PARTICLE SIMULATION OF PLASMAS ON THE MASSIVELY PARALLEL PROCESSOR

I. M. A. Gledhill   L. R. O. Storey

Space, Telecommunications and Radioscience Laboratory
Department of Electrical Engineering
Stanford University
Stanford, CA 94305

ABSTRACT

Particle simulations, in which collective phenomena in plasmas are studied by following the self-consistent motions of many discrete particles, involve several highly repetitive sets of calculations that are readily adaptable to SIMD parallel processing. We describe a fully electromagnetic, relativistic plasma simulation for the MPP. The particle motions are followed in $2\frac{1}{2}$ dimensions (two spatial and three velocity) on a 128×128 grid, with periodic boundary conditions. The two-dimensional simulation space is mapped directly onto the processor network; a Fast Fourier Transform is used to solve the field equations. Particle data are stored according to an Eulerian scheme, i.e., the information associated with each particle is moved from one local memory to another as the particle moves across the spatial grid.

The method is applied to the study of the nonlinear development of the whistler instability in a magnetospheric plasma model, with an anisotropic electron temperature. The wave distribution function is included as a new diagnostic to allow simulation results to be compared with satellite observations. Since the physics of self-gravitating systems is quite similar to plasma physics, incorporation of free-space boundary conditions and alteration of the field equations enable our code to be used for the study of density waves in galaxies.

Keywords: Particle Simulation, Plasma Physics, Stellar Dynamics, Parallel Processing.

1. INTRODUCTION

The research work described in this paper belongs to a relatively new category: the exploratory use of parallel processors for the particle simulation of rarefied media. By a 'rarefied' medium we mean one in which the collisional mean free paths are not necessarily small on the scale of the systems considered, with the result that it cannot always be approximated satisfactorily as a continuous fluid. Plasma, which is the main constituent of the Universe, is a prime example. Among the very diverse phenomena that can occur in a plasma, some of the most interesting are highly nonlinear, and therefore difficult to analyze theoretically. For the plasma theorist, a powerful alternative and complement to analytic study is the use of particle simulation in a computer.

'Particle simulation' is the generic term for computational procedures in which a medium is represented in the computer as an assembly of discrete interacting particles. In a plasma the particles are ions and electrons, interacting through the electric and magnetic fields that they themselves create. At each time step in a simulation, the computer has two distinct tasks to perform: it must update the positions and velocities of the particles, taking account of their accelerations due to the fields, and it must update the electric and magnetic fields, the sources of which are, respectively, the charge density calculated from the positions of the particles, and the current density calculated from both their positions and their velocities.

Now, even a modest volume of plasma may contain a very large number of particles: in Earth's ionosphere, for instance, there are typically 10^{12} electron-ion pairs per cubic metre. Computer simulations necessarily involve much smaller numbers, not more than a few times 10^6 for present-day single-processor computers ('uni-processors'), so each of the simulation particles actually stands for very many particles in the real world; they are sometimes called 'superparticles' for this reason.

Besides the reduction in the number of particles, other simplifications are often made in order to reduce the demand for computing time. The commonest is to reduce the dimensionality, by considering systems in which all physical quantities vary in only one or two dimensions. This simplification can be very helpful, provided that it is authorized by the symmetry of the problem, but otherwise it is objectionable because it makes the simulation unrealistic. Another common way of simplifying plasma simulations is to ignore electromagnetic radiation by assuming that the speed of light is infinite. However, the real world is three-dimensional and the speed of light is finite, so most of the outstanding problems in plasma physics that are amenable to particle simulation will ultimately need to be tackled in three dimensions (3-D) using a fully electromagnetic (EM) code.

Probably the most advanced EM particle code in existence is the TRI-dimensional STANford code TRISTAN, written in assembler language for the Cray-1 computer; see Ref. 1 for an account of an earlier version of this code. It follows the motion of about 5×10^6 particles in a cubical volume divided up into 128^3 cells, i.e., each side of the cube is divided into 128 units. These numbers are not extravagant: if anything, they err on the side of modesty, and codes with more particles and cells are likely to be required in the future. Already, however, the preliminary tests of TRISTAN have revealed a severe problem of computer usage. Though the assembler-language code has been carefully optimized, a single time step requires 2-3 minutes of CPU time on the Cray-1, and a typical simulation requires 500-1000 steps. Difficulty in obtaining the requisite amount of computer time has already set back several research programs where the use of TRISTAN was envisaged. This is just one instance of a critical situation which is now widely recognized, namely that advanced particle simulation is up against a barrier due to the speed limitations of uni-processors.

Other advanced approaches to 3-D plasma simulation exist, namely the statistical methods involving numerical integration of the collisionless Boltzmann equation or of the Fokker-Planck equation. Formally, they are equivalent to following the motion of a fluid in a 6-D space, which has three dimensions of velocity as well as the three dimensions of position. For a given number of cells in position space, however, these methods demand even greater computing speed.

