HIGH-PERFORMANCE COMPUTING AND PLASMA PHYSICS

Computer models such as the 'numerical tokamak' will advance all areas of plasma science, including basic plasma physics, fusion physics, space plasmas and industrial plasma processing.

John M. Dawson, Viktor Decyk, Richard Sydora and Paulett Liewer

The physics of ionized gases is a relatively new science. Not until the development of the electrical industry were controlled experiments on ionized gases possible, and so plasma physics is only about 100 years old. The early part of this century saw some pioneering studies of gas discharges and radio propagation in the ionosphere. However, the real impetus came with the initiation of the controlled thermonuclear reaction programs in the 1950s and with the discoveries of the Van Allen belts and the solar wind in the 1960s. Studies in these areas showed that plasma behavior is much more complex than had been anticipated.

Plasma behavior is often quite nonlinear. Plasmas exhibit fluid-like turbulence, and because of their interactions with electromagnetic fields, they exhibit many types of collective motion not encountered in more common fluids. These can interact nonlinearly, expanding greatly the types of nonlinear behavior displayed by plasmas. The particle orbits within the collective motions also can be nonlinear, and this gives rise to a large variety of nonlinear phenomena such as the generation of radiation. Plasmas are often created by subjecting a low-density gas to large electric and magnetic forces, which start it out in a highly nonlinear and often turbulent state.

John Dawson is a professor of physics at the University of California, Los Angeles. **Viktor Decyk** is a senior research physicist at UCLA. **Richard Sydora** is a research physicist at UCLA. **Paulett Liewer** is a member of the technical staff in the space and astrophysics section at the Jet Propulsion Laboratory and a member of the NSF center for research in parallel computation at Caltech, in Pasadena, California.

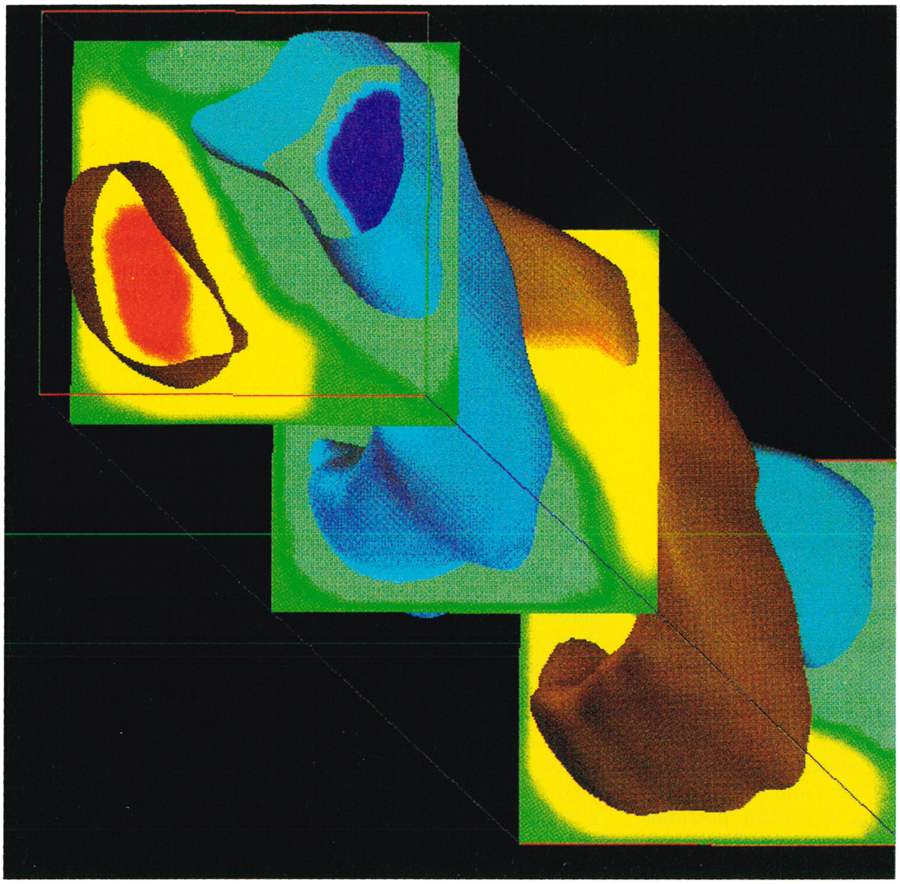
There are many low-temperature plasmas used in plasma processing applications. Such plasmas contain highly reactive chemical radicals and exotic molecules—for example, molecules containing electronically excited atoms, such as Kr*F, found in krypton–fluoride lasers. Many complicated molecular states can be important in these types of plasmas, and so even the task of following just the important ones presents a great challenge that can be met only with the aid of large computers.

