JOURNAL OF COMPUTATIONAL PHYSICS 106, 147–154 (1993)

Accelerated Solution of the Boltzmann Equation

G. J. Parker

Department of Physics, University of Wisconsin, Madison, Wisconsin 53706

W. N. G. Hitchon

Department of Electrical and Computer Engineering, University of Wisconsin, Madison, Wisconsin 53706

AND

J. E. Lawler

Department of Physics, University of Wisconsin, Madison, Wisconsin 53706

Received January 7, 1992

Methods are presented for accelerating a numerical procedure for self-consistent solution of a kinetic equation and Poisson's equation in plasma simulation. The kinetic equation is solved using a propagator technique, although other approaches would also benefit from the accelerated procedure. The kinetic equation is solved in a phase space of at least one spatial variable and two velocity coordinates, (Z, V_Z, V_P). V_P is in the direction perpendicular to Z. In these variables it is possible to advance the reduced distribution function $g(Z, V_Z)$ in time very efficiently using several "short" time steps within a desired "long" step. We can then use the results of the "short" steps to find the quantities needed to calculate the change in the full distribution $f(Z, V_Z, V_P)$ during a single "long" time step. In the case studied here, the electric field and/or transition matrix in the (Z, V_Z) space are calculated from the reduced distribution, at each of C time steps of roughly one tenth of the plasma period. The full step is then for $C/10$ plasma periods, thus removing the limit on the integration imposed by the plasma period. Applications to an rf discharge are presented. © 1993 Academic Press, Inc.

I. INTRODUCTION

For some time, particle simulations, such as Monte Carlo and particle-in-cell Monte Carlo, have been an important tool for describing complex plasmas at the kinetic level [1–4]. Alternative approaches which offer some advantages over particle simulation have now become available for solution of kinetic equations, including a scheme based on propagators (Green's functions) [5–7]. The method employed here deals with densities of particles on a grid. Particle equations of motion are used to find the way the density is redistributed, but individual, discrete particles are not used. Methods in which discrete particles are followed on a grid, including those in Refs. [8–10], are distinct from this. This paper describes schemes which accelerate the solution of the kinetic equation.

The introduction of implicit methods into particle simulation has been claimed to be the major innovation in that technique in recent years [1]. The meaning of implicit in this context needs clarification. Approximate and/or iterative procedures are used for the construction of quantities such as the electric field at the end of the time step. These are then used in the calculation of the particle motion during the step. Some of the techniques used here are "implicit" in this sense, but not truly implicit.

The basic scheme will be described briefly next. In Section II the underlying propagator (Green's function) scheme is reviewed, with some details of how it is made to conserve certain quantities and how it is optimized for accuracy and efficiency. In addition, a thorough description of the accelerated method for this scheme is given. In Section III the results of applying the method to simulation of an rf discharge in helium are discussed.

In this work we are principally concerned with the time evolution of the distribution function $f = f(Z, V_Z, V_P)$ for electrons, which is calculated self-consistently with the electrostatic potential Φ. The ion distribution is also calculated, but since it varies much more slowly it can easily be followed using much longer time steps, so it presents few problems. Z is a spatial coordinate perpendicular to the plane parallel electrodes, V_Z is the corresponding velocity, and V_P is the component of velocity perpendicular to Z. In much of what follows, we exploit the fact that the spatial coordinates perpendicular to Z are ignorable.

The major expenditure of effort in the numerical calculation is associated with updating the density in each of

0021-9991/93 $5.00

the large number of cells in the phase-space mesh. The two-dimensional (2D) distribution function $g = g(Z, V_Z)$ can be found from f by integrating over V_P, namely

$$g(Z, V_Z) = 2\pi \int f(Z, V_Z, V_P)\, V_P\, dV_P. \qquad (1)$$

Of course, the integral turns to a sum on the numerical mesh.

In the absence of collisions, f and g obey the same equation of motion and may be found using the same propagator; g is found more quickly, however, because *there is no need to update the many* V_P *cells involved in* specifying f. If we compute g from f, and then separately we advance each of them in time, using the same time step Δt and calculating the potential Φ self-consistently at each step; then the results agree to numerical accuracy. That is, if g is calculated directly and Φ found from it, the results are identical to those found from f, at each step.

