample the full set of propagators. Making the timesteps $\Delta\tau$ very small ensures that the errors introduced by using an approximate short-time propagator are small. The calculation of the ground state (Equation 6) proceeds along lines very similar to the Monte Carlo methods used to simulate neutron transport. One computes trajectories of the system, with each trajectory followed by sampling the short-time propagator.

For a static interaction, we employ the following formula to approximate the short-time propagator:

$$G_{\Delta\tau}(\mathbf{R},\mathbf{R}') \approx G^0(\mathbf{R},\mathbf{R}') \prod_{i<j} \frac{g_{ij}(\vec{r}_{ij},\vec{r}'_{ij})}{g^0_{ij}(\vec{r}_{ij},\vec{r}'_{ij})} \quad . \qquad (8)$$

The free particle propagator G^0 is simply a product of Gaussian paths. According to this formula, the full Green's function for all A particles is given by a product of pair propagators divided by their free-particle equivalents. The simplest approximation to this ratio is

$$g_{ij}/g^0_{ij} = \exp\{-(\Delta\tau/2)[V_{ij}(r) + V_{ij}(r')]\} \quad . \qquad (9)$$

The potential and, as a consequence, g are operators in spin-isospin space.

A better approximation to the short-time propagator is obtained by sampling so-called "internal" paths (Figure 2). Given an initial point $\mathbf{R}$ and a final point $\mathbf{R}'$ in the $3A$-coordinate space, we sample a set of Gaussian paths between the two points. Equation 9 can be summed over these internal paths to improve the approximate Green's function. This improvement is not computationally expensive because we treat only two nucleons at a time.

Typically, 10 to 20 runs are required for a given interaction model, each run containing about 1000 copies of the system. Since the energy scales in the interaction range up to hundreds of MeV, the timestep $\Delta\tau$ is taken to be about 3×10^{-4} MeV^{-1}. The subdivisions into internal paths yield an effective timestep of less than 10^{-4} MeV^{-1} for the pair propagators g_{ij} (Equation 9). Each copy of the system must be iterated through hundreds of timesteps to ensure both convergence to the ground state and sufficient statistical accuracy.

The $L \cdot S$ operators involve a derivative operator and hence cannot be treated by the purely static means described above. Instead, we evaluate the $L \cdot S$ term acting on the free-particle Green's function. This allows one to obtain an expression for the full Green's function valid to first order in the timestep. The same method can be used for the two-pion-exchange, three-nucleon interaction, which, although momentum-independent, has a rather complicated spin-isospin structure. The resulting full propagator is

$$G_{\Delta\tau}(\mathbf{R},\mathbf{R}') = [1 - \sum_{i<j<k} \Delta\tau V_{ijk}] \times [1 - \sum_{i<j} \Delta\tau V_{LS} L \cdot S_{ij}] G^0(\mathbf{R},\mathbf{R}') \prod_{i<j} g_{ij}/g_{ij}^0 \quad . (10)$$

Results

Both VMC and GFMC results are summarized in Table 1. The table includes the total ground-state energy E, its various two- and three-body contributions, point nucleon root mean square (rms) radii, and the D-state probability, which provides a measure of the importance of the tensor force. The Argonne V8 (AV8) interaction model used here is designed to mimic as closely as possible the Argonne V14 (AV14) model[7] and is somewhat different from that used previously.[5] Also, the three-nucleon interaction has been refit to produce the correct $A = 3$ binding energy with the AV14 interaction. In this model of TNI, which is significantly less attractive than the previous Urbana 7 model,[1] the attractive two-pion-exchange piece has been weakened, and the strength of the repulsive term has been increased.

Calculations using two-nucleon interactions alone underbind the alpha particle by roughly 4 MeV out of the experimental energy of 28.3 MeV. Variational methods underestimate the true binding energy by 1 to 2 MeV in the alpha particle, so GFMC methods are necessary to perform precise tests of the three-nucleon interaction. The underestimates obtained in variational calculations are a significant fraction of typical TNI effects.

The variational wave function for the AV8 + TNI interaction was that which optimized the AV14 + TNI model 7 interaction. It is not the optimum variational wave function for the energy, but it has the correct experimental asymptotic separation energy and will do a better job in predicting many observables. Ground-state expectation values other than the energy

	Reid V8 VMC[a]	Reid V8 GFMC[b]	AV8 VMC	AV8 GFMC	AV8+TNI VMC	AV8+TNI GFMC
E (MeV)	−23.1(0.1)[c]	−24.6(0.2)	−23.7(0.2)	−24.9(0.2)	−25.8(0.2)	−29.2(0.2)
KE (MeV)	101.6(0.7)	109.2(2.0)	88.5(1.0)	94.4(1.5)	108.0(0.8)	109.8(1.5)
V_{NN} (MeV)	−125.4(0.8)	−135.0(2.2)	−112.9(0.9)	−121.2(1.5)	−131.7(0.8)	−137.4(1.5)
V_{Coul}(MeV)	0.71(0.01)	0.71(0.02)	0.71(0.01)	0.72(0.01)	0.75(0.01)	0.75(0.01)
TNI_S (MeV)					5.8(0.1)	4.8(0.2)
$TNI_{2\pi}$ (MeV)					−8.7(0.1)	−10.6(0.3)
$\langle r_i^2 \rangle^{1/2}$ (fm)	1.58(0.01)	1.53(0.01)	1.61(0.02)	1.54(0.04)	1.46(0.01)	1.44(0.02)
D (%)	14.8(0.1)	15.5(0.2)	13.7(0.1)	14.1(0.1)	17.4(0.1)	16.4(0.4)

[a] Variational Monte Carlo.
[b] Green's function Monte Carlo.
[c] Statistical errors are indicated in parentheses.

Table 1. Comparison of alpha-particle results, using VMC and GFMC methods.

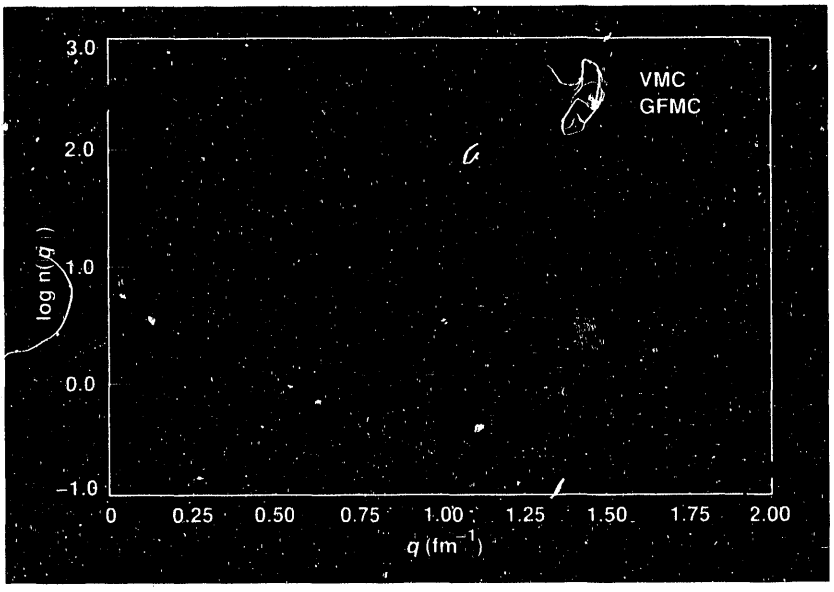


Figure 3. VMC and GFMC calculations of the momentum distribution in the alpha particle. This calculation used a previous version of the AV8 model and included no TNI. There is an additional high-momentum tail to the distribution because of the strong core in the nucleon-nucleon interaction.

are extrapolated from variational and "mixed" estimates:

$$\langle\Psi_0|O|\Psi_0\rangle \approx 2\langle\Psi_T|O|\Psi_0\rangle - \langle\Psi_T|O|\Psi_T\rangle \quad . \quad (11)$$

With GFMC we solve for Ψ rather than Ψ^2, since the analogy between the Schrödinger equation and a diffusion equation is only valid for the wave function itself. Consequently, we must extrapolate from the trial to the true ground-state expectation value, a second-order extrapolation in the error of the original trial function. More accurate calculations of expectation values are possible but quite elaborate. No extrapolation is necessary for the energy, of course, because the true wave function is an eigenstate of the Hamiltonian.

We can calculate the momentum distribution of the nucleons using GFMC as well (Figure 3). Although not directly accessible experimentally, the momentum distribution has significant effects on quasi-elastic electron scattering. The variational wave function provides an adequate estimate of the momentum distribution, at least up to $\approx$1.5 fm^{-1}. Only the variational and mixed GFMC estimates are plotted here. Since the statistical errors are similar to the differences between the two curves, we have not plotted the extrapolated result (Equation 11). There are significant differences in the distributions, though, since the kinetic energy (the second moment of the momentum distribution) in the two cases differs by 10%.

The charge form factor and the Coulomb sum rule of light nuclei are also important experimental quantities. The charge form factor gives information about the distribution of charge in the nucleus as a function of the distance from the center of mass. The simplest approximation is the so-called "impulse" (one-body) term, where the charge and current operators are assumed to be given by the sum of individual nucleon contributions. However, this approximation breaks down rapidly as a function of q^2, which is the momentum transfer in elastic electron scattering. There are important two-body charges and currents in nuclear physics arising from the exchange of charged mesons. Previous variational calculations of $A = 3$ and $A = 4$ nuclei successfully describe the form factors of ^{3}He and ^{3}H by incorporating models of the two-body currents. However, these same calculations underestimate the form factor of the alpha particle in the region of the second maximum.[8]

To date, we have included only the one-body currents in the GFMC calculations. As is apparent from Figure 4, which compares the VMC and GFMC calculations, there is a significant difference in the region of the second maximum. This suggests that the disagreement of previous exchange-current calculations may be attributable at least in part to inaccuracies in the variational wave function. At present we are incorporating the exchange-current models into the GFMC calculation so that a direct comparison can be made with experimental results.