Plasma experiments are difficult and expensive and give limited information. The theory of plasma behavior is also difficult and generally requires gross simplifications and approximations. Thus progress in plasma physics, though substantial and steady, has been slow and tortuous. Computer simulation offers a powerful tool for the study of plasmas and promises to expedite progress in the field.

Models

Plasma physics has been a leader in the use of computer modeling. A national computing center for fusion research was established at Lawrence Livermore National Laboratory in May 1974. This facility, which became the National Energy Research Supercomputer Center, was a prototype for subsequent supercomputing centers.

Particle computer models of plasmas[1–3] are among the most successful, and we will focus much of our attention on them in this article. These models emulate nature by following the motion of a large number of charged particles in their self-consistent electric and magnetic fields. Today's supercomputers, such as those



Electrostatic potential surfaces in a fusion plasma, as modeled on a Cray computer at the National Energy Research Supercomputer Center at Lawrence Livermore National Laboratory. Each color represents a surface of constant potential. The three-dimensional contours were taken after 870 ion cyclotron periods, by which time the internal kink instability was saturated. **Figure 1**

made by Cray, can follow the full dynamics of several million electrons and ions and their fully self-consistent fields for 10^4 plasma oscillation times. [The plasma oscillation time, or period τ_p of charge-density vibration, is given by $2\pi/\omega_p$ or $(\pi m_e/2ne^2)^{1/2}$, where ω_p is the plasma frequency, m_e is the electron mass and n is the electron density.] The Concurrent Supercomputing Consortium's Intel Delta Touchstone parallel computer at Caltech has run models containing over 10^8 particles with excellent efficiency. While the number of particles used in computer models is much smaller than that in laboratory plasmas, techniques have been developed to make these models quite realistic—for example, by reducing collisions and noise.

Particle models have proved capable of describing not only the collective motions but also kinetic and nonlinear effects; it is possible to see plasma echoes, wave breaking, anomalous resistivity and much more. Particle simulations have led the way in studies of parametric instabilities produced by intense electromagnetic radiation propagating in a plasma.[1,3] They have also led the way in studies of plasma accelerators for possible use in high-energy physics[4] and of many microwave radiation sources. And they have elucidated many basic physical processes in plasmas.[1]

Solving the full dynamical equations for the charged particles is often impractical. Such problems involve long time evolution or very large systems—10^4–10^6 Debye lengths in size—and call for a variety of strategies. (The Debye length is given by $v_{T,e}/\omega_p$, where $v_{T,e}$ is the thermal velocity of the electrons.) One method is to extend particle models by implicit integration of the particle equations of motion.[2,5] This provides a means to suppress high frequencies and short wavelengths. Another method uses "gyrokinetic" models.[6] Here one averages over the particles' rapid gyration about a magnetic field, keeping only slow drifts. The particles are treated as rings of charge and current that move according to well-known drift equations, allowing much larger time steps.[7] The biggest limitation comes from the rapid motion of electrons along the magnetic field. However, the important disturbances tend to have long wavelengths along the field, and by keeping only these we eliminate the associated electron high frequencies. (The electron frequencies become comparable to the wave frequencies.) A third strategy, used when electron inertia is unimportant, treats the electrons as a massless fluid and the ions as particles.[8] This hybrid model allows computations on the ion time scale. One of the problems described below uses such a model.

Another method is to use fluid models. Magnetohydrodynamic models have existed since the beginning of plasma modeling.[9] Over the years more complete fluid models have been developed that include resistivity, viscosity, heat conduction and other nonideal effects.[10,11]

Simple fluid models leave out the physics that is responsible for much of the important behavior of plasmas. They leave out kinetic effects, which contribute to damping and to nonlinear saturation of unstable modes and can drive instabilities. Such effects are probably at the heart of determining properties of plasma and heat transport across magnetic fields. A promising, somewhat more complicated approach is to incorporate kinetic effects into fluid models. One can do this by making some simple approximations to the kinetic effects and assuming that the plasma deviates locally only slightly from a Maxwellian velocity distribution. This is known as a gyrofluid model.[12,13]

The remarkable increases in the speed and memory of computers, together with improvements in algorithms, have brought modeling to the point where it can tackle many important problems. Below we will give some examples of problems that have been run on existing supercomputers. Although it is not difficult to find important problems that are beyond the capabilities of present computers and techniques, the great increase in power made possible through parallel computing offers the potential for successful attacks on many of these problems.

Parallel computing on basic particle codes

The use of parallel processing in plasma physics goes back quite a few years. An early example, from the mid-1970s, was an experimental Culler Harrison computer at the University of California, Los Angeles, that had internal parallelism and was designed specifically for particle simulation of plasmas.[14] Its power matched that of the CDC 7600, the supercomputer of its day, and it routinely performed simulations of rf heating of fusion plasmas containing 2.5×10^5 particles. Its programming language was quite similar to FORTRAN-90, which is now used by Thinking Machines Corp.