The prospects for large increases in the speed of uniprocessors are not encouraging: fundamental physical constraints on VLSI technology are expected to limit them to factors of less than 100. For larger increases, we must look to multi-processors, i.e., computers consisting of multiple processors arranged in parallel architectures, such as the MPP.

Though the MPP was designed originally for processing image data from the Landsat satellites, its architecture, involving a large number of simple processors with nearest-neighbor connections, is well suited to the particle simulation of rarefied media, and incidentally to fluid simulation as well (Ref. 2). The motion of particles or fluid in a given spatial cell is determined only by conditions prevailing inside that cell, and at its boundaries with neighboring cells. (At least, this is the case so long as we refrain from making the approximation in which disturbances propagate across the array of cells instantaneously). Hence problems concerning such motion can be mapped readily onto simple arrays of processors in which direct connections exist only between nearest neighbors.

In the present program of research, our initial aims were to gain enough experience in the use of the MPP to be able to answer the following questions:

- Which of our plasma simulation problems can be solved on this type of multi-processor?

- Is there a significant gain in speed, compared with plasma simulations on a uni-processor?

- Could the capability of the MPP for plasma simulation be improved in any simple way?

- Would any other type of multi-processor be better suited to our problems, and if so which?

In sum, we wished to investigate the potentialities and limitations of massively parallel processors for plasma particle simulation.

Shortly after the investigation began, however, its scope was extended to include particle simulation of problems in stellar dynamics, this in collaboration with Dr. Bruce Smith of NASA Ames Research Center. Stellar dynamics is very similar to plasma electrodynamics in respect of its basic physics. Both plasmas and stellar systems are examples of rarefied media, and problems concerning either of them are cases of the classical N-body problem with inverse square law interactions. Each, of course, also has its specificity: stellar dynamics involves long-range interactions, from which plasmas are exempt because of Debye shielding; plasma electrodynamics involves magnetic fields, which have no counterpart in non-relativistic stellar dynamics. Nevertheless, the two fields have much in common, so naturally there are close similarities between the methods of particle simulation that have been developed for each of them. For the same reason, many possibilities exist for cross-fertilization between them, and for joint efforts to solve common difficulties. These considerations led us to extend the scope of the program in this way.

The present paper is an account of the work performed during the first year. Section 2 outlines the two physical problems, one in plasma electrodynamics and the other in stellar dynamics, that have been chosen for simulation, and how we are approaching them; for simplicity, both simulations are in two spatial dimensions instead of three. Section 3 describes the numerical techniques used for the plasma simulation, and Section 4 the modifications required for the gravitational simulation. Section 5 indicates how these various algorithms are being implemented on the MPP, while the current status of the work is described in Section 6. Finally, in Section 7, we draw some provisional conclusions and sketch our plans for the future, in both the short and long terms.

2. PHYSICS PROBLEMS

Recapitulating, our immediate objectives are to program the MPP to solve, by particle simulation, a significant problem in each of the two fields mentioned above, namely stellar dynamics and plasma electrodynamics.

The simpler of the two problems is the one in plasma electrodynamics, which concerns the nonlinear development of the Doppler-shifted electron gyroresonance instability, or 'whistler instability' for short. It is a very suitable problem for solution on the MPP, for several reasons. Firstly, it is one in which only the electrons are involved, while the ions can be treated as an inert neutralizing background. The simulation algorithms are simpler if only one type type of particle needs to be represented. Secondly, the whistler instability is of the velocity-space variety, so it can occur even in a spatially uniform plasma. By assuming uniformity as an initial condition, we should succeed in sharing the computing load evenly between the different processors. Finally, it is an electromagnetic instability, which means that field disturbances originating in any one spatial cell propagate to the other cells at finite velocities. Such propagation can be modeled readily on the MPP, in which each processor is connected only to its four nearest neighbors. Despite its apparent simplicity, the nonlinear evolution of the whistler instability has not been simulated before in the conditions in which we intend to do so. This problem has applications both to space and to fusion plasma physics.

To the best of our knowledge, all previous simulations of the whistler instability have been one-dimensional. The most recent, by Bharuthram and Baboolal (Ref. 3), were in several other respects quite similar to those that we are proposing. Thus, for instance, these authors used a fully electromagnetic code, gave the electrons a bi-Maxwellian velocity distribution at the outset, and followed the growth of the instability into the nonlinear regime. They obtained a variety of interesting scientific results: for instance, the wave mode that, during the initial linear phase of the instability, grew most rapidly, died out in the post-saturation phase, and ultimately was replaced by a slower-growing mode which dominated the wave spectrum at the end of the simulation.

Using our MPP code, we intend to study the nonlinear evolution of the whistler instability under much the same conditions, except that our simulations will be performed in $2\frac{1}{2}$-D (i.e., in two spatial dimensions, but with all three components of velocity). We expect that the increase in dimensionality will lead to qualitatively new scientific results: in particular, we anticipate that waves will be generated over a broad frequency band, with their wave normal directions spread widely around the direction of the steady magnetic field. We shall study how the characteristics of the wave field in the nonlinear regime depend on the initial plasma conditions.