One set of accelerated schemes we have considered exploit the fact that the potential Φ is the same, whichever way it is found, to overcome the limit placed on the time step Δt by the plasma period T_P. In an "explicit" scheme the time step is limited to the smaller of $T_P/10$ or $T_C/5$, where T_C is the mean time between collisions for particles at a speed where they collide most frequently. Here we are interested in plasmas for which $T_P \ll T_C$.

The two main approaches we explore here are: (i) integrating g over a large number of short steps, then constructing f from the behavior of g, and (ii) the use of "damping" to increase the short time step Δt. The latter is straightforward. After each time step, an instaneous electric field $\mathbf{E}(Z, t + \Delta t)$ is found from the electron and ion distributions. We then compute a damped field $\mathbf{E}^d(Z, t + \Delta t)$ from

$$\mathbf{E}^d(Z, t + \Delta t) = \alpha \mathbf{E}^d(Z, t) + (1 - \alpha)\, \mathbf{E}(Z, t + \Delta t),$$

where $0 \leqslant \alpha \leqslant 1$. With $\alpha = 0$, the damped electric field is just the instaneous electric field, while in the other limit ($\alpha = 1$), the electric field is independent of time. We find that using $\alpha \approx 0.6 - 0.8$ is effective in reducing numerical noise.

We now return to the use of g to find f. At time t, g is computed from f; g is then advanced in time, using C steps of duration Δt such that $\Delta t \approx T_P/10$. Two procedures will be described to update f. In the simplest, at each of these steps, $\Phi(Z)$ is calculated and its values stored. After all C steps, the average potential $\langle \Phi(Z) \rangle$ is used in the basic calculation to advance f, using a time step of duration $\Delta t_{\text{long}} = C\,\Delta t \sim C T_P/10$. (In order to ensure energy conservation, particles are reallocated to velocity bins in accordance with the average potential Φ, obtained from the average electric fields found in the short time steps (see Section II below).) The advantage of this procedure is twofold. It allows f to be updated using long time steps, whereas normally the step is limited to be about $T_P/10$, thus reducing computation time. The second advantage is that the long step results in the decrease of "numerical diffusion."

The second approach to using g to find f requires that we store not $\Phi(Z)$ but rather the transition probabilities for particles going from initial cell (Z'', V''_Z) to a final cell (Z, V_Z). The overall probability during C "short" steps of going from each initial cell to each final cell is constructed by taking the transition matrix created during step $i-1$ which gives the transition probabilities after $i-1$ steps. We then update it during step i at the same time as calculating the ith update of g. This is repeated until short step $i = C$ is reached. This is costly in terms of memory, but more accurate than the first scheme. It will be described in more detail below.

The calculation of g at a single step is very fast in comparison with that of f. The actual speed achieved depends on the number of V_P values and the efficiency of the matrix manipulation. Between long steps, collisions are included.

The methodology outlined can be used to set up a Newton's method formulation of a steady-state problem. If it converges, it yields a truly implicit solution for the steady-state distribution and electric field. The next section describes the procedure in more detail, followed by results in Section III.

II. THE ACCELERATED BOLTZMANN SOLVER

In this section, we will first review the Convected Scheme for solution of the kinetic equation and then describe the justification for and implementation of the accelerated scheme. In the section that follows, we will present results of its application to an rf discharge in helium.

Propagator Solution of the Boltzmann Equation

We now review the convected scheme [5–7] for solution for the Boltzmann equation. In the normal convected scheme, the electron (as well as ion and neutral) distribution function $f(Z, V_Z, V_P)$ is advanced in time according to a propagator $p(Z, V_Z, V_p, Z'', V''_Z, V''_P, \Delta t)$ such that

$$\begin{aligned} f(Z, V_Z, V_P, t + \Delta t) &= \int f(Z'', V''_Z, V''_P, t) \\ &\quad \times p(Z, V_Z, V_P, Z'', V''_Z, V''_P, \Delta t)\, dZ''\, d\mathbf{V}''. \qquad (2) \end{aligned}$$

That is, the propagator (or Green's function) $p(Z, V_Z, V_P, Z'', V''_Z, V''_P, \Delta t)$ determines the fraction of particles moving from cell (Z'', V''_Z, V''_P) to cell (Z, V_Z, V_p) in the time step

Δt. In the case studied here, it is assumed that the distance between the two electrodes, $Z_{\max}$, is much less than the lateral dimensions of the electrodes, so that fringing of the electric field can be ignored and so the electric field $\mathbf{E}$ is only a function of position Z.