Parallel processing became more widely available when the Cray computer started to support multitasking in the mid-1980s. The Cray is a shared-memory machine, and multitasking is subroutine-level parallelism. David Anderson of the National Energy Research Supercomputer Center has reviewed some of these early successful efforts.[15] One of the first parallel particle-in-cell codes on shared-memory computers was developed by Erick Horowitz at NERSC.[16] In this code, the particle advancement was trivial to parallelize, and charge deposition was parallelized by using an address-sorting scheme developed earlier for vectorization, which guaranteed that particles would have no conflicts when depositing charge.

Particle-in-cell codes were also developed for the Connection Machine, a distributed-memory Thinking Machines Corp computer in which message-passing is hidden from the user by programming in a high-level language. One early example was a code by Robert Jackson and Ernest Zaidman at the Naval Research Laboratory that was programmed in *LISP, before FORTRAN-90 was available.[17] Other codes soon followed. At that time there was substantial pessimism that distributed-memory parallel computers using message-passing would prove useful. That pessimism turned out to be unfounded, and particle-in-cell codes were soon also developed for hypercube computers.

Two methods have generally been used for partitioning a problem on a distributed-memory parallel computer such as the hypercube. The simpler technique is to distribute the particle data among the processors and replicate all other quantities on each processor.[18] Particle calculations are then trivially parallel, except for the addition of charge and current densities across the processors. Field calculations are not parallelized; because the particle calculations dominate, this works quite well when there are only about ten processors and the field quantities are small enough to fit in the memory of each processor. One of the problems described below uses this technique.

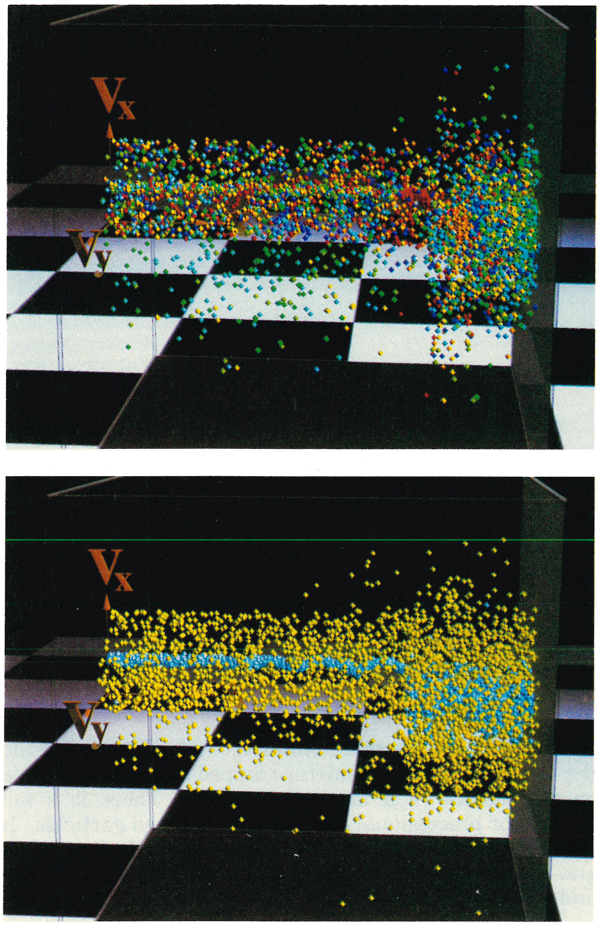
For larger problems with more processors, this technique becomes inefficient or even impossible. One must resort to a domain decomposition scheme where both particles and fields are distributed. An example of such a domain decomposition is the general concurrent particle-in-cell algorithm, where each processor is responsible for some given region of space and the particles that reside there.[19] As particles move from one spatial region to another, the GCPIC algorithm passes them to the appropriate processor. This scheme is efficient because each particle must reference many field values during the calculation of its position and velocity, and there are fewer data to communicate if particles are sent to the fields than if fields are sent to the particles.

Implementing this scheme did not generally change the particle calculations, but a new subroutine was created to manage the passing of particles from one processor to another. A distributed fast Fourier transform also was needed. This scheme has been used in shock-wave studies in one dimension.[20] Computational load balance requires that each processor have approximately an equal number of particles, so that if the density is not uniform, the spatial regions are not equal. Dynamic load balancing, where the spatial boundaries are allowed to change during the course of the simulation to maintain an equal number of particles, has also been tried.[21]

The numerical tokamak

A good example of how plasma physicists are beginning to employ parallel computing is the so-called numerical tokamak. Around 1990 it was realized that present techniques of plasma simulation, in combination with the power of parallel computers coming into use, offered the possibility of modeling experimental tokamaks using basic physics models.[22] Such models would significantly improve our understanding of energy and heat transport in fusion devices and so would have a great impact on the fusion program. We could cheaply test how a fusion reactor might be improved before building costly machines. Of course, modeling would not replace experiments. A model, no matter how sophisticated, is only an approximation to a device. However, modeling might save a generation or more of machines.