For these simulations, the code will have to be enhanced with sophisticated diagnostics, especially for the fields. The fact that it already includes a two-dimensional FFT should be very helpful in this respect, enabling wave power spectra, dispersion relations, temporal autocorrelation functions, etc., to be derived by existing techniques, developed previously for serial processors.

Additionally, we propose to employ a new diagnostic procedure, hitherto used only for analyzing whistler-mode wave data from satellites in the Earth's magnetosphere. This is the so-called 'wave distribution function' (WDF), which specifies how the wave energy density is distributed with respect to frequency and to wave normal direction (Refs. 4–7). The ready availability of this diagnostic will, in the future, facilitate comparisons between the results from numerical simulations of the whistler instability, and observations of natural whistler-mode wave fields generated by the same instability in space.

In stellar dynamics, we shall investigate the stability of an axisymmetric system (galaxy or protoplanetary nebula), albeit that problems of this type have been simulated previously on conventional computers. The interest of repeating one of these simulations on the MPP is that it will introduce some of the difficulties that are bypassed in the whistler instability problem. Although, again, the dynamics involve only one type of particle, the absence of a neutralizing background of particles of another type means that the inter-particle forces are not shielded in any way, so they are truly long-range. For this reason, all physically significant systems are spatially non-uniform. Finally, the times required for gravity to propagate across such systems are very small compared with the time scales characterizing their dynamics (e.g., the rotation period of a galaxy), so little accuracy is lost and much computing time saved by assuming that the propagation is instantaneous. Thus, besides the physical interest of this problem, its simulation on the MPP will have interesting computer science aspects.

3. NUMERICAL TECHNIQUES

Details of the particle-mesh method may be found in the standard texts on plasma simulation (e.g., Refs. 8–10). We summarize here the techniques chosen for the MPP code.

The purpose of the simulation model is to follow the self-consistent dynamics of fields and particles for several thousand time steps. Fields, densities and other spatial functions are stored as values at points on a grid. Initially, our simulation is restricted to two spatial dimensions for simplicity. The motion of a large number of particles (of the order of 10^5 to 10^6), each representing many particles in the real world, is followed through successive time steps.

Initially, particles are arranged in positions (x, y) that result in the required plasma configuration. For our present purposes, uniform density is specified. Particles are assigned random velocities (v_x, v_y, v_z) according to a Gaussian distribution with suitable thermal and drift velocities.

As is usual in numerical simulations, point particles are replaced by clouds of charge to prevent impulse-like interactions which give rise to collisional effects (Ref. 11). Long-range particle potentials are retained in the Coulomb form to model collective effects, but a shape function

$$S(\mathbf{r}) = \frac{1}{2\pi a^2} \exp(-r^2/2a^2) \qquad (3.1)$$

is introduced so as to 'soften' the particle potential within the range $r < a$ (usually of the order of a Debye length).

In order to calculate the fields set up by the charges, it is necessary to obtain the charge and current densities ρ and $\mathbf{J}$ on the spatial grid. A simple algorithm is used which produces the same results as the Subtracted Dipole Scheme (Ref. 12). The charge density array is compiled from the superposition of a unit charge at each particle's nearest grid point (NGP), plus small contributions at its four neighboring grid points which reproduce the dipole moment of the charge relative to its NGP.

The electric and magnetic fields (**E** and **B**) are governed by Maxwell's equations. With the inclusion of the shape factor, Ampère's and Faraday's laws become

$$\frac{\partial \mathbf{E}(\mathbf{r})}{\partial t} = c\,[\nabla \times \mathbf{B}(\mathbf{r}) - 4\pi\,\mathbf{J}(\mathbf{r}) * S(\mathbf{r})] \qquad (3.2)$$

and

$$\frac{\partial \mathbf{B}(\mathbf{r})}{\partial t} = -\,c\,\nabla \times \mathbf{E}(\mathbf{r})\,, \qquad (3.3)$$

where the asterisk denotes convolution and c is the speed of light. Gauss' law,

$$\nabla \cdot \mathbf{E}(\mathbf{r}) = 4\pi\,\rho(\mathbf{r}) * S(\mathbf{r}) \qquad (3.4)$$

is satisfied by the initial conditions and will continue to hold if the continuity equation is satisfied. However, microscopic inconsistencies between ρ and $\mathbf{J}$ arise due to the use of the mesh (Ref. 13). To prevent the resulting growth of noise in the fields, it is common to split the electric field into longitudinal and transverse components $\mathbf{E}_l$ and $\mathbf{E}_t$ (Ref. 9), using Gauss' law to obtain $\mathbf{E}_l$ and Faraday's law to deduce $\mathbf{E}_t$. It is thus necesary to solve both elliptic and hyperbolic equations in this system; the use of Fourier or Hartley transforms provides a simple solution, as well as helpful diagnostics.