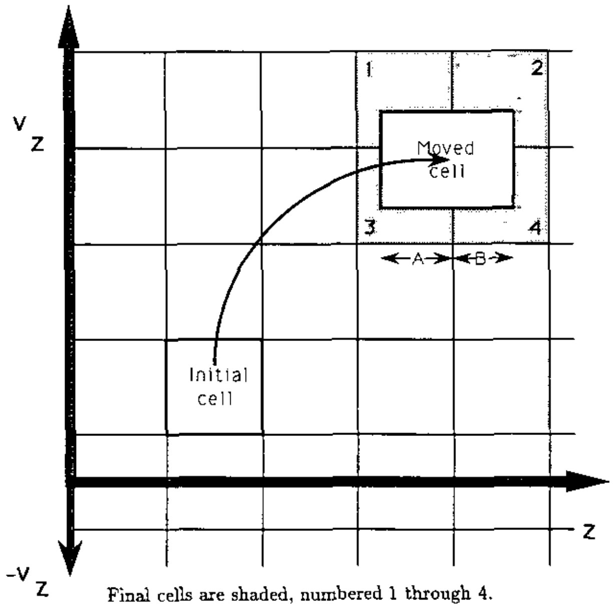
The propagator can be broken up into two distinct parts, corresponding to a ballistic motion followed by collisions. The only limits on the time step are physical limits. The time step, Δt, must satisfy the relation

$$\Delta t \ll T_C \equiv [N v \sigma_{\text{tot}}(v)]^{-1},$$

where N is the density of scatterers, v is the relative speed, and $\sigma_{\text{tot}}(v)$ is the total cross section. This criterion ensures that the probability of a scattering event taking place in Δt is very much less than one. There is also a constraint on Δt in that

$$\Delta t < T_P \equiv \frac{2\pi}{\omega_P} \equiv \left(\frac{\pi m_e}{N_e e^2}\right)^{1/2},$$

where m_e is the mass of an electron, N_e is the density of electrons, and e is the magnitude of the electron charge (in gaussian units). This ensures that the electric field $\mathbf{E}(Z)$ does not change appreciably during the ballistic move. In the case studied here, $T_P < T_C$.



Final cells are shaded, numbered 1 through 4.
A and B are spatial distances. Fractions of density to each final cell are:

$$\text{Cell 1: } \left(\frac{A}{A+B}\right)\left(\frac{\mathscr{E}^0-\mathscr{E}_3}{\mathscr{E}_1-\mathscr{E}_3}\right)$$

$$\text{Cell 2: } \left(\frac{B}{A+B}\right)\left(\frac{\mathscr{E}^0-\mathscr{E}_4}{\mathscr{E}_2-\mathscr{E}_4}\right)$$

$$\text{Cell 3: } \left(\frac{A}{A+B}\right)\left(\frac{\mathscr{E}_1-\mathscr{E}^0}{\mathscr{E}_1-\mathscr{E}_3}\right)$$

$$\text{Cell 4: } \left(\frac{B}{A+B}\right)\left(\frac{\mathscr{E}_2-\mathscr{E}^0}{\mathscr{E}_2-\mathscr{E}_4}\right)$$

Where $\mathscr{E}^0$ is total initial energy, and
$\mathscr{E}_i = \frac{1}{2} m V_{Z_1}^2 - e\Phi(Z_i)$

FIG. 1. Ballistic movement of phase space cells, in the independent variables (Z, V_Z, V_P). Left and right faces of initial cell are moved independently. Note that the V_P-axis has been omitted for clarity.