Models with demonstrated predictive capability are the key to such an approach. Modeling has duplicated many experimental features of tokamaks.[22] However, the prediction of the behavior of even one experimental discharge has not been attempted. No simulation codes exist that can treat a large enough system, nor do any of the models include enough physics to be fully realistic. Some of the present codes may do a good job of treating the plasma dynamics of the central core, but they do not incorporate the atomic physics of fuel recycling, impurity ionization or the like. Other codes treat the atomic physics quite well but cannot do the plasma physics. Combining these two types of codes presents no fundamental difficulty but will require substantial manpower.



Solar wind termination shock as simulated on the Intel Delta Touchstone computer at Caltech. The one-dimensional simulation is known as a "hybrid" because it treats the electrons as a massless fluid and the ions as particles. The pictures show ion phase space: Each sphere represents an ion in the simulation and shows the ion's position and two of its velocity components. The ions are injected from the left; the shock forms on the right and propagates to the left. The top panel shows the ions colored by processor; the lower panel shows the ions colored by population, with thermal solar wind ions shown in blue and hydrogen pickup ions in yellow. (Visualization by Erik Matson, Jet Propulsion Laboratory Supercomputing Project.) **Figure 2**

To give some idea of the state of particle simulations on present-day Cray computers, we note that gyrokinetic models have been run at UCLA with dimensions corresponding to minor radii of 128 ion Larmor radii (256 grid spaces across the plasma) and with 32 grid spaces in the toroidal direction; the toroidal grid size is about 100 times the minor cross-section grid size. This should be compared with values for a modest-size tokamak of 300 ion Larmor radii in the radial direction. The Larmor radius for ions is about 2 mm, and experiments indicate that the important turbulence scale is somewhat larger. Another significant comparison is the number $a_0\omega_p/c$ of collisionless skin depths within the minor radius a_0. In present calculations this ratio is between 8 and 64, whereas for a tokamak of modest size it runs from 60 to 600. These numbers imply a need to increase the number of grid points in the minor cross section by 10–100. Also, it may be desirable to increase the toroidal resolution by, say, 2. These estimates imply an increase in the number of simulation particles from 10^6–10^7 (found empirically to be adequate for present simulations) to a few times 10^8. The speed of present Cray computers is sufficient to follow existing systems through about 10^{-4} seconds of discharge. Establishing steady-state turbulence levels and hence transport rates probably requires millisecond simulations and thus another order of magnitude in speed. One does not have to simulate a full confinement time any more than one would have to wait for equalization of temperature throughout a body to determine thermal conductivity.

Inclusion of atomic effects will probably not increase the required computing by more than a factor of two. The calculations are fairly straightforward and generally do not have to be carried out every time step.

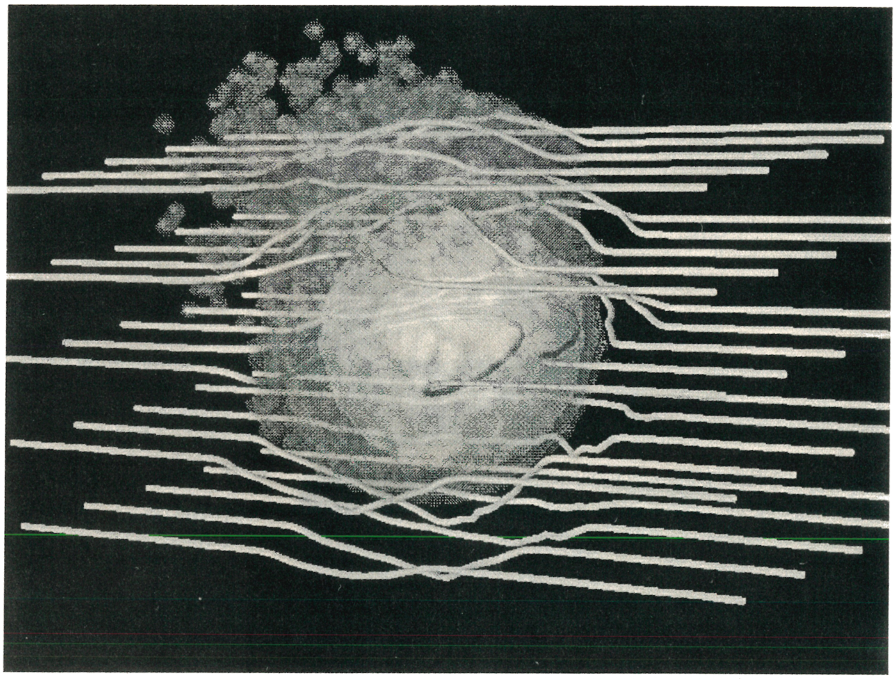
Parallel computers are being built with hundreds to thousands of processors and sufficient memory to handle the tokamak problem. For example, a three-dimensional electrostatic particle-in-cell code implemented by researchers at the Jet Propulsion Laboratory and UCLA on the 512-node Intel Delta Touchstone at Caltech has run 1.47×10^8 particles and shows very high parallel efficiency.[23] Experience at the Princeton Plasma Physics Laboratory on the CM2 shows that this machine provides the required memory and substantially improved speed over single-processor machines.