The field equations are integrated in time using a leap-frog method with time step Δt. Fourier transforming (3.2), (3.3) and (3.4) and including the time differencing gives

$$\mathbf{E}_t^n(\mathbf{k}) \;=\; \mathbf{E}_t^{n-1}(\mathbf{k}) + c\,\Delta t\,\{i\,\mathbf{k}\times\mathbf{B}^{n-\frac{1}{2}}(\mathbf{k}) - 4\pi\,S(\mathbf{k})\,[\mathbf{J}^{n-\frac{1}{2}}(\mathbf{k}) \;-\; (\mathbf{J}^{n-\frac{1}{2}}(\mathbf{k})\cdot\mathbf{k}/k^2)\,\mathbf{k}]\} \qquad (3.5)$$

$$\mathbf{B}^{n+\frac{1}{2}}(\mathbf{k}) \;=\; \mathbf{B}^{n-\frac{1}{2}}(\mathbf{k}) - ic\,\Delta t\,\mathbf{k}\times\mathbf{E}_t^n(\mathbf{k}) \qquad (3.6)$$

$$\mathbf{E}_l^n(\mathbf{k}) \;=\; -4\pi\rho^n(\mathbf{k})\,S(\mathbf{k})\,\mathbf{k}/k^2 \qquad (3.7)$$

The superscripts denote the time step at which the field is known:

$$f^n \;\equiv\; f(t_0 + n\,\Delta t) \qquad (3.8)$$

The stability of the method is governed by a Courant-Friedrichs-Levy (CFL) condition:

$$\frac{(k_{\max}\,c\,\Delta t)^2}{4} < 1 \qquad (3.9)$$

where $k_{\max}$ is the highest spatial frequency occurring in the simulation.

Once the fields are known on the spatial grid, the forces on the particles may be calculated by interpolation. The relativistic equation of motion is

$$\frac{d\mathbf{p}}{dt} \;=\; q\mathbf{E} + \left(\frac{q}{m_0 c}\right)\frac{\mathbf{p}\times\mathbf{B}}{\gamma} \qquad (3.10)$$

where p is the relativistic momentum, m_0 is the rest mass, and

$$\gamma = 1/\sqrt{1 - v^2/c^2} \qquad (3.11)$$

A problem arises because of the appearance of the momentum in the vector product in equation (3.9). A well-established solution due to Boris (Ref. 14) is employed: the momentum is advanced using the electric field only for half a time step, then a rotation of the resulting intermediate momentum vector by the magnetic field is carried out and finally the second 'half-push' by the electric field is performed.

One of the advantages of applying the full relativistic treatment is that a top speed for both field and particle propagation is provided, and can easily be tailored so that the CFL condition (3.9) is obeyed.

Advancement of the particles' positions closes the main simulation loop, since the charge and current densities may be evaluated again.

4. GRAVITATIONAL SIMULATION

Particle-mesh simulation is a well-known technique in the investigation of the evolution of spiral galaxies (Ref. 8). Since the algorithms for gravitational simulations are so closely related to those for plasma, it seems worthwhile to construct them in parallel. The main differences between the programs are summarized here.

The gravitational potential $\phi(\mathbf{r})$ is used instead of the fields. It is governed by Poisson's equation

$$\nabla^2\phi(\mathbf{r}) \;=\; 4\pi G\,\rho(\mathbf{r}) \qquad (4.1)$$

where ρ is the mass density and G the gravitational constant. Solutions are obtained by convolving the Green's function for this equation, i.e., the single particle potential, with the particle number density function ρ/m_0. The point particle potential

$$\phi \;=\; -\frac{G\,m_0}{r} \qquad (4.2)$$

is replaced by the 'soft' potential

$$\phi_p \;=\; -\frac{G\,m_0}{\sqrt{r^2 + a^2}} \qquad (4.3)$$

so that shape factors do not appear explicitly in this formulation. The convolution is accomplished in Fourier transform space.

For modeling isolated galactic systems, free-space boundary conditions are imitated by confining the particles to 1/4 of the grid, a square of sides L, and truncating ϕ_p beyond $x = \pm L/2$ and $y = \pm L/2$

(Ref. 8). Although periodic images of the simulation plane are still produced by the fast Fourier transform, they can no longer influence each other across the empty buffer zone.

5. IMPLEMENTATION ON THE MPP

It will be seen from the above description that there are two main areas of the simulation algorithm with a high degree of parallelism: handling the fields and densities on the spatial grid, and processing the large number of particles moving though the space. This dichotomy appears in the data structure adopted for the simulation.