Figure 1 is a schematic of the movement of the phase space cells during the time step. The faces of the cells are moved independently according to the initial velocity and the local electric field. The number of particles going to each final spatial cell is determined by the fractional overlap of the "moved" cell with a particular final spatial cell, compared to the total spatial width of the "moved" cell. Once this is computed, the final V_Z distribution is computed not via the equations of motion, but by energy conservation. Specifically, we know that the initial total energy is

$$\mathscr{E}_{\text{init}} = \tfrac{1}{2} m_e (V_{P\text{init}}^2 + V_{Z\text{init}}^2) - e\Phi(Z_{\text{init}}) \tag{3}$$

and the final energy in a particular final cell at Z_{final} is

$$\mathscr{E}_{\text{final}} = \tfrac{1}{2} m_e (V_{P\text{final}}^2 + V_{Z\text{final}}^2) - e\Phi(Z_{\text{final}}), \tag{4}$$

where $V_{P\text{init}} = V_{P\text{final}}$. Equating these expressions, we solve for $V_{Z\text{final}}$. In general, $V_{Z\text{final}}$ does not correspond exactly to a velocity cell on the mesh. The two neighboring V_Z cells are then found, and the fraction of particles that go to each is chosen so as to ensure that the average total energy is conserved.

Figure 2 is a schematic of the collision operator [6].

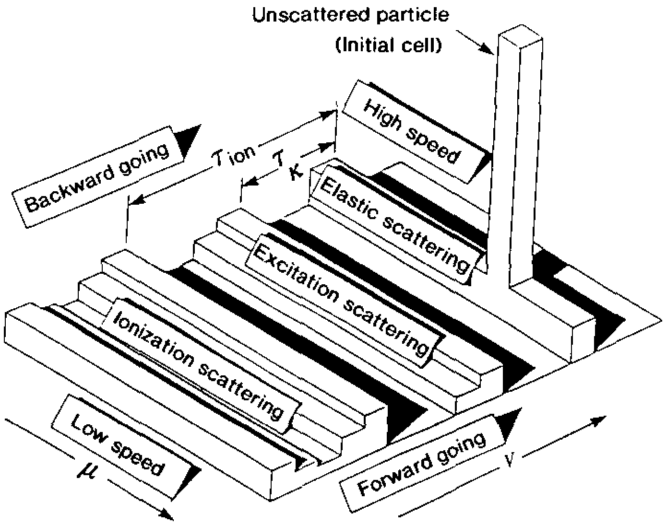


FIG. 2. Schematic of the electron collision operator. Electrons leaving a scattering event are distributed isotropically; anisotropic elastic scattering is described through the elastic momentum-transfer cross section. Electrons involved in inelastic collisions lose a fixed amount of energy according to the type of collision. μ is the cosine of the angle that the velocity makes with the Z-axis, i.e., $\mu = V_Z/(V_Z^2 + V_P^2)^{1/2}$ and $V = (V_Z^2 + V_P^2)^{1/2}$. Individual mesh cells have been omitted for clarity.

Collisions take place after the long time step, not during the accelerated part of the calculation. This is justified since the probability of colliding during any one small time step is very much less than one. It only is appreciable after many small time steps, namely a long time step.

Accelerated Propagator Scheme

The accelerated version of the scheme is now considered in detail. We begin by noting that the E-field in this scheme is damped, as described in the Introduction. Thus at each "short" step, $\mathbf{E}^d(Z, t)$ is constructed and used in the next short ballistic step.

The starting point of the justification of the accelerated scheme is the Lagrangian of the system. Since we have an approximately infinite parallel plate discharge, the electric field, $\mathbf{E}$ is only a function of Z, the distance along the discharge axis. Therefore the instantaneous electrostatic potential Φ is also only a function of Z. Specifically, the Lagrangian for an electron is given by (ignoring collisions)

$$L = \tfrac{1}{2} m_e (V_Z^2 + V_P^2) + e\Phi(Z). \qquad (5)$$

The conjugate momentum $p_P = m_e V_P$ is constant during the ballistic motion. This greatly simplifies matters, since once the coordinates (Z, V_Z) of a particular moved cell with initial location (Z'', V_Z'') is found, all V_P cells from the same initial location move to the same (Z, V_Z) coordinates. In other words, instead of finding the moved location of a cell labeled by $(Z_i, V_{Z_j}, V_{P_k})\ \forall i, j, k \in$ the mesh, we only have to compute the moved location for a given value of (Z_i, V_{Z_j}) once for, say, $V_{P_k} = 0$, since this final location is the same for all V_P at (Z_i, V_{Z_j}). The value of V_P of any cell is not changed during the ballistic motion, but it can be changed during a scattering event.