Another interesting quantity is the proton-proton pair distribution, the probability for two protons to be separated by a distance r. Experimentally, the Fourier transform of this quantity, $\rho_{pp}(q)$, can be extracted from the Coulomb sum rule. The sum rule can be obtained by integrating the results of many electron-scattering experiments and is characterized by a diffraction minimum due to the strong repulsive core in the two-body force. Recently, $\rho_{pp}(q)$ has been extracted from experimental results in $A = 3$ (see Reference 9). These results appear to indicate an even stronger repulsive core, as evidenced by a

larger peak in correlation near the second maximum. As shown in Figure 5, our results indicate that VMC calculations of ρ_{pp} are fairly accurate, producing results similar to GFMC calculations. Exchange-current effects are expected to be small here but should be included in the calculations before one draws any strong conclusions.

Conclusion and Future Directions

These calculations are the first GFMC calculations of the alpha particle which include the nonperturbative effects of the three-nucleon interaction. VMC calculations provide a good overall description of the ground state, but GFMC results indicate that there are significant disagreements in some expectation values. GFMC is important because of its ability to provide tests of the nuclear interaction models, especially the importance of the three-body force.

In the immediate future, we will incorporate exchange-current effects in the GFMC calculations. These currents are very important in many areas of nuclear physics and are best probed using a wide variety of experimental information. In addition, it is important that this research be extended to heavier nuclei. By studying systems of A = 6–8, we can gain valuable information on the isospin dependence of the nuclear interaction. Determining the interaction in this region would strongly affect the calculated properties of neutron stars. Another longer-term goal is to be able to calculate the dynamic properties of few-nucleon systems. Toward this end, we have developed first-principles methods for treating low-energy regimes, and we have obtained approximate methods in the higher-energy region. Progress in each of these areas—that is, in the study of heavier nuclei, in incorporating exchange-current effects, and in the dynamic response of nuclei—is vital to our understanding of present and future experimental results. ■

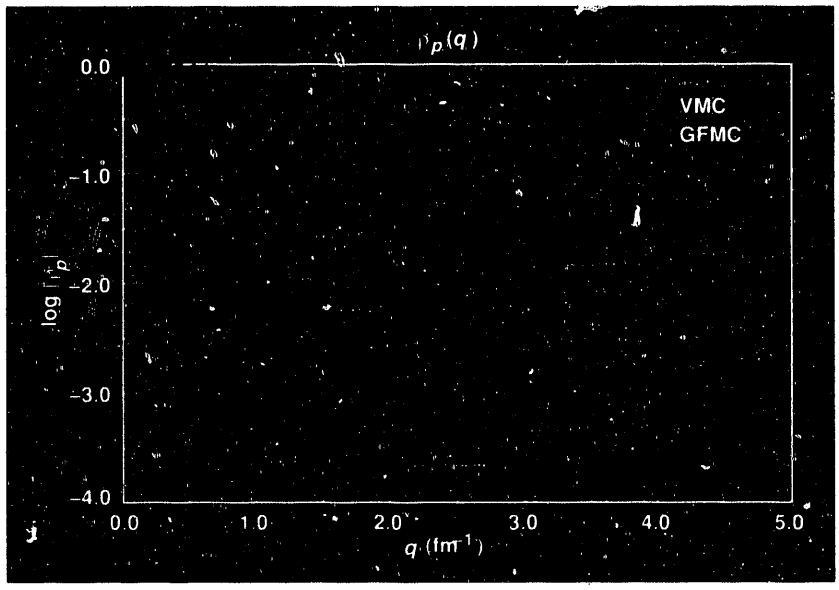


Figure 4. A comparison of VMC and GFMC calculations of the Fourier transform of the one-body proton density, $\rho_p(q)$.

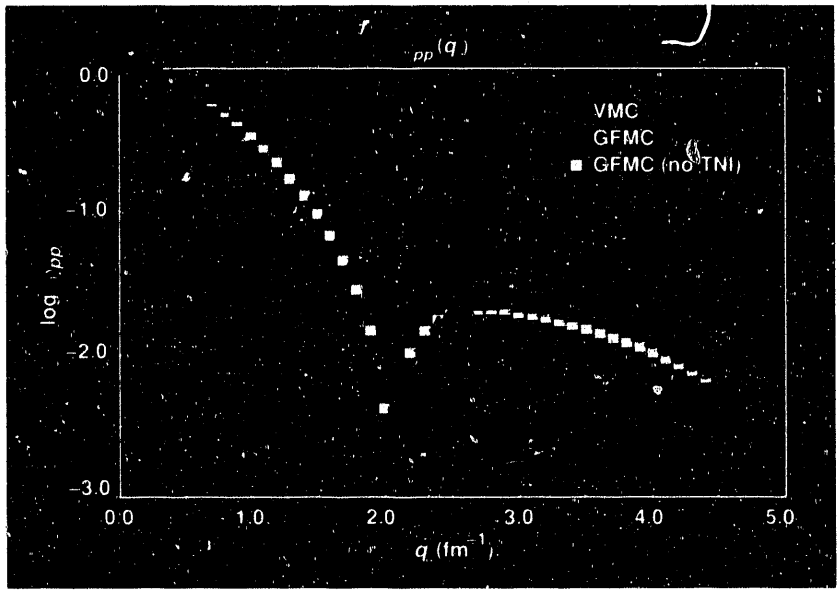


Figure 5. The Fourier transform of the proton-proton distribution function, $\rho_{pp}(q)$, with the AV8 interaction. Statistical errors are the same size or smaller than the symbols, except very near the diffraction minimum.

Acknowledgment

This work was supported by the U. S. Department of Energy.

References

1. J. Carlson, V.R. Pandharipande, and R.B. Wiringa, *Nucl. Phys.* **A401**, 59 (1983).
2. R. Schiavilla, V.R. Pandharipande, and R.B. Wiringa, *Nucl. Phys.* **A449**, 219 (1986).
3. M.H. Kalos, *Phys. Rev.* **128**, 1791 (1962).
4. J. Carlson, *Phys. Rev.* **C36**, 2026 (1987) and **C38**, 1879 (1988).
5. J. Carlson, *Nucl. Phys.* **A508**, 141-C (1990).
6. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, *J. Chem. Phys.* **21**, 1087 (1953).
7. R.B. Wiringa, R.A. Smith, and T.L. Ainsworth, *Phys. Rev.* **C29**, 1207 (1984).
8. R. Schiavilla, V.R. Pandharipande, and D.O. Riska, *Phys. Rev.* **C40**, 2294 (1989) and CEBAF preprint (1989).
9. D.H. Beck, *Phys. Rev. Lett.* **64**, 268 (1990).

Joseph A. Carlson is a staff scientist in the medium-energy nuclear theory group of the Theoretical Division at Los Alamos National Laboratory. Earlier, he was a J. Robert Oppenheimer Fellow at Los Alamos and a postdoctoral fellow at New York University with Dr. M.H. Kalos. His graduate work was done with Dr. V.R. Pandharipande at the University of Illinois at Urbana-Champaign. Dr. Carlson's research interests include the study of strongly interacting quantum systems in nuclear and condensed-matter physics.

Parallel Processing

Bruce Curtis, Lawrence Livermore National Laboratory

Parallel processing—running a code simultaneously on multiple CPUs—makes more efficient use of supercomputer resources.

Parallel processing is the technique of decomposing a computational task into a set of subtasks and then performing the subtasks simultaneously. The other side of the coin, where subtasks are performed sequentially, is called serial processing. The National Energy Research Supercomputer Center (NERSC) is committed to providing parallel processing capabilities on its supercomputers.

At the time of this writing, NERSC has four supercomputers available to its users, and each of these supercomputers has more than one Central Processing Unit (CPU), as shown in Table 1. On each machine, the multiple CPUs operate concurrently and share a common memory, so it is possible for them to collaborate in executing a single program. To do so, however, the program must be broken into pieces that can be executed in parallel. To run concurrently, the pieces must have some degree of independence. When a dependence exists among the pieces, the CPUs must synchronize their efforts so that the dependence is satisfied.

Three basic steps from Gaussian elimination illustrate how this works:

(1) Find the best pivot element.

(2) Interchange rows to position the pivot element.

(3) Eliminate the *k*th unknown.

Each of the three steps can be broken up into smaller pieces that can be executed simultaneously by dividing the data among the CPUs. For example, CPU "A" can examine the first 128 elements for a good pivot element, CPU "B" can take the next 128, and so on. However, among the pieces there are order dependences that must be maintained. In step 1, each CPU produces a candidate pivot element. All of the candidates must be examined to determine which is best. Therefore, all pieces of step 1 have to be completed; then one CPU picks the best candidate pivot element, while the other CPUs wait. This stage is known as a serial region. Furthermore, the serial region from step 1 has to be completed before any of the pieces from step 2 can start. Similarly, all pieces from step 2 must be completed before any of the pieces from step 3 can begin. If this order of executing the subtasks is not strictly followed, the results will be incorrect.

The Advantages of Parallel Processing

Why apply multiple CPUs to a single job? Since we can use a multi-CPU machine by scheduling independent jobs on the different CPUs, why go to all this trouble? In fact, for some situations, independent job streams are the best approach. But there are some conditions for which parallel processing is beneficial:

- *Maximum-memory jobs.* If the entire memory is taken up by a small number of jobs, then one or more of the CPUs may be idle even though there might be plenty of other jobs to run.
- *Dedicated machine.* If the computer is devoted to running a single job, then all but one CPU become idle.
- *Light workload.* If the number of jobs waiting for a CPU ever falls below the total number of CPUs, then one or more of the CPUs becomes idle.

Parallel processing eliminates the idle time. If the workload is divided among all the

Machine	Number of CPUs	Memory size (million words)
Cray X-MP	2	2
Cray-2	4	67
Cray-2	4	134
Cray-2	8	134

Table 1. Supercomputers at NERSC.