The numerical tokamak project was proposed in 1990, but it was realized that no existing laboratory or university had the capability to carry out this ambitious program.[22] It was also clear that a large plasma modeling capability existed within the fusion community. In January 1991 plasma modelers met at UCLA with some parallel computing experts to explore the possibility of establishing a consortium to carry out a numerical tokamak program. Additional meetings followed. A proposal was submitted to the Department of Energy's High Performance Computing and Communications Initiative as a "grand challenge" in computing. The physics aspect of the challenge was to model successfully the core transport in tokamaks.

As for the computing aspect, the proposal contained four goals:

▷ to learn how to use parallel computers and new computational tools effectively with the wide variety of codes used in fusion plasma models (particle models, gyrokinetic models, fluid models and so on)
▷ to develop an effective means of comparing results from different codes
▷ to develop effective methods for handling and transmitting very large data sets
▷ to develop effective methods for extracting useful information from very large three-dimensional results.

The prospective consortium asked HPCCI for \$1.26 million in addition to what DOE's Office of Fusion Energy was providing. While DOE could not support the initial ambitious plan, HPCCI nonetheless declared the numerical tokamak a grand challenge problem, and a program was initiated by HPCCI and OFE. HPCCI provided approximately \$200 000 for 1992–93, and the Office of Fusion Energy concentrated its modeling efforts on this grand challenge. An active consortium was formed comprising 11 institutions: Cornell University, General Atomics, the Institute for Fusion Studies–University of Texas, Jet Propulsion



Artificial comet AMPTE as simulated by Ross Bollens of the University of California, Los Angeles, using a three-dimensional hybrid model. The AMPTE plasma cloud of neutral barium atoms and the solar wind magnetic field lines are shown. The solar wind blows from the upper right to the lower left. The magnetic field lines drape around the cloud and show a three-dimensional distortion. A jet of accelerated barium ions can be seen coming from the cloud top. (From Bollens's PhD thesis, UCLA, 1993.) **Figure 3**

Laboratory–Caltech, Lawrence Livermore National Laboratory, Los Alamos National Laboratory, the National Energy Research Supercomputer Center, Oak Ridge National Laboratory, the Princeton Plasma Physics Laboratory, the University of California at Berkeley, and the University of Calfornia at Los Angeles.

Different consortium members have different approaches to numerical tokamak models, because no single model encompasses all of the physical effects involved in plasma transport. With members agreeing to compare results, a better understanding of the codes, of the suitability of various parallel computers and of plasma transport should emerge. In all probability different techniques will prove to be best for different aspects of the problem. Each approach will need the most powerful foreseeable computer.

Results of tokamak modeling

To give a flavor of the type of work being done by the consortium, we look here at a few results from simulations carried out at UCLA and JPL.

One of the more promising approaches to the study of heat and particle diffusion due to low-frequency microinstabilities is the gyrokinetic particle method. This discrete-particle method retains finite-ion-gyroradius effects and wave–particle interactions involved in particle motion along magnetic field lines. Generally speaking, low-frequency microinstabilities tend to be driven unstable by density or temperature gradients perpendicular to the equilibrium magnetic field. The threshold for instability is a strong function of the steepness of the pressure profiles, and the actual threshold values for instability are strongly affected by kinetic processes in the plasma, such as Landau damping. Furthermore, when the instabilities grow and saturate and establish a turbulent spectrum, kinetic effects play an important role. For the electrostatic branches of the instability, the $\mathbf{E}\times\mathbf{B}$ motion of the plasma is dominant, and fine-grained turbulence can cause anomalous diffusion of particles as well as heat.

In tokamak magnetic geometry, particles exhibit complex motions. There are two particle populations: circulating particles that follow the twisted **B** lines around the torus, and particles mirror-trapped in the weak **B** field on the outside of a magnetic surface. There is a net toroidal or precessional drift of these trapped particles. It is this drift motion that can resonate with collective waves and give rise to instability and hence to anomalous heat and particle transport.