The second observation we need to make to justify the acceleration, is that up to (a fraction of) the collision time, T_C, the electron motion (and the corresponding electric field) is the same if we propagate the full 3D distribution $f(Z, V_Z, V_P)$ or if we propagate the 2D distribution $g(Z, V_Z)$. This says that we can update the 2D distribution, g, and still have the same density, and by Poisson's equation, the same electric field, as we would obtain from the 3D distribution function. A complication arises, however, in that to describe collisions we need the full 3D distribution, f.

One can easily see that updating the 2D distribution is faster than integrating the 3D distribution, since the number of cells that need to be moved is reduced from $M_z M_v (M_v + 1)/2$ to $M_z M_v$. M_z is the number of spatial bins, M_v is the total number of V_Z bins (positive, negative, and zero), and $(M_v + 1)/2$ is the number of V_P cells. (Note that the number of V_P cells can be independent of the number of V_Z cells.) The savings are somewhat more modest than this, since the extra factor of $(M_v + 1)/2$ corresponds to moving the V_P cells, which, according to the argument involving the Lagrangian, given above, are less time consuming to move. We can see, however, that the larger the mesh, the more saving in computation time that the acceleration will give.

Choice of Independent Variables

Before we describe the implementation of the acceleration, some remarks on the choice of variables (Z, V_Z, V_P) should be made. In previous CS codes, generally the variables used were (Z, V, μ), where V was the speed, and μ was the cosine of the polar angle that $\mathbf{V}$ made with the Z-axis. Such a set made the collision operator straightforward (equal numbers of particles were placed in equal intervals of μ for isotropic scattering), but introduced complexities in the ballistic motion. Specifically, none of these variables are ignorable in the ballistic motion. So, to update the distribution function, one had to move each cell face through a unique trajectory, not the same as that of any other cell faces. Furthermore, to implement an accelerated method of this kind, one would have to transform the 3D distribution to the 2D distribution, which involves extra overhead. The main advantage to using those variables is that the collision operation is straightforward. (It is also the case that in a certain sense, very fine "energy resolution" at low V_Z is obtained with these variables on a constant sized velocity mesh ($\Delta V = \text{const}$). For each V, there are a number of μ's, so the projection of V onto the Z-axis gives a range of values spaced more finely than just ΔV. However, the true resolution is still only ΔV.)

In the variables (Z, V_Z, V_P), the collision operator is more complex and fine resolution at low V_Z is not ensured but must be provided by the choice of mesh spacing. Fine resolution at low V_Z is important for accurate description of electron movement. While the collision operator is more complex in these variables, this can be overcome, as we shall now describe.

For each type of scattering event, one can compute the final energy of an electron, given its initial energy and the amount of energy lost. The details of a collision are independent of spatial location. Thus a collision operator in the form of a matrix which redistributes particle densities in velocity space is constructed once and only once at the

TABLE I

Parameters of the Discharge

Quantity	Value	Units
Neutral density	9.65	10^{15} cm^{-3}
Neutral temperature	300	K
Peak driving voltage	300	Volts
Driving frequency	13.56	MHz
Discharge length	6.7	cm

beginning of the simulation for each type of collision. Care must be taken while constructing such a matrix, since energy and number conservation must be adhered to. This matrix is discussed below.

Currently, we assume isotropic scattering for all scattering processes (anisotropic elastic scattering is described by the isotropic elastic momentum transfer cross section), but anisotropic scattering could be implemented. Isotropic scattering is easily implemented by constructing two matrices. The first matrix, for a given initial energy and type of collision, gives the final energy and redistributes particles on an intermediate (one-dimensional) energy mesh so that their average energy is correct.