CPUs, the amount of work done per unit time (throughput) increases. The additional CPUs act as accelerators instead of sitting idle.

Even if there is practically no idle time, the fact that additional CPUs act as accelerators can lead to still more benefits. From the operating system's point of view, a computer program is a consumer of such resources as CPU cycles, memory words, and input/output (I/O) channels. By accelerating a code's execution, the system can satisfy the code's computing requirements with fewer resources. For example, a code that requires 40 minutes of CPU time and 40 megawords of memory consumes 1600 megaword-minutes of memory resource when run using only one CPU. If we can apply four CPUs, the 40 minutes of CPU time can be divided into four concurrent 10-minute chunks. Thus the code needs to be in memory only 10 minutes and consumes only 400 megaword-minutes of memory resource, provided it has perfect parallelism. Furthermore, in a timesharing system such as NERSC's Cray Time-Sharing System (CTSS), codes are "swapped" in and out of memory to serve more codes than can fit simultaneously in memory. A parallel code needs to be in memory a shorter time and thus needs to be swapped less often, thereby consuming less I/O resource.

From the user's point of view, the benefits of parallel processing are twofold. Faster turnaround on parallel jobs is one benefit, because a parallel code will execute in less real time, although how much less is hard to predict in a timesharing system. Reduced resource consumption provides the other benefit in the form of reduced cost. At NERSC, the amount a user is charged for running a code is based on the amount of real resources used. In the above example, 1600 megaword-minutes of memory charge would be accumulated by the serial approach and 400 megaword-minutes by four-way-parallel processing.

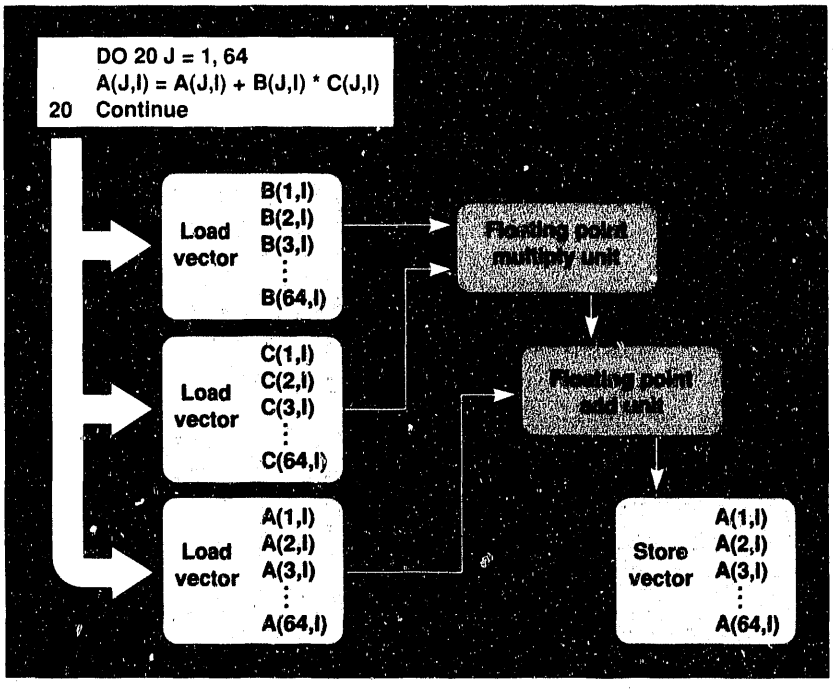


Figure 1. Low-level parallelism. The inner DO loop is vectorized. Each of the blocks corresponds to a single instruction on Cray computers. Parallelism is realized in two ways. The first is within a functional unit, like the floating point multiply unit. Portions of the 64 multiply operations B(1,I)*C(1,I), B(2,I)*C(2,I),...,B(64,I)*C(64,I) overlap in time. The second form of parallelism is between functional units. The memory unit can load vector A(1,I),...,A(64,I) while the multiply unit is executing its 64 operations.

Parallel Computer Architectures

Parallel processing is a means of getting the effect of a faster single CPU, but there are many different computer architectures to achieve that effect. The Cray computers used at NERSC have a small number of very powerful processors that share a large memory. Other computers—like the Sequent, BBN Butterfly, and Encore Multimax systems—have a larger number of less-powerful CPUs that share memory.

Still other computers—like the Intel and NCUBE hypercubes—have up to thousands of processors that do not share a memory; rather, each processor has its own memory. This is called a distributed memory. In computers with distributed memories, the processors cooperate and communicate by sending messages to each other over a switching network. There are many choices for the layout of the communication network, leading to diversity among distributed-memory systems.

All of the systems mentioned above belong to a class of computers known as multiple-instruction-stream/multiple-data-stream (MIMD) machines. This class includes systems with an array of associated processors, each executing its own independent instruction sequence. In another class of computers, known as single-instruction-stream/multiple-data-stream (SIMD)

machines, the processors always execute the same instruction at the same time, but on different data. The Thinking Machines Corp.'s Connection Machine (with up to 65,536 processors) and the ICL DAP (4,096 processors) are examples of the SIMD class. SIMD machines usually have a distributed memory.

Data-flow machines do not organize their instructions into a stream. Instead, instructions are executed when, and only when, all of their operands have been produced by previous instructions. In this asynchronous architecture, a large amount of instruction-level parallelism can be exploited. The Massachusetts Institute of Technology's tagged token computer is an example.

The number of CPUs in new computers is now rising because the cost of CPUs is dropping relative to the cost of other components, such as memory and secondary storage. Computers with hundreds to thousands of processors are known as massively parallel systems. Much research is under way to develop efficient interconnection strategies for massively parallel systems.

Multiprocessor systems have the advantage of higher availability. If one CPU has a hardware failure, it can be disabled, allowing the rest of the machine to function normally (with proportional performance degradation). A single-processor machine is useless when its CPU is down. The goal of limiting the impact of a component's failure is known as fault tolerance. Tandem's multiprocessors, for example, are designed with an emphasis on fault tolerance.

For any of these widely varying parallel computer architectures, the total aggregate computing power depends on more than just the speed and number of CPUs. Each of the architectures has unique strengths and weaknesses that affect total performance. For example, on distributed-memory machines the cost of communication between processors can be high. Consequently, to get good performance, the characteristics of the code—such as the amount and level of parallelism and the type of operations performed most frequently—must be a good match for the machine's architecture.

Levels of Parallelism

Computer codes contain several levels of parallelism, as shown in Figures 1, 2, and 3. The lowest levels of parallelism (Figure 1) are those between the instructions within a

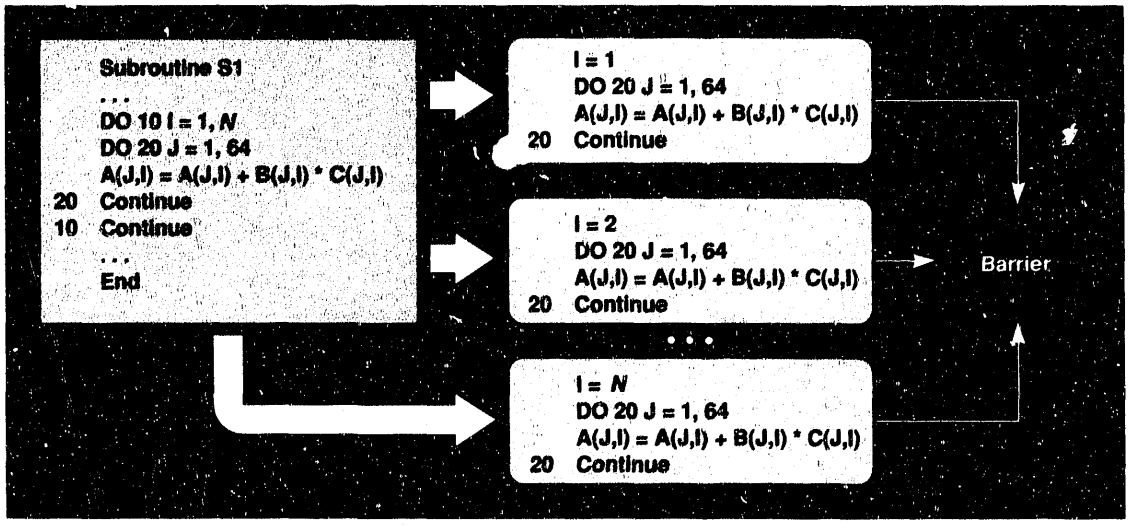


Figure 2. Intermediate-level parallelism. A portion of subroutine S1 consisting of a doubly nested DO loop is decomposed into *N* independent tasks. The decomposition is done by splitting the data on the second dimension of the two-dimensional arrays A, B, and C so that each task does one iteration of the outer loop. Then the decomposition must be mapped onto a machine. If the machine has *P* CPUs, and *P* is less than *N*, each processor might be assigned *N/P* tasks. Synchronization consists of a barrier that prevents processors from continuing execution beyond the region in question until all the tasks have been completed.

CPU and pipelining (vector processing). Compiler technology has improved to the point where the compiler automatically detects and exploits these forms of parallelism, so the research scientist using NERSC's supercomputers need not be concerned with the difficult problems relating to this type of parallelism.

The intermediate levels of parallelism (Figure 2), those between the elements of a program within a module, are our present target. The most important form of intermediate parallelism is the "DO" loop, because a high percentage of a code's execution time is spent inside DO loops. Furthermore, it is apparent that much parallelism exists in DO loops. Our goal is to make detecting and exploiting the intermediate levels of parallelism as automatic as for the lower levels.

DO loops offer a wealth of parallelism. In scientific codes, DO loops are often used to perform operations on a set of arrays. The arrays can be split into smaller sections to which each processor applies the identical set of operations. The splitting may give each CPU contiguous array elements; it may cause successive iterations of the DO loop to be performed in different CPUs; or two or more independent DO loops can be spread among the various CPUs. It is common for DO loops to afford both vector and parallel operations. To take best advantage of the inherent parallelism, an automatic system must make difficult decisions regarding decomposition and synchronization for parallelism, without adversely affecting the performance of vector operations.