In the simplest approach, a spatial grid of 128×128 cells is mapped directly onto the two-dimensional array of processing elements (PE's), as illustrated in Figure 1. The local memory holds ρ, $\mathbf{J}$, $\mathbf{E}_l$, $\mathbf{E}_t$ and $\mathbf{B}$ associated with the corresponding grid point. The contents of the array unit (ARU) memory may be shifted across the network and wrapped around at the edges; extensive use is made of this feature in the fast Fourier transform for solving the field equations. Because it is a global method, the FFT has the disadvantage of requiring communication between distant processors in the network. An attractive alternative is to employ local methods of field solution, in which each processor would demand data from its nearest neighbours only (Ref. 15). However, the FFT is retained for the present as a useful general tool, since its numerical stability is well understood, and also it provides standard diagnostics.

The particle data structure presents a more interesting problem. Provision must be made for the information associated with the particles to move across the network in order to interact with appropriate data on the grid. Two alternative schemes, which may be used in some combination, exist.

In one disposition, information for a given particle is stored in the memory location corresponding to its position on the grid (Figure 1). Local calculations of densities and forces may then be made without communication beween processors. However, the packet of information must be moved when the particle travels to a new cell in the grid. This method has the disadvantages that memory overflow may occur in some locations, and that the workload is not distributed evenly among the processors unless the particle density is more or less uniform.

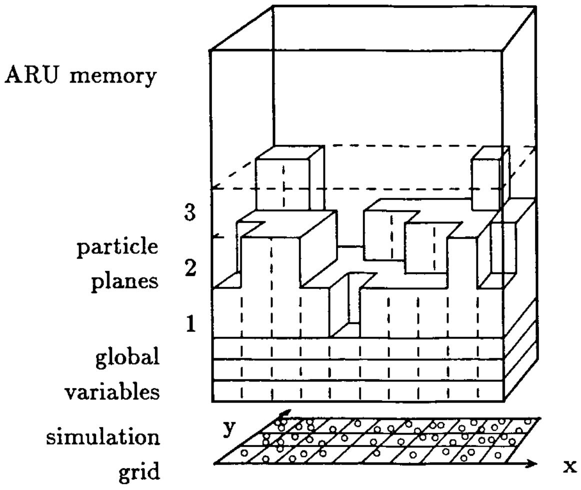

Figure 1. Mapping the particle data to memory. Each local memory contains data for particles in the corresponding cell in the simulation plane.

The alternative is to store particles in a uniform way in memory, in arbitrary locations. Particle and/or field data must then be transported across the network to a common processor when calculations involve both types. This scheme should prove useful when large density variations exist or when particles move though several cells per time step; Hoshino and Takenouchi (Ref. 16) proved a variation of it to be the more efficient scheme in a particle-particle molecular dynamics model on the PAX parallel processor.

In the case of an initially homogeneous plasma, however, large density differences are not produced except in extreme circumstances, and particles may be restricted from traveling more than one cell in a time step by setting the speed of light to this value. For our present purposes, therefore, we decided to use the first of the two schemes outlined above. The scheme also has the attraction of being a 'natural' mapping, and for this reason a variant of it was used in a 3-dimensional particle-mesh simulation of galactic gas dynamics on the ICL DAP by Johns and Nelson (Ref. 17).

Figure 2 shows a flow chart outlining the algorithm. Calculations are carried out in two major procedures, titled 'Particles' and 'Fields'. The only global parallel arrays that have to be carried between these are the variables required by both: ρ and $\mathbf{J}$ (the output of 'Particles' and the input of 'Fields'), and $\mathbf{E}_l$, $\mathbf{E}_t$ and $\mathbf{B}$ (the output of 'Fields' and the input of 'Particles'). These data are stored in the ARU throughout the main time loop.

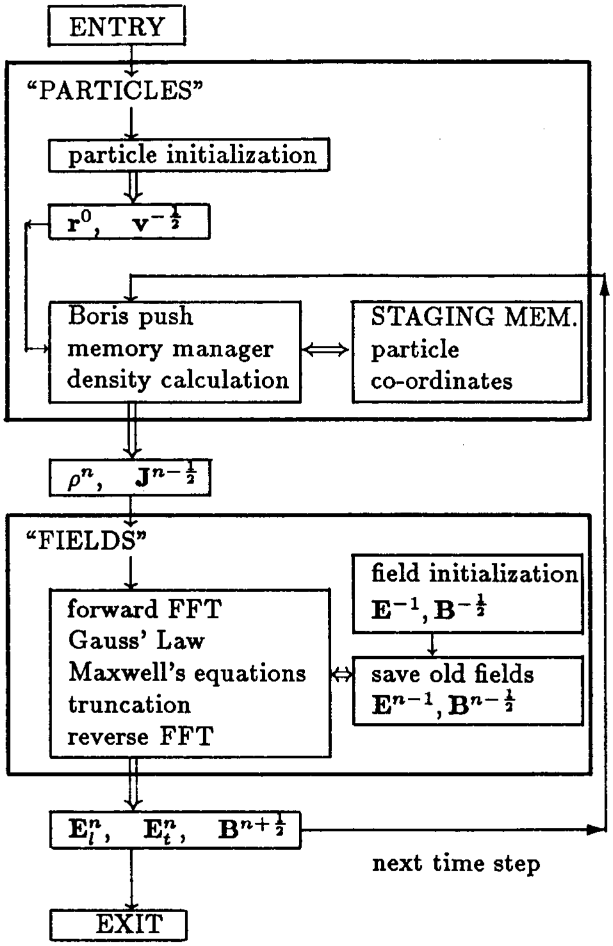

Figure 2. Flow chart for the plasma simulation algorithm.