The linear and nonlinear behavior of this trapped-particle drift-wave instability has been captured in a gyrokinetic particle model.[22] As the linear instability develops, the fastest-growing mode is dominant. At large amplitudes, sideband and higher harmonic modes grow; there is strong mode coupling and energy transfer between wavelengths. The final state shows smaller-scale disturbances and hence shorter wavelengths. The turbulent fluctuations give rise to particle and heat transport that are about two orders of magnitude larger than what one expects from classical Coulomb collisions in the same magnetic geometry. This result is in the same range as experimental observations of transport, but the detailed scaling with plasma parameters must still be determined.

Another class of low-frequency instabilities consists of the current-gradient-driven, or kink-type, instabilities. These involve more macroscopic changes in plasma motion. Because these instabilities involve current, they result in perturbations in the magnetic structure of the plasma. These can be long-wavelength perturbations involving, say, the $m=1$ poloidal and $n=1$ toroidal modes; such perturbations give global magnetic topology changes, while coupling to $m>1$, $n>1$ modes gives smaller-scale magnetic perturbations. Magnetic islands can form in regions known as rational-q surfaces, where field lines close on themselves. Small-scale irregularities can cause island overlap, stochastic magnetic fields and rapid radial transport of heat results.

Figure 1 shows an example of the macroscopic evolution of a kink instability. Again, the gyrokinetic model was used; in this case magnetic perturbations due to parallel currents and inductive electric fields were included. The figure shows electrostatic potential surfaces at saturation. The $n=1$ kink structure in the z, or toroidal, direction is clear; the corresponding poloidal mode number, $m = 1$, is also evident. Magnetic field reconnection occurs at the rational $q = 1$ surface, where field lines close after one transit of the torus; the reconnection leads to dissipation of the magnetic energy into particle kinetic energy. The purely collisionless instability seen in the simulations is extremely rapid—on a time scale of a few microseconds. That is much faster than resistive dissipation, which occurs on millisecond time scales, and there is evidence from high-temperature tokamaks that the collisionless dissipation time scale seen in the simulations is closer to observations.

Parallel computers and space plasmas

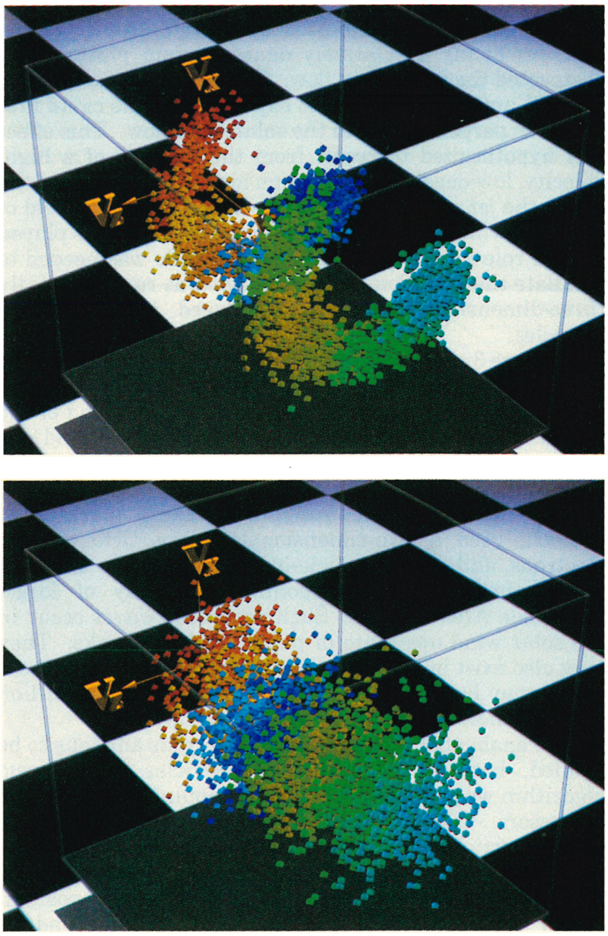
Plasma physicists are using particle-in-cell codes widely to study a variety of problems in space plasma physics. Parallel computers are starting to be used, and much heavier use can be expected soon. To finish this article, we give results from three applications: two run on Caltech's Intel Delta Touchstone using two types of particle-in-cell codes and the two parallel decomposition strategies for particle-in-cell codes described above, and one run at the Cornell National Supercomputer Facility using 2 gigabytes of memory and 12 central processors on two IBM 3090-600J supercomputers.