The highest levels of parallelism (Figure 3), those between portions of a program not within a module, are important because they are most appropriate for the computer architectures we have at NERSC. Unfortunately,

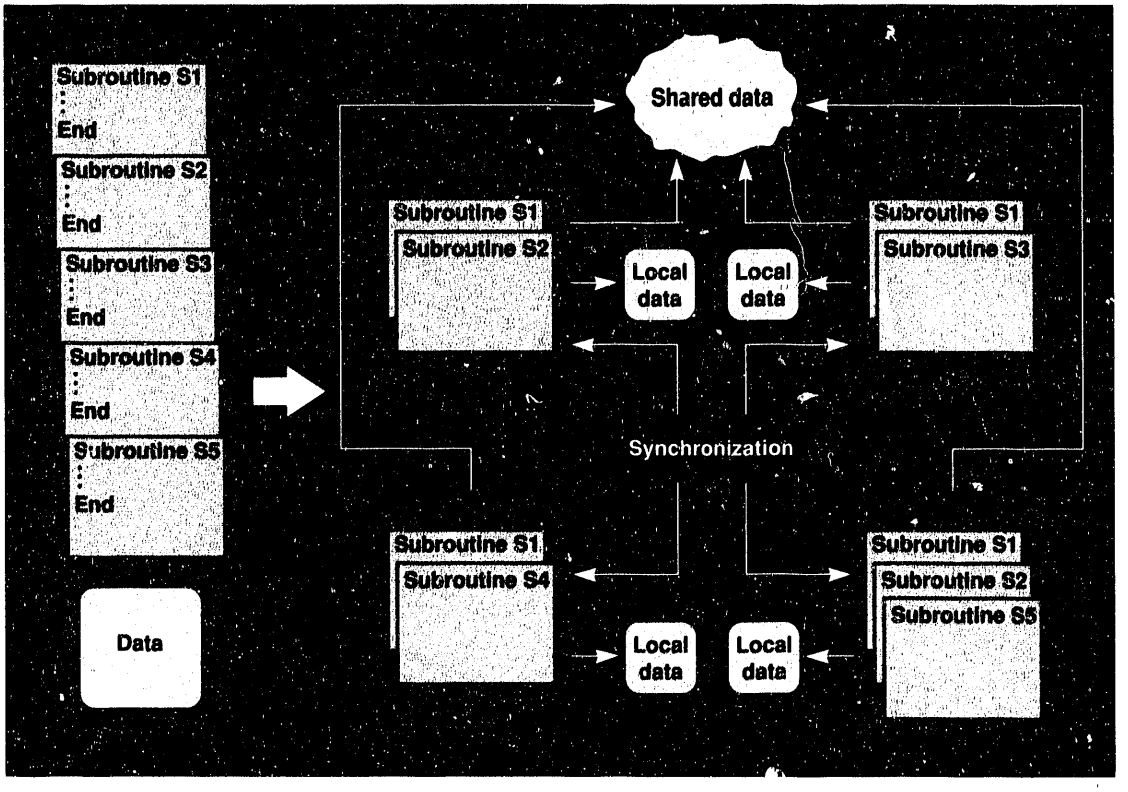


Figure 3. High-level parallelism. A program consisting of five subroutines and some data is decomposed into four concurrent tasks, data that are private to each task, data that are shared among the tasks, and synchronization procedures to control the tasks. Subroutine S1 appears in all tasks; the data that S1 operates on are split between the tasks. Similarly, S2 appears in two tasks, each taking half the data. Subroutines S3, S4, and S5 are functionally disjointed and execute in separate tasks without the need for their datasets to be divided. The synchronization procedures enforce any dependences. For example, if S2 produces data that S4 uses, S2 must complete execution before S4 starts.

high-level parallelism is extremely difficult to reach by automatic exploitation. This is because the detection phase—that is, determining whether portions of a code can be run concurrently in a safe manner—becomes much harder when module boundaries are crossed. We provide a programming environment through which users can express the higher-level parallelism. The user must establish how the code is to be broken into subtasks and must explicitly program the synchronization between the subtasks.

It is evident from Amdahl's Law (Figure 4) that maximum performance derives from maximizing the total amount of parallelism in a code. Therefore our strategy at NERSC is to support all levels of parallelism in combination.

NERSC's Parallel Programming Support

NERSC offers state-of-the-art tools to its user community. For example, the Cray Research, Inc. autotasking system has been available to users since September 1989. This system includes:

(1) A source code analyzer that discovers parallelism, decides how the decomposition is to be done, and restructures the code to enhance vectorization.

(2) A translator that performs the decomposition which the analyzer has specified and inserts the required synchronization.

(3) An optimizing, vectorizing compiler that produces an executable program.

In addition, we are developing parallel processing capability in our own compiler (called CIVIC). This compiler will combine the technologies produced by top researchers throughout the nation and will perform all three of the above functions.

A comprehensive parallel programming environment is needed to make the most effective use of multiprocessors, complete with the tools needed to enhance the design, development, portability, debugging, and analysis of parallel codes. One tool within our environment at NERSC is MOJO, a graphical post-processor that displays visually the execution of a parallel code. With MOJO, scientists can discover inefficiencies in a parallel code, find certain bugs, and measure the code's performance. Figure 5 shows sample MOJO screens. Another tool we have is FORGE, a product of Pacific Sierra Research, Inc., with capabilities that include timing analysis and interactive parallelization. FORGE is in the early stages of being able to detect high-level parallelism. Debugging is a particularly critical aspect of the multiprocessing environment, and we are collaborating with Cray Computer Corporation to substantially improve debugging capabilities.

In addition to tools, a complete parallel computing environment needs programming language constructs to augment the

f_p (%)	N=2	N=4	N=8	N=16	N=32	N=64	N=128	N=256
0.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10.0	1.05	1.08	1.10	1.10	1.11	1.11	1.11	1.11
20.0	1.11	1.18	1.21	1.23	1.24	1.25	1.25	1.25
30.0	1.18	1.29	1.36	1.39	1.41	1.42	1.42	1.43
40.0	1.25	1.43	1.54	1.60	1.63	1.65	1.66	1.66
50.0	1.33	1.60	1.78	1.88	1.94	1.97	1.98	1.99
60.0	1.43	1.82	2.11	2.29	2.39	2.44	2.47	2.49
70.0	1.54	2.11	2.58	2.91	3.11	3.22	3.27	3.30
80.0	1.67	2.50	3.33	4.00	4.44	4.71	4.85	4.92
90.0	1.82	3.08	4.71	6.40	7.80	8.77	9.34	9.66
91.0	1.83	3.15	4.91	6.81	8.44	9.60	10.30	10.69
92.0	1.85	3.23	5.13	7.27	9.20	10.60	11.47	11.96
93.0	1.87	3.31	5.37	7.80	10.09	11.83	12.94	13.58
94.0	1.89	3.39	5.63	8.42	11.19	13.39	14.85	15.71
95.0	1.90	3.48	5.93	9.14	12.55	15.42	17.41	18.62
96.0	1.92	3.57	6.25	10.00	14.29	18.18	21.05	22.86
97.0	1.94	3.67	6.61	11.03	16.58	22.15	26.61	29.60
98.0	1.96	3.77	7.02	12.31	19.75	28.32	36.16	41.97
99.0	1.98	3.88	7.48	13.91	24.43	39.26	56.39	72.11
99.2	1.98	3.91	7.58	14.29	25.64	42.55	63.49	84.21
99.4	1.99	3.93	7.68	14.68	26.98	46.44	72.64	101.19
99.6	1.99	3.95	7.78	15.09	28.47	51.12	84.88	126.73
99.8	2.00	3.98	7.89	15.53	30.13	56.84	102.07	169.54
99.9	2.00	3.99	7.94	15.76	31.04	60.21	113.58	203.98
100.0	2.00	4.00	8.00	16.00	32.00	64.00	128.00	256.00

Figure 4. Amdahl's law says that the slowest component of a program dominates the program's performance. With respect to parallel processing, Amdahl's law can be stated as $S_N = 1/(f_s + f_p/N)$, where S_N is the maximum speedup obtained by running on N processors, f_s is the fraction of the program that is serial, and f_p is the fraction of the program that is parallel. The slowest component is the serial part, so it is important to make as much of the program parallel as possible. This chart gives the speedup (S_N) for different values of f_p and N. For example, a program that is 90% parallel, when run on a 16-processor machine, has a maximum theoretical speedup of only 6.4. (Source: Cray Research, Inc.)