Particle data exist in the ARU only as local variables in the procedure 'Particles', and are stored in the staging memory (SM) during the rest of the execution. The procedure contains a loop which is executed once for each plane of particle data in the SM, as follows. A 'particle plane' is moved from the SM to the ARU. Values of the magnetic field and the total electric field are interpolated in each cell at the particle positions (x, y). (Operations are masked out in cells containing no particles). New velocity components are obtained by applying the Boris push, and the position is updated with the new velocity. The new co-ordinates (x, y) are tested to see whether the particle has moved to a new cell on the simulation grid, and a moving direction D assigned accordingly. The speed of light is normalized to one cell per timestep, so that no particle can move further than to one of its eight nearest neighbours; hence there are nine possible values of D, including the null move.

An inner loop is then carried out for each direction. The particles with appropriate D are copied into a workplane which is shifted to the new location. The densities ρ and $\mathbf{J}$ are updated with the new information. Distribution functions are accumulated when needed via a cascade sum. The workplane contents are then copied to the top of the stack of particles already moved. If full planes exist in the stack, they are sent to the SM. The next SM plane is then retrieved. After the last plane has been processed, the remaining particle planes are written out to the SM to clear the ARU of particle data.

In a fully electromagnetic program, the field solver requires nearly all the available ARU space. The 1-dimensional FFT that we use is based on a straightforward Cooley-Tukey algorithm (Ref. 18), and operates on 128 elements along either rows or columns.

Equations 3.5 to 3.7 are solved in transform space using the $\mathbf{E}_l$, $\mathbf{E}_t$, and $\mathbf{B}$ values from the previous time step. The shape factor and coefficients involving $\mathbf{k}$ are recalculated at each time step to minimize the space required for global variables. Transverse fields are truncated for $\mathbf{k}$ greater than some value specified by the user to meet the CFL condition 3.9. The spatial variation of one component of the field of a single particle is shown in Figure 3.

Particle initialization is accomplished in 'Particles' by setting up co-ordinates in the ARU plane by plane. A random number generator based on a random bit-plane generator developed at NASA Goddard Space Flight Center is used to initialize velocities. Up to twelve uniformly distributed random numbers are added to produce a distribution function closely approximating a Gaussian. Field initialization takes place through the inclusion of 'old' fields in Maxwell's equations: external fields are introduced in this way.