Figure 2 shows results from the first example, a hybrid model (fluid electrons and particle ions) run on the Delta to study the solar wind termination shock.[8] As the solar wind expands supersonically beyond the solar system, it interacts with the interstellar plasma. The solar wind termination shock exists where the wind makes a transition to subsonic flow in response to the pressure of the interstellar medium. The Voyager and Pioneer spacecraft may soon encounter this shock. The purpose of this work was to study the structure of the shock and its potential for accelerating particles to cosmic-ray energies. The simulations may indicate what precursors of the shock we might expect in the impending encounters.

Hybrid codes typically have hundreds of particles per grid point, and particle computation uses up most of the time. The simple parallel decomposition described above was used to insure balanced particle loads on the processors. The processors start with equal numbers of particles, and the particles remain in their original processors. As new particles enter from the left, they are divided among the processors using a card-dealing algorithm. Each processor has a copy of the entire grid of typically 3000 points so that it can do interpolations without interprocessor communication. Global sums of the ion densities and currents over all the grids are used to find the fields.

The simulations of the solar wind termination shock have two ion populations: thermal solar wind ions and energetic "pickup" ions. Pickup ions are formed by ionization of interstellar neutrals that enter the heliosphere and are swept up by the solar wind. It has been hypothesized that these pickup ions are the seed population for low-energy anomalous cosmic rays. In this hypothesis, the pickup ions are swept out to the solar wind termination shock, where they are further energized to cosmic-ray energies by shock processes.

The results in figure 2 are shown in ion phase space: Each sphere represents an ion in the simulation and is plotted in $xv_x v_y$ phase space—that is, the position of the sphere shows the ion position and two velocity components. Two percent of the ions are plotted. The solar wind ions enter the simulation box from the left, moving to the



Decay of large-amplitude Alfvén waves as seen in a simulation that takes into account the electric and magnetic interactions of particles. The two-dimensional simulation was run on the Intel Delta Touchstone computer at Caltech. The upper and lower panels show ion phase space early and late in the simulation, respectively. Each sphere shows an ion's position in the magnetic field direction y and its two velocity components perpendicular to the field. The ions are colored by processor, and the domain decomposition used in the parallel code shows up as bands of color along the y axis, the direction of decomposition. (Visualization by Matson.) **Figure 4**

right. The right-hand boundary is reflecting, and the interaction of the ions moving to the right with those reflected causes the shock. The left-moving shock is at the interface between the low-density and high-density regions. In the top panel, at $t = 60$, each of the ions is colored according to which processor computes its orbit.

In the lower half of figure 2, for $t = 80$, the ions are colored according to population: Thermal solar wind ions are blue, and hydrogen pickup ions are yellow. Note that a few ions have moved out ahead of the shock. These ions were reflected from the shock and, having gained energy in the process, are moving back upstream at about twice the shock speed (large negative v_x). The reflected pickup ions generate a large-amplitude compressional magnetosonic wave upstream of the shock.

This wave produces the small kinks in the thermal ion distribution ahead of the shock. The solar wind flow sweeps the waves back into the shock. At later times, some of the reflected pickup ions are apparently scattered back toward the shock by the wave, again gaining energy in the process. This represents the first stages of a first-order Fermi acceleration process in which ions are energized by bouncing back and forth between the converging waves and the shock. Future simulations will study this acceleration of pickup ions further to determine if the pickup ions are indeed the source of the anomalous cosmic rays.

The second example is a simulation of the artificial comet AMPTE. This simulation, run at the Cornell National Supercomputer Facility and carried out by Ross Bollens of UCLA, used a hybrid model like that described above.

In the AMPTE experiment a cloud of neutral barium atoms was released in the solar wind near the Earth. The barium was gradually ionized by uv radiation from the Sun, producing a relatively dense barium plasma that obstructed the solar wind flow. The cloud was about 1000 km in diameter. Contrary to expectations, the cloud first deflected perpendicular to the solar wind flow. This effect was hypothesized to result from the ejection of a high-velocity, low-density jet of barium ions from one side of the cloud; the jet was accelerated by the $\mathbf{v} \times \mathbf{B}$ electric field of the solar wind. It was clear that ion kinetic effects played a large role here, and hence a hybrid code was needed to simulate the situation. The code that was run was a fully three-dimensional one that employed about 10^7 ion particles.

Figure 3 shows a view of the AMPTE cloud and the solar wind magnetic field lines. The jet can clearly be seen emanating from the top of the cloud. The draping of the magnetic field lines around the cloud is visible, and one can see that the lines take on a three-dimensional structure.

The third example, run on the Delta machine at Caltech, used a two-dimensional, full-particle (kinetic electrons and ions) particle-in-cell code in a study of plasma heating by the spontaneous decay of large-amplitude Alfvén waves. Such nonlinear waves occur in the solar wind upstream of planetary bow shocks. They may also exist in the lower solar corona; heating by these waves may be partially responsible for the high (million degree) temperatures of the outer corona.