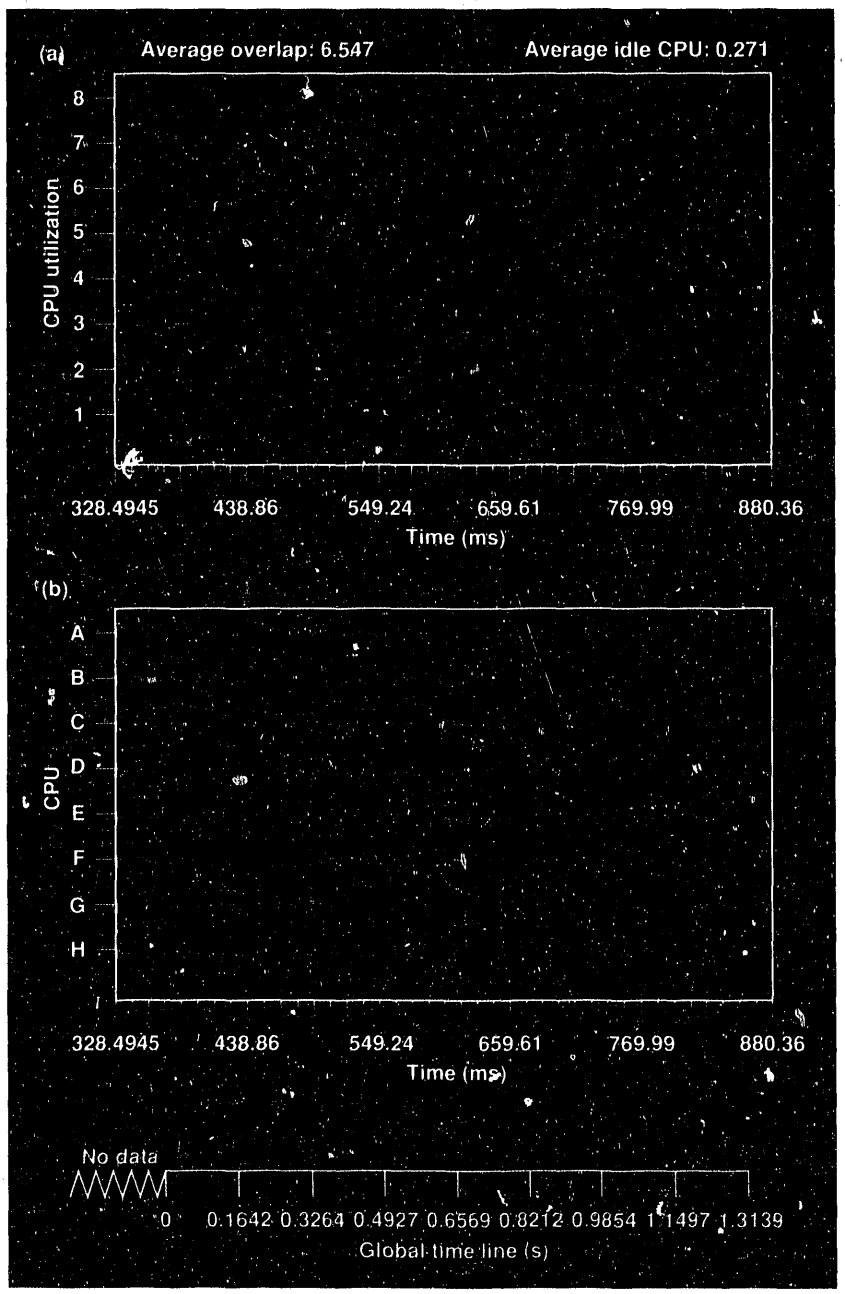


Figure 5. Sample MOJO screens. MOJO displays information about a parallel code. Data collected while the parallel code runs are analyzed by MOJO after the code completes and are presented graphically. MOJO allows the user to display any portion of the execution time. The arrows on the global time line show which portion is being displayed. Here, both screens are displaying the same portion of the same code, an interval of about half a second. The top screen (a) is the concurrency graph, showing the number of CPUs the code had versus time. The red areas are where CPUs had no work to do. This code had somewhat regular serial regions, seen as downward spikes of the red idle time. The bottom screen (b) shows the start and end of each time slice and can help in interpreting the concurrency graph. For example, at about the 600-ms mark, the concurrency graph shows a drop to four processors for a short time. The time-slice diagram indicates the reason for the drop: four time slices (CPUs A, F, G and H) ended at about the 600-ms mark, and new slices started about 50 ms later.

automatic detection of parallelism. At NERSC, we are contributing to national efforts to specify standard programming language features to support parallel processing. A parallel computing environment also requires run-time library support. Some of the more time-consuming mathematical library routines, such as matrix multiplication, will be parallelized. For parallel codes that use these routines, the total amount of parallelism will be increased without any effort on the user's part.

CPU Scheduling

It is the responsibility of the operating system to assign processors to jobs. In a time-sharing system such as CTSS, all jobs are given processors for many short intervals of time, called "time slices," rather than for the entire duration of the job. This sharing of CPU time is to satisfy the computing requirements of jobs with opposing needs. Interactive jobs like text editing need very little CPU time, but they need it right away, or else the interactive response time will be unacceptably long. On the other hand, big number-crunching jobs need a lot of CPU time, although they do not necessarily need this time immediately, just soon enough for acceptable turnaround. Improving the service to one type of job tends to degrade service to the opposite type. The problem of assigning processors to jobs—while meeting constraints such as response time, turnaround time, and system throughput—is known as CPU scheduling.

For a parallel job, the operating system must assign processors so that the time slices overlap to a large extent. The parallel performance of a code is limited by the CPU overlap delivered by the system Parallel jobs would be well served if the system could schedule CPUs so that each job that needs K processors actually receives K processors within a certain (small) amount of time. But that is easier said than done. Each of the various approaches to CPU scheduling has its benefits and drawbacks.

Preemptive scheduling. In this approach, when a processor becomes available to give to a parallel code, the system also preempts enough other CPUs to satisfy the parallel

code's needs. But each code that has been preempted gets degraded performance, since shortening its time slice lengthens its turnaround time. And if a preempted code were itself parallel, it would get reduced overlap as well. The system can try to make up for preempting a serial code by increasing its next time slice, but the same code may get preempted so frequently that turnaround is still impacted. The system can try to avoid preempting other parallel codes, but that is not always possible. It can prevent the reduction in overlap by preempting all of the CPUs that a parallel code has, but that would impact turnaround time as it does for a serial job.

Nonpreemptive scheduling. Here, the system gives processors to a parallel job as they become available. A CPU is available when the time slice of a previous job expires, so the overlap a parallel job realizes depends on the "get time," the average time it takes to get an additional processor, which can vary widely. The system can improve the overlap by assigning longer time slices. If the time slice is long compared to the get time, overlap is good. This is the method currently used by CTSS.

Predictive scheduling. The system examines many jobs, perhaps its entire list of jobs, and adjusts the time-slice length and the order of the list, trying to accommodate parallel codes. It tries to predict a feasible schedule. For instance, if the system sees a code that needs three CPUs, it chooses the three CPUs ahead of time and assigns time slices to the codes ahead of the job in question so that the three CPUs are available at the same time. The cost of producing such a feasible schedule can be prohibitive, however, and also the feasible schedule can often be foiled by codes that do not use their entire time slice. Predictive scheduling is normally used as an adjunct of a nonpredictive method. For instance, a simple heuristic scheme can sometimes improve the overlap in nonpreemptive scheduling.

CPU scheduling is further complicated by the dynamic nature of parallel processing. Parallel codes often vary the number of CPUs needed, and many have serial regions. Another problem arises when the system must respond to some external event, like the completion of disk I/O. In such a case, the system briefly takes a CPU away from the code, deals with the external event, and returns the CPU to the code. This is called an interrupt. Interrupts affect the performance of parallel codes because the other CPUs may have to wait at a synchronization point for the return of the interrupted processor.

The parallel-processing support library must cooperate with the operating system. The library informs the system of the job's changing requirements, such as number of CPUs and memory size, and the system informs the library when each requirement is fulfilled. The system's CPU scheduling strategy particularly influences how the library works. As a parallel code changes its short-term processor needs, the library may need to make longer-term decisions, based on the scheduling algorithm. Serial regions are a simple example. When a program executes a serial portion of a code, the library must decide whether to return any extra CPUs to the system for reassigning to another code or to hold onto them in anticipation of more parallel processing. The cost of idling the extra processors, the cost of exchanging the processors between the library and the system, and the scheduling algorithm must all be factored into the decision.

The nonpreemptive scheduling method currently used by CTSS compels the library to perform what are known as context switches. When a processor reaches a synchronization point, the other CPUs may not even be working on the same job. The CPU trying to synchronize may have a long wait until the other CPUs finish their work on other jobs, start new time slices on the job in question, and reach the synchronization point. As illustrated in Figure 6 (a) and (c), this can cause severe performance problems. The library, however, using context switching, can avoid the waiting by assigning one CPU to perform the work previously assigned to other CPUs that don't yet have time slices. In this way, the synchronization point can be reached by all CPUs, even though only one was really doing the work. As a result, the program can perform well with the varying overlap produced by nonpreemptive scheduling, as shown in Figure 6 (b) and (d).