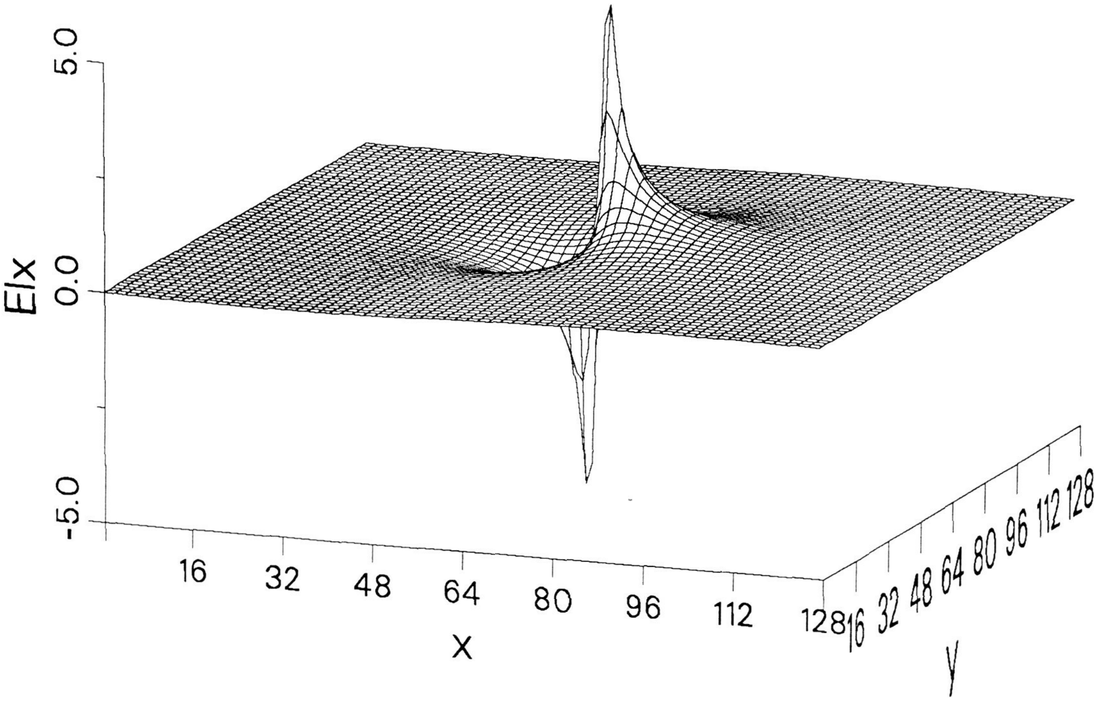

Figure 3. Spatial variation of the x-component of the longitudinal electric field due to a charge with radius 1 grid cell placed at the center of the grid. Units are arbitrary.

6. STATUS OF THE CODE

Considerable time has been spent in developing basic tools such as the FFT, the random number generator, input/output routines, etc. These are now in working order. The particle memory manager is complete and is being tested with the particle initialization. The calculation section of 'Particles', which contains the interpolation, the Boris push, the distribution function calculation and the density accumulation, is in the final stages of development. The procedure 'Fields' awaits only the debugging of the transverse field section; preliminary results have been shown in Figure 3. Thus, as we go to press, the code appears to be close to final assembly.

7. CONCLUSIONS

Since we have not yet performed a complete simulation on the MPP, it would be premature to try to give any firm answers to the four questions raised in Section 1. Nevertheless, we feel that sufficient progress has been made to allow us to express opinions as to what the answers are likely to be. For instance, it is already clear that the two physics problems we have chosen, one in plasma physics and the other in stellar dynamics, can be simulated on the MPP without posing any difficulties of principle. The main practical difficulty is likely to be that of load balancing, i.e., of ensuring that the computational work load is shared evenly between all the processors in the array. In our present computational scheme, where the simulation domain is mapped directly onto the processor array, load balancing becomes difficult whenever the physical medium is inhomogeneous. Thus our stellar dynamics problem is more difficult, in this respect, than our problem in plasma physics, as may be seen from their descriptions in Section 2. Other plasma physics problems, however, such as those involving magnetically confined plasmas, would also be subject to this difficulty, which we perceive as the main one to be overcome in order that the potential of the MPP for particle simulation of rarefied media may be fully realized. The foregoing remarks relate to the first two questions raised in Section 1.

As regards the third question, we can already suggest several features that might be incorporated into a second-generation MPP, so as to improve its capability for particle simulation. There is a clear need for a larger random-access memory (RAM) associated with each processing element (PE), particularly if some future simulations are to be made in 3-D. At present, with only 1 Kbit of RAM per PE, we would be restricted to 2-D simulations with fewer than 10 particles per spatial cell, were it not that we are also using the staging memory for storing data on the positions and velocities of the particles; however, the use of the stager for this purpose entails a speed penalty. Another measure that might be taken to increase the computing speed is to introduce some degree of parallelism into the PE arithmetic; the present capability of varying the word length by 1-bit increments would probably have to be sacrificed, but this feature is less important in particle simulation than it is in image processing.

On the fourth and last question, as to the relative merits of the MPP architecture *versus* possible alternative architectures in the application to rarefied media simulation, we are now doubtful as to whether whether our present research program will yield a clear-cut answer. At the outset, we felt that the very simple way in which our square simulation domain can be mapped directly onto the square array of PE's made the MPP architecture a natural choice, but now we are less sure, in view of the acuteness of the load-balancing difficulty mentioned above. In any case, a final decision could not be taken until after some experimentation with other architectures such as the hypercube.

These considerations govern our plans for the future. In the short term, continuing the present research program through a second year, we shall endeavor to fully realize our immediate physics and computer science objectives by exploiting the possibilities of the MPP to the utmost. The physics objectives are to achieve realistic 2-D simulations of the chosen plasma and gravitational phenomena. The computer science objectives are to improve the simulation algorithms, notably by solving for the fields by direct numerical integration of Maxwell's equations rather than by transform methods. Moreover, in order to promote load balancing, we shall investigate the use of sort/merge routines to manage the particles, permitting the data concerning them to be stored in PE's other than those that correspond directly to their spatial positions in the simulation domain. Without going so far as to store these data in arbitrary locations, we feel that it may be helpful to allow them a certain latitude. This more flexible storage scheme should certainly help to reduce load unbalance resulting from random fluctuations of particle concentration in a statistically uniform medium, such as the plasma considered in our whistler instability simulation; it is unlikely, however, to be able to cope with unbalance resulting from large-scale gradients of particle concentration, as occur in our gravitational simulation.