To enable the heating of both electrons and ions to be studied, both were treated as particles. The GCPIC algorithm was used to divide the computation among the processors. As described above, the GCPIC algorithm divides particles among the processors by partitioning the simulation domain and assigning each processor a grid partition and all the particles in it. A slab decomposition of the two-dimensional grid was used; domains had all values of x and a range of values of y.

Figure 4 shows ion phase space at two times in the simulation. Each sphere represents an ion's position in the magnetic field direction y and its two velocity components perpendicular to the field. Ions are colored by processor, and the domain decomposition used shows up as bands of color along the y axis.

The simulation is started with a single large-amplitude, circularly polarized Alfvén wave propagating along y. The helical structure in phase space at the early time results from the coherent ion motion in the wave. This wave is observed to decay spontaneously into other waves (Alfvén and ion sound waves) propagating both parallel and at angles to the magnetic field.[24] Some of these are rapidly damped by the plasma. The lower part of figure 4 shows the ion phase space after the original wave and decay waves have been damped. The ions have been heated in this process, as can be seen by the much broader distribution of ion velocities.

References

1. J. M. Dawson, Rev. Mod. Phys. **55**, 403 (1983).
2. C. K. Birdsall, A. B. Langdon, *Plasma Physics via Computer Simulation*, McGraw–Hill, New York (1985).
3. W. L. Kruer, *The Physics of Laser Plasma Interactions*, Addison–Wesley, Redwood City, Calif. (1988).
4. C. Joshi, W. B. Mori, T. Katsouleas, J. M. Dawson, J. M. Kindel, D. Forslund, Nature **311**, 525 (1984).
5. J. U. Brackbill, B. I. Cohen, eds. *Multiple Time Scales in Computational Techniques*, Academic, Orlando, Fla. (1985).
6. W. W. Lee, J. Comput. Phys. **72**, 243 (1987).
7. L. Spitzer Jr, *The Physics of Fully Ionized Gases*, 2nd revised ed., Interscience, New York (1962).
8. P. C. Liewer, B. E. Goldstein, N. Omidi, J. Geophys. Res. (1993), to be published.
9. See, for example, D. Potter, *Computational Physics*, Wiley, New York (1973).
10. L. C. Charlton, J. A. Holmes, H. R. Hicks, V. E. Lynch, B. A. Carreras, J. Comput. Phys. **63**, 1071 (1986).
11. S. C. Jardin, N. Pomphrey, J. Delucia, J. Comput. Phys. **66**, 481 (1986).
12. G. Hammet, F. W. Perkins, Phys. Rev. Lett. **64**, 3019 (1990).
13. O. T. Kingsbury, R. E. Waltz, Phys. Fluids B **3**, 3539 (1991).
14. R. W. Huff, J. M. Dawson, G. Culler, in *Parallel Computations*, G. Rodrigue, ed., Academic, New York (1982), p. 365.
15. D. V. Anderson, Comput. Phys. Rep. **11**, 117 (1989).
16. E. J. Horowitz, D. E. Shumaker, D. V. Anderson, J. Comput. Phys. **84**, 279 (1989).
17. R. Jackson, E. Zaidman, in *Proc. Twelfth Conf. on the Numerical Simulation of Plasmas*, Lawrence Livermore Natl. Lab., Livermore, Calif. (1987), paper iW2.
18. P. C. Liewer, V. K. Decyk, J. M. Dawson, G. C. Fox, Math. and Comput. Modeling **11**, 53 (1988).
19. P. C. Liewer, V. K. Decyk, J. Comput. Phys. **85**, 302 (1989).
20. P. C. Liewer, V. K. Decyk, J. M. Dawson, B. Lembege, J. Geophys. Res. **96**, 9455 (1991).
21. P. C. Liewer, E. W. Leaver, V. K. Decyk, J. M. Dawson, in *Proc. Fifth Distributed Memory Computing Conf.*, vol. 2, D. W. Walker, Q. F. Stout, eds., IEEE Comput. Soc. P., Los Alamitos, Calif. (1990), p. 939.
22. J. M. Dawson, R. D. Sydora, V. K. Decyk, P. C. Liewer, R. D. Ferraro, Int. J. Supercomput. Appl. **5**, 13 (1991).
23. P. Lyster, P. C. Liewer, R. Ferraro, V. K. Decyk, "Implementation of a Three-Dimensional Particle-in-Cell Scheme on Distributed-Memory Multiple-Instruction Multiple-Data Massively Parallel Computers," preprint, available from the authors.
24. P. C. Liewer, T. J. Krucken, R. D. Ferraro, V. K. Decyk, B. E. Goldstein, in *Solar Wind Seven*, *COSPAR* Colloq. Ser., vol. 3, E. Marsh, R. Schwenn, eds., Pergamon, Oxford (1992), p. 481. ■