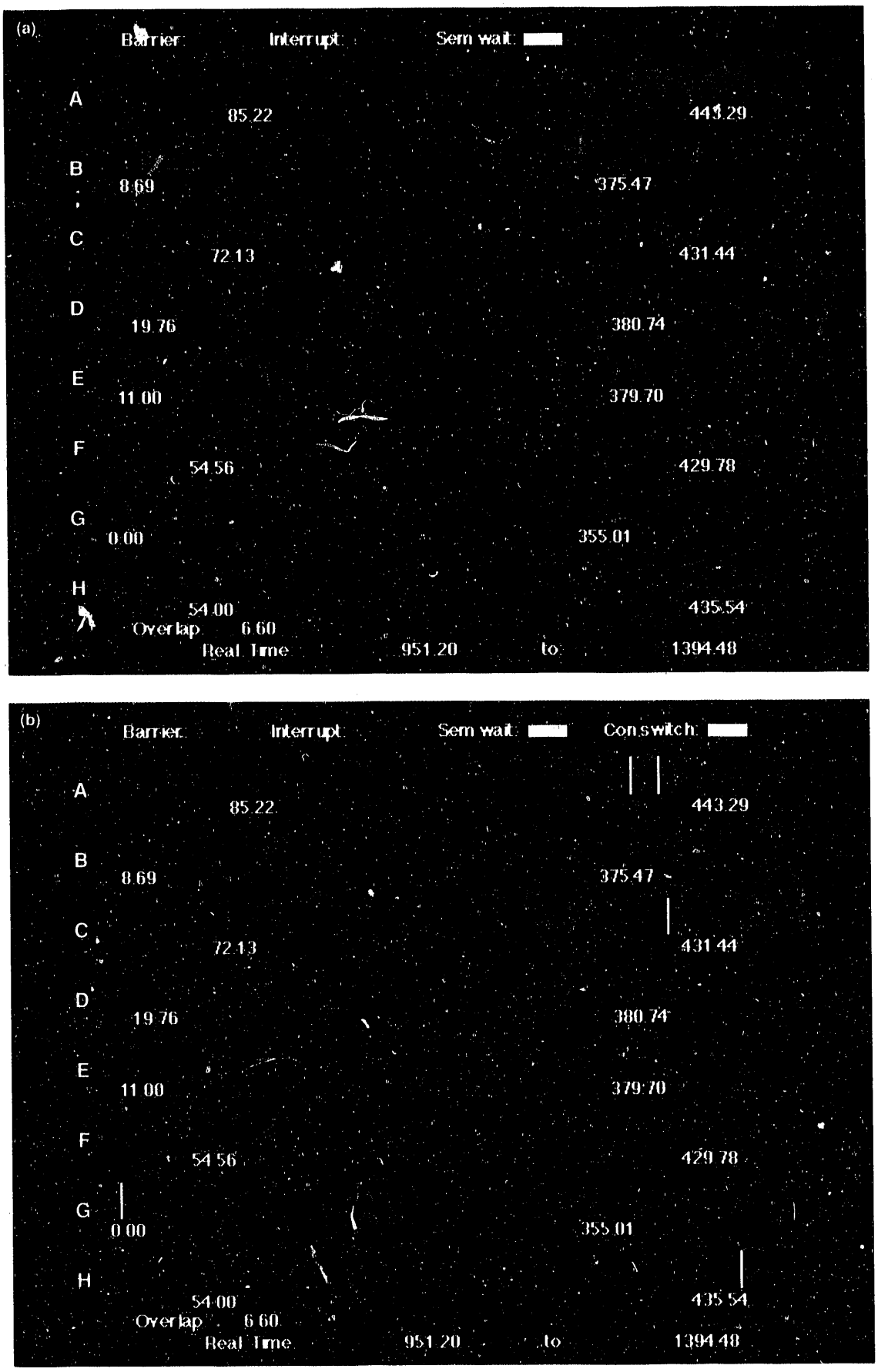
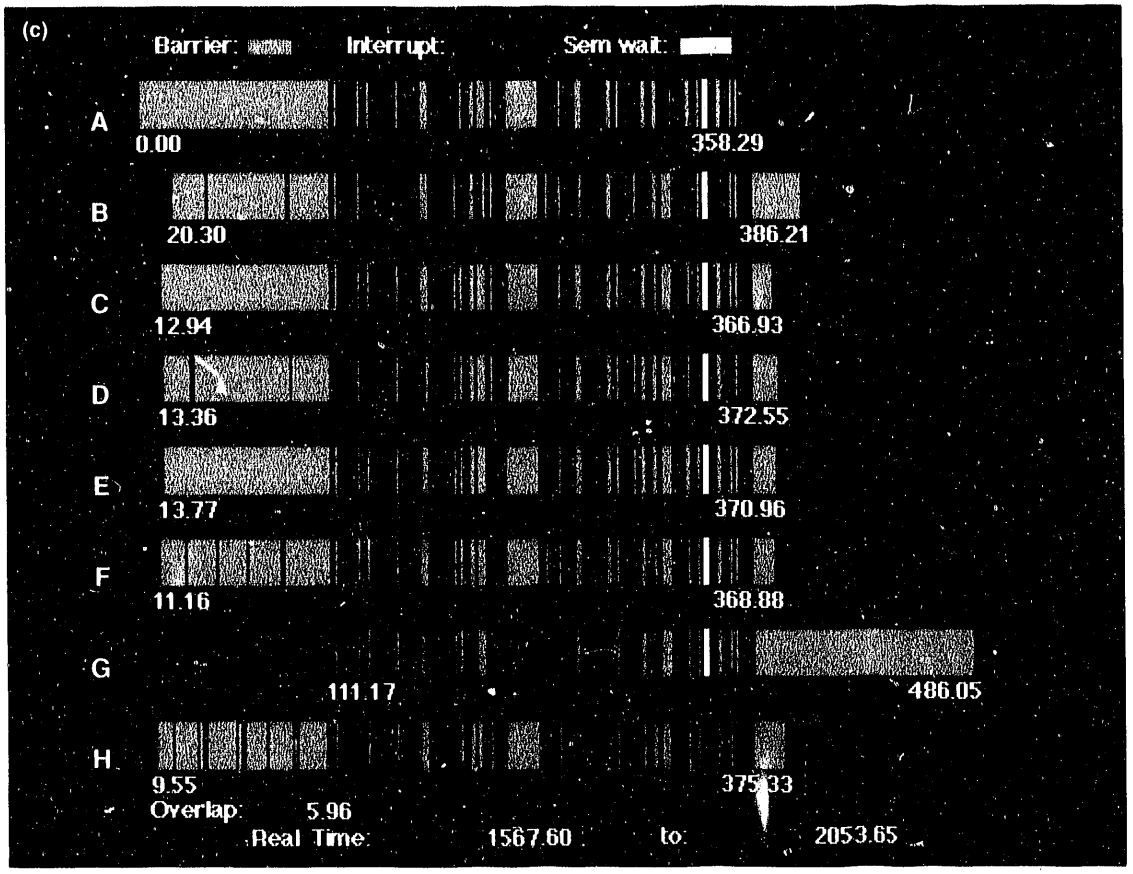


Figure 6. Two examples of nonpreemptive scheduling–(a)/(b) for time 951.20 to 1394.48 and (c)/(d) for time 1567.60 to 2053.65–showing how CPU utilization improves when context switching is applied. In each case, the lower picture shows the use of context switching.

(a) and (c) Nonpreemptive scheduling *without* context switching. The bars represent the time slices of a parallel code on particular CPUs. In (a), at time 0, CPU "G" starts a time slice of length 355.01 ms; at time 8.69 ms, CPU "B" starts a time slice; and so forth. The effect of nonpreemptive scheduling is seen as staggered overlap. The staggering leads to inefficiency because of the amount of time some CPUs must wait at synchronization points for the late-arriving CPUs; these intervals where CPUs must wait are shown in blue. Clearly, the periods at the beginning and end of the time slice, where fewer than eight processors are assigned to the code, are poorly utilized. The effects of interrupts (shown as red areas) can also be seen. If an interrupt is sufficiently long, it causes all of the other CPUs to wait at a synchronization point.

(b) and (d) Nonpreemptive scheduling *with* context switching. Here, the parallel-processing



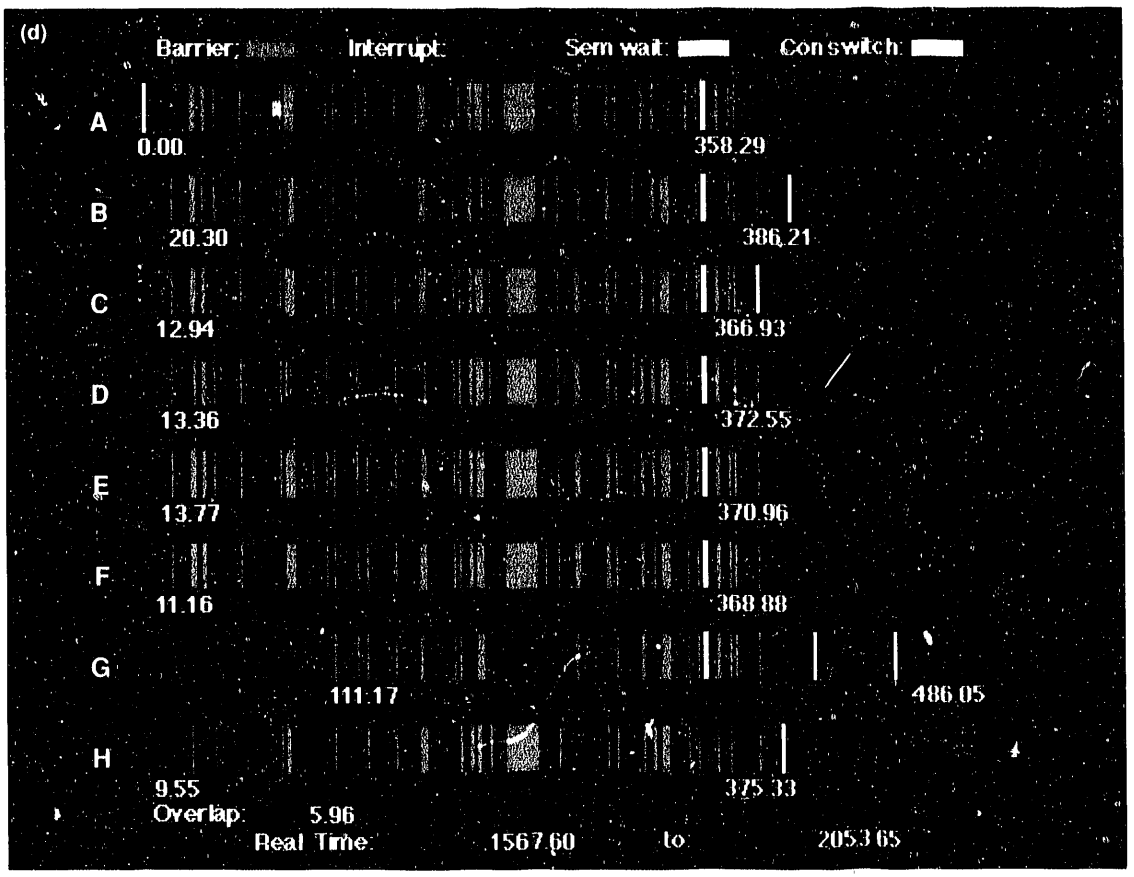


Figure 6, cont. support library performs context switches to solve the performance problem. If, for example, CPU "B" is waiting at a synchronization point and CPU "D" does not yet have a time slice, "B" takes over the context of "D"—that is, it assumes the work assigned to "D." When "D" eventually gets a time slice, the library assigns it new work. Consequently, the time periods where fewer than eight processors have slices are used profitably. (These pictures are the output of MICSIM, a system performance measurement tool, running in this instance on NERSC's eight-processor Cray-2.)