In the long term, more powerful solutions to the problem of load balancing will have to be found. Probably the sole viable general solution is to divide up the simulation domain with a non-uniform grid, which is then mapped onto the processor array either directly, or with a certain degree of latitude as described above. If the large-scale distribution of particles is likely to change in the course of a simulation, then the grid may have to be adaptive as well. We hope to take part in these developments, which could be tried out on the MPP in its present form. Only when this problem has been adequately dealt with can a fair comparison be made with other architectures. We anticipate that the comparison is likely to favor the MPP architecture, and to provide strong motivation for the development of a second-generation MPP specifically for particle and fluid simulation.

ACKNOWLEDGEMENTS

The material above is based upon work financed by the National Science Foundation under Grant No. PHY-8506023, with additional funding provided by the NASA Ames University Consortium under Joint Research Interchange No. NCA2-144, and by the Stanford-Ames Institute for Space Research. The NASA Office of Space Science and Applications granted us free use of the MPP at NASA Goddard Space Flight Center (GSFC). We are most grateful to these organizations for their support. We also wish to express our gratitude to the NASA Space Physics Analysis Network (SPAN) pilot project for use of networking facilities, and to the staff of the Space Data and Computing Division at NASA/GSFC, particularly in the MPP User Support Office, for the many ways in which they have facilitated our work with the MPP.

We also wish to thank Dr. Bruce Smith, at NASA Ames Research Center, for advice and encouragement on the gravitational simulations, while, at Stanford University, our thanks are due to Prof. Oscar Buneman for similar guidance with the plasma simulations, and to Mr. Shepherd Shi and Mr. Igor Dayen for programming assistance.

REFERENCES

1. Buneman, O., C.W. Barnes, J.C. Green, and D.E. Nielsen, "Principles and capabilities of 3-D, EM particle simulations", *J. Comput. Phys.*, 1980, 38, pp. 1-34.

2. Bruno, J., "Final report on the feasibility of using the Massively Parallel Processor for large eddy simulations and other computational fluid dynamics applications", NASA Ames Research Center, Report No. RIACS–TR–84.2, 1984.

3. Bharuthram, R., and S. Baboolal, "Theory and simulation of the whistler instability driven by a velocity-space anisotropy", *S. Afr. J. Phys.*, 1985, 8, pp. 45-49.

4. Storey, L.R.O., and F. Lefeuvre, "Theory for the interpretation of a random electromagnetic field in space", in *Space Research XIV*, edited by M.J. Rycroft and R.D. Reasenberg, Akademie-Verlag, Berlin, 1974, pp. 381-386.

5. Storey, L.R.O., and F. Lefeuvre, "The analysis of 6-component measurements of a random electromagnetic wave in a magnetoplasma — I. The direct problem", *Geophys. J. R. Astr. Soc.*, 1979, pp. 255-269.

6. Lefeuvre, F., and C. Delannoy, "Analysis of a random electromagnetic wave field by a maximum entropy method", *Ann. Télécommun.*, 1979, 34, pp. 204-213.

7. Delannoy, C. and F. Lefeuvre, "MAXENTWDF — A computer program for the maximum entropy estimation of a wave distribution function", *Comput. Phys. Comm.*, 1986, 40, pp. 389-419.

8. Hockney, R.W. and J.W. Eastwood, *Computer Simulation using Particles*, McGraw-Hill, New York, 1981.

9. Dawson, J.M., "Particle simulation of plasmas", *Rev. Mod. Phys*, 1983, 55, pp. 403-448.

10. Birdsall, C.K. and H.B. Langdon, *Plasma Physics via Computer Simulation*, McGraw-Hill, 1985.

11. Okuda, H., and C.K. Birdsall, "Collisions in a plasma of finite size particles", *Phys. Fluids*, 1970, 13, pp. 2123-2134.

12. Ratliff, S.T., *Computer Simulations of the Toroidal Cusp Experiment*, University of California, Los Angeles, Ph.D. Thesis, 1983.

13. Morse, R.L. and C.W. Nielson, "Numerical simulation of the Weibel instability in one and two dimensions", *Phys. Fluids*, 1971, 14, pp. 830-840.

14. Boris, J.P., *Proceedings of the Fourth Conference on Numerical Simulation in Plasmas*, Naval Research Laboratory, Washington, D.C., 1970, pp. 3-67.

15. Buneman, O., "Fast numerical procedures for computer experiments on relativistic plasmas", in *Relativistic Plasmas*, edited by O. Buneman and W.B. Pardo, Benjamin, New York, 1968, pp. 205-219.

16. Hoshino, T. and K. Takenouchi, "Processing of the molecular dynamics model by the parallel computer PAX", *Comput. Phys. Comm.*, 1984, 31, pp. 287-296.

17. Johns, T.C. and A.H. Nelson, "Particle simulation of 3D galactic hydrodynamics on the ICL DAP", *Comput. Phys. Comm.*, 1985, 37, pp. 329-336.

18. Cooley, J.W. and J.W. Tukey, "An algorithm for the machine calculation of complex Fourier series", *Math. Comp.*, 1965, 19, pp. 297-301.