The Future of Parallel Processing

As the need for computational power grows, so does the need for parallel processing. There are limits on the speed of individual hardware components; thus, for any hardware technology, exploiting parallelism is the ultimate way to improve performance. The supercomputers of the future are likely to be massively parallel. And less powerful systems, including personal computers, may also be highly parallel. The diversity in parallel architectures will continue to grow, with new forms appearing and old ones falling into disuse, as though by natural selection.

Methods for programming parallel systems will certainly change as well. Automatic parallelization techniques will continue to improve, but their effectiveness may be limited by the semantics of current programming languages. Furthermore, the paradigms used for programming sequential machines are inappropriate for parallel machines and could also inhibit parallelizers. New paradigms must be developed, and scientists must convert to the new programming strategies.

The inherent difficulties in automatically extracting parallelism from conventional programming languages can be addressed either by language extensions or by developing new languages. In data-flow languages like SISAL, for instance, parallelism does not need to be extracted from sequential statements; rather, parallelism is the norm, and serial operations must be explicitly programmed. Whether or not new languages become popular, there must be advancements in conventional parallel programming to support the enormous volume of software already written.

In a world in which energy research becomes increasingly important, more resources are needed in the search for new energy sources. Parallel computing will continue to assist scientists as the computer models they build require more and more computational power. The evolution of parallel computing is just beginning. ■

Bruce Curtis, now in his 10th year at NERSC, received a B.A. in mathematics from the University of Arizona in 1978 and an M.S. in computer science from Purdue University in 1980. His interests include parallel processing, optimizing compilers, and symbolic computing.

Scientists of the Future: Learning by Doing

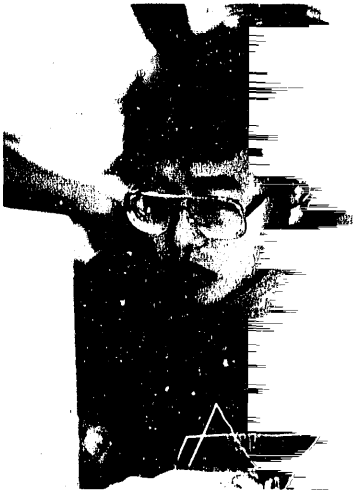
Will there be enough trained scientists, mathematicians, and engineers available in the future to keep the United States in the technological forefront? The National Energy Research Supercomputer Center (NERSC) is tackling this problem by giving high school students hands-on experience with supercomputers through two related programs.

The first program brings top math and science students to Lawrence Livermore National Laboratory for two weeks of work with NERSC's supercomputers. The second program, newly inaugurated, involves training high school teachers in the use of supercomputers so that they, in turn, can introduce scientific supercomputing to their students. To support this project, a one-processor Cray X-MP is being made available, through the cooperation of Cray Research, Inc., as an on-line resource to high school students across the country.

The "Superkids" Program

A select group of students—dubbed "superkids" by the NERSC staff—spent two weeks on site in June 1990. The sixth annual National High School Supercomputing Honors Program drew 58 students—one chosen by the governor of each state, the District of Columbia, and Puerto Rico. (In addition, a few foreign nations are invited to send participants.)

The students learned how supercomputers are used in scientific research, and in workshop settings they performed com-



three-dimensional graphic images. The students also toured research projects at LLNL and had the opportunity to talk at length with laboratory scientists.

Near the session's end, several students summed up their experience.

"I used to think science was a boring field, only for special people," said Nohemi Molina of Mexico. "I thought scientists weren't very interesting and were really in touch with people."

Now she's changed her mind. "I now see science available for anybody to get into. This program has encouraged us to get into science and to make good that field."

Jeb Willenbring of North Dakota was particularly interested in the algorithms used to solve problems in how problems are broken up for computational purposes. "With supercomputers, you can do math problems that you used to think about but couldn't do," said. "For example, the number of calculations it takes to make one graphic image is mind-boggling."

"This experience makes you realize how much there still is to learn," he said. "When you learn something new, that only opens up more things to learn."

Roger Flugel of Connecticut was impressed with seeing the experiments in laser and magnetic fusion and liked talking with the scientists who conduct that research—he thinks he might someday be interested in working in one of those areas.

"This program has opened my eyes to large-scale simulation," he said. "I've seen the value of it and learned how complicated such simulations can be."

The program is one of several high school "honors" programs sponsored by the U.S. Department of Energy (DOE) at national laboratories. NERSC, with the initial session in 1985, was the first DOE facility to offer such a program. The most recent group of students brings to 337 the total number of participants.

Teaching the Teachers

In the second program, which ran concurrently with the first, six high school teachers—from Wisconsin, California, and the District of Columbia—spent three weeks at NERSC in a DOE-funded pilot workshop designed to bring supercomputing directly to their classrooms. The idea is to use supercomputing as both a teaching tool and a catalyst to spark student interest in science and math.

"With this program, we're reaching beyond just a select group of bright students," said John Fitzgerald, assistant director, planning and finance, at NERSC and a key champion of the new program. "Each of these teachers has 150 to 200 students who will now have access to a very attractive resource in the form of a supercomputer. We hope this makes their coursework more interesting and that it will entice more students into science or math."

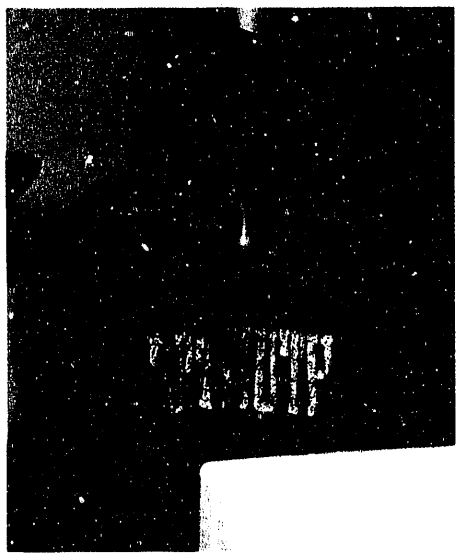
During the session, the teachers spent two weeks learning the basics of supercomputing along with the honors students. In the third week, they concentrated on curriculum development, working on specific ways to incorporate supercomputing into the courses they teach.

At the end of their stay, the teachers were enthusiastic about supercomputing's possibilities and were full of ideas for passing their excitement on to their students.

"I plan to use some of the projects we did on an interactive demonstration basis," said Mark Klawiter, who teaches chemistry and physics in Ladysmith, Wisconsin. "I'll use this class-wide and also with individual students who show a lot of interest in computer programming. I want them to use the supercomputer as a real scientist might and to come up with their own ideas. This will let students experience or 'do' science. They can conjure up their own uses and explore them."

In addition to using the computer as a teaching tool (for example, looking at particle motion), he planned to introduce supercomputing as a discussion topic, emphasizing the interactions of science, technology, and society.

Steve Harmon, who teaches math in Oakland, California, envisioned setting up group projects to illustrate the application of math concepts—perhaps having students use the supercomputer to compile a data base on the environmental history of a lake or pond. "Kids learn by doing science and practicing math," he said, "and there's no better format than a real-world project. That's how real learning takes place. Kids have ideas, in raw form, and that's an important part of science."



"Using the supercomputer will be something that makes a difference," Harmon continued. "You have to get the students excited— you try to get them to do their own science. By bringing specialized information to their level, they can be a part of it."

Tim Emholtz, who teaches math and computer science in New Richmond, Wisconsin, would like to see every one of his students exposed to supercomputing. He expects that levels of capability will vary but hopes that some students will be especially ignited.

"This is the kind of thing that will make better students," he said. "The students are going to be online with supercomputers, which they've never had an opportunity to do before, and that's exciting. It's the difference between hearing about something and doing or seeing it first hand, which is the fun way for kids to learn."

"These machines are capable of a lot of power," he said, "and the kids are capable of just as many amazing things. When you put the two together, it's exciting to think about."

Making Connections

One important goal of the program is to establish a link between the big science that occurs at the national laboratories and the teachers and students of high school science and math—to bring that kind of science directly into the classroom. "We want to provide a network for the teachers and students," said Fitzgerald, "and to develop a sense of community between the scientists here and people in the classroom."

The teachers seemed particularly grateful for that connection.

"Sometimes it feels like we're operating in a vacuum," said Klawiter. "This has helped bridge the gap with scientists. Now I know that if I need to consult, they're just a phone call away and willing to help. That's going to be great."

Emholtz also spoke of bridging the gap. "It's frustrating trying to keep up in computer science, because there are so many changes, so fast," he said. "After teaching 180 to 200 students and prepping for three different classes, I don't have much time left to do research or even reading. I was surprised with the ease with which the scientists here were able to communicate at my level and at their willingness to facilitate my growth. This experience has bridged the gap temporarily and has also provided an avenue for building a long-term bridge."

Expanding the Scope

In a related effort, NERSC's Brian Lindow in July led a two-week supercomputing workshop at Eau Claire, Wisconsin, for another 12 high school teachers.

All of these newly trained teachers—and their students—will have access to the Cray X-MP, christened the "National High School Supercomputer." NERSC, DOE, and Cray Research are joining together to make the supercomputer available for this purpose, and the NERSC staff will provide support to the teachers as they implement this project.

"This year's summer workshop was a first step," said Fitzgerald. "In the future, we'd like to see more teachers involved. This supercomputer is a unique educational resource, and we'd like to see it used not only in the classroom but also by kids in science and computer clubs or in gifted student programs, or perhaps even by scout troops. Just like supercomputing itself, the possibilities are enormous." ■

—*G.V.K.*

For more information about these programs, contact Sue Wiebe at (415) 423-9394.

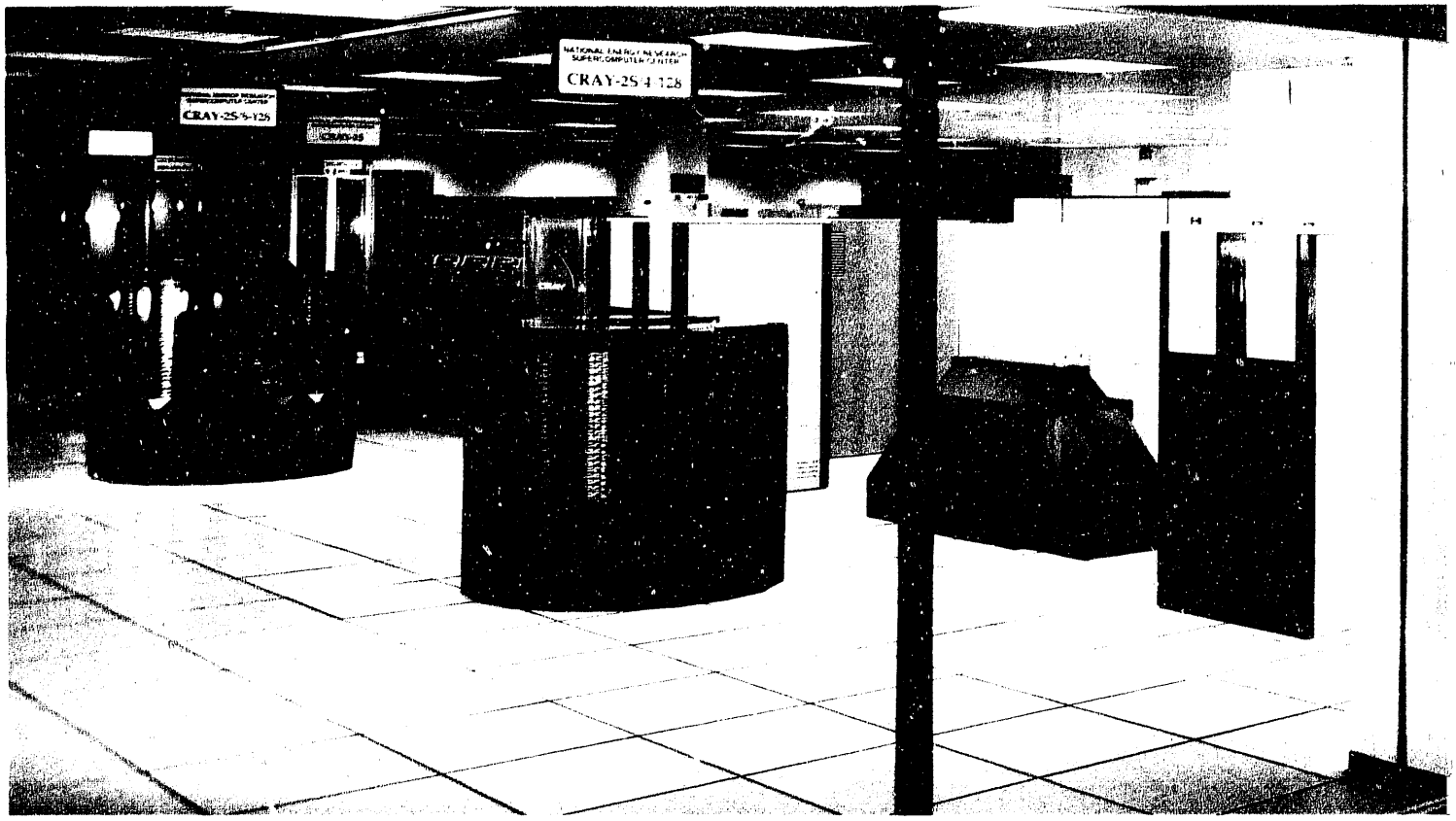
About the National Energy Research Supercomputer Center

The National Energy Research Supercomputer Center (NERSC) provides supercomputing capability for researchers whose work is supported by the Office of Energy Research (OER) of the Department of Energy. Currently there are about 4,500 people who use our supercomputers. These users work within nearly 150 different institutions across the United States, including national laboratories, universities, private laboratories, and industrial organizations. The OER programs we serve include (1) Magnetic Fusion Energy, (2) Superconducting Super Collider, (3) High Energy and Nuclear Physics, (4) Basic Energy Sciences, and (5) Health and Environmental Research.

Initially called the Controlled Thermonuclear Research Computer Center and later known as the National Magnetic Fusion Energy Computer Center, NERSC was formed in 1974 to meet the computational demands of the national magnetic fusion energy program. This center was the first organization to provide centralized supercomputing via network access. In 1983, OER expanded the center's role to provide service to other OER projects besides those in magnetic fusion. This expanded purpose was officially recognized in April 1990, when we acquired our present name.

NERSC now has four multiprocessor supercomputers available to users: a Cray-2 with eight processors and 134 million words of memory, a Cray-2 with four processors and 134 million words of memory, a Cray-2 with four processors and 67 million words of memory (serial 1), and a Cray X-MP with two processors and 2 million words of memory. All of these Crays operate with the Cray Time-Sharing System (CTSS). Within the next couple of years we anticipate acquiring a next-generation supercomputer.

A newly installed network, called ESNET (for Energy Sciences Network), connects various user sites across the United States to the central facility at Livermore and to each other. ESNET, which NERSC administers for the Department of Energy, supports several transmission protocols (for example, TCP/IP and DECNET), thus facilitating service to different



research communities. The ESNET backbone, based on fiber-optic technology, supports data transmission at the T1 rate (1.5 million bits per second). ESNET is cross-connected to several other major backbone networks (for example, NSFNET) and has international connections to Europe and Japan.

Our archival file storage system is the Los Alamos Common File System (CFS), which uses as its main storage mechanism a Storage Technology Corporation automated cartridge system, with the storage cartridges accessed robotically. The present storage capacity of this system is about 4 trillion bytes.

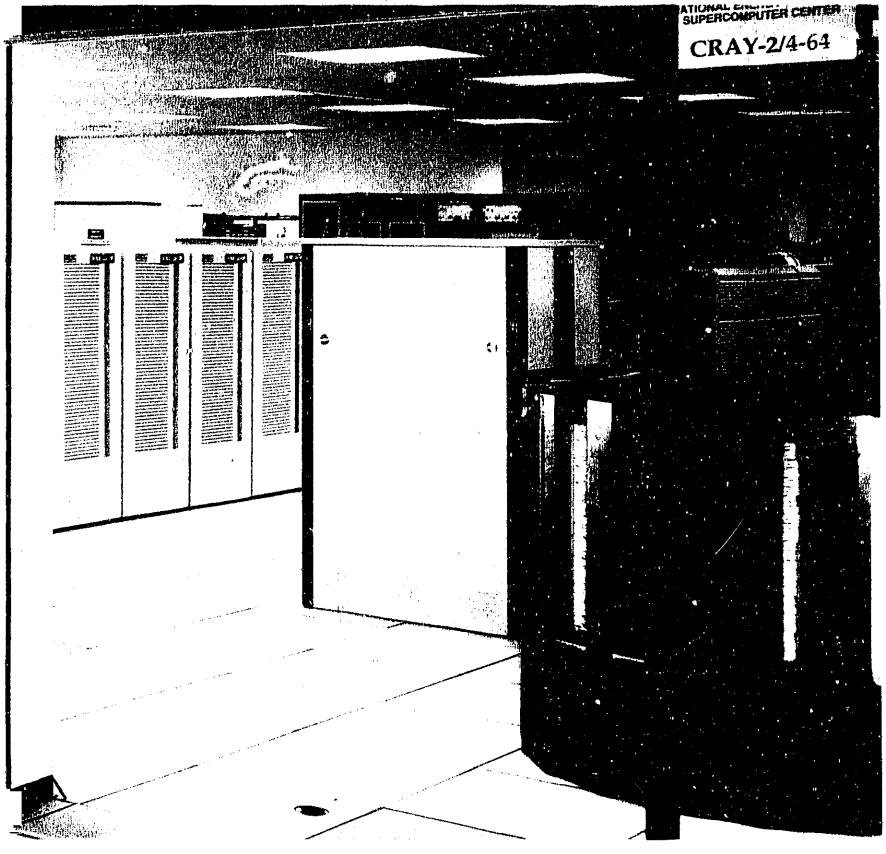
The supercomputing environment at NERSC includes an array of services for users:

- General consulting.
- Network information, support, and troubleshooting.
- User training and education.
- Support of on-line bulletin boards.
- A monthly newsletter (the *Buffer*) that focuses on systems issues and applications.
- An electronic mail system that supports return receipts and attached files for communications within the NERSC domain.
- On-line documentation with a menu-driven interface.
- A library of applications code abstracts containing more than 300 entries.
- Consulting support for more than 70 applications codes.

We have available a number of important user libraries, and we actively support the multitasking of applications codes. We also provide advanced graphics software, such as the visualization tools produced at the National Center for Supercomputing Applications. ■

—*A.A.M.*

INSTANT ACCESS. Data can now move through ESNET's fiber-optic circuits at a rate of 1.5 million bits per second, connecting research centers with each other and with the NERSC supercomputers. From the control center at NERSC, operators keep track of ESNET's performance and, if necessary, troubleshoot via remote control.



A ROOMFUL OF POWER. The red-clad superstars of the NERSC machine room are, left to right, a Cray X-MP (tall and to the rear), an eight-processor Cray-2, a four-processor Cray-2, and a second four-processor Cray-2 (at extreme right). Combined, the four supercomputers online to NERSC users have 18 processors and 337 million words of memory, with a peak computing capacity of 8.2 billion arithmetic operations per second. (In the time it typically takes to read this caption, these machines can perform about 180 billion arithmetic operations.)

END

DATE FILMED

02/20/91