The second matrix, given the final energy and assuming isotropic scattering, indicates how to place electrons back on the 3D mesh so that the final energy is correct and isotropy is ensured. Specifically, the second matrix is indexed by the final energy values on the intermediate mesh. For each energy, a maximum $V_{Z_{\max}}$ is found. V_Z runs from $-V_{Z_{\max}}$ to $+V_{Z_{\max}}$. The fraction that goes to each valid V_Z cell is just the width of the V_Z cell divided by $2V_{Z_{\max}}$, in isotropic scattering. That is, equal numbers are placed in equal intervals of V_Z/V, where V is the total speed. Then, in general, there are two adjacent V_P's that go with each of the V_Z to conserve the average (final) energy. A simpler and faster scheme which gives less resolution in pitch-angle, but less numerical diffusion in energy, has been described elsewhere [11]. For elastic scattering of electrons (neglecting energy loss, which is included later), this involves using only those final cells whose energy is identical to that of the initial cell, so angular resolution is limited, but numerical diffusion in energy is eliminated from this process. (During ionization, it is important for charge balance to add new ions at the same time as adding new electrons, even though the ions are not otherwise updated as often as electrons.)

Good resolution at low V_Z is obtained by using a nonconstant velocity mesh. Currently, we have a velocity mesh such that $\Delta(\frac{1}{2}mV_Z^2) \propto (\frac{1}{2}mV_Z^2)^x$. That is, the width of a velocity cell is such that the change of energy across one cell is proportional to the energy of that cell raised to some power x. If $x=\frac{1}{2}$ then this would be a constant velocity mesh ($\Delta V_Z = \text{const}$). We use $x=1$ here, giving a mesh with energy spacing on a logarithmic scale ($\Delta \ln(\frac{1}{2}mV_Z^2) = \text{const}$). At the lowest energies, however, $\Delta V_Z = \text{const}$ is used. In the example treated here, the low energy spacing is $\Delta V_Z = 90$ km/s (corresponding to roughly 0.025 eV) and $\Delta \ln(\frac{1}{2}mV_Z^2) = 0.287$ at higher energies, the latter spacing providing resolution $\Delta(\frac{1}{2}mV_Z^2) = \pm 14\%$ and $\Delta V = \pm 7\%$.

Implementation of the Accelerated Propagator Scheme

Now we wil describe in some detail the implementation of the acceleration. There are many differences in detail in the possible algorithms. Two methods based almost directly on the above argument are described here. One stores $\Phi(Z)$ from the short steps to update f. The second creates the overall $(Z'', V''_Z) \rightarrow (Z, V_Z)$ transition matrix while updating g.

The first method is straightforward. At the beginning of the ballistic move, we compute $g(Z, V_Z)$ from the current $f(Z, V_Z, V_P)$. Then we begin to take short time steps of $\Delta t \approx \frac{1}{10}T_P$. We propagate the 2D distribution, g, for some number of steps C, such that $C\,\Delta t \leqslant T_C/5$, keeping track of the intermediate electric fields, $\mathbf{E}_i$ computed by means of Poisson's equation ($\nabla^2\Phi = -4\pi\rho^{\text{net}}$) from the 2D electron distribution, g, and the ion distribution. After C short time steps, a time average electric field is calculated at each spatial location and used to update the full 3D distribution, f, in a time step of $\Delta T = C\,\Delta t$. While this is straightforward to implement, it is found that typically C may only have values $\leqslant 5$ or so. If C is more than five, then numerical instabilities occur. Namely, the ionization rate tends to grow without bound and the electron density grows. It is also observed that if $C\,\Delta t$ is an integer divisor of T_P, then resonance occurs and again, unphysical instabilities occur.

We now outline the matrix-update procedure. Here, instead of storing the intermediate values of $\Phi(Z)$ from the short steps, we construct a matrix which gives the probability of each initial cell going to a set of final cells. Specifically, the matrix is indexed by the ordered pairs (Z, V_Z). If we call the matrix A, then the element $A_{i,j}$ gives the fraction of particles in initial cell j (which corresponds to (Z'', V''_Z)) going to the final cell i (corresponding to (Z, V_Z)). For each short time step, such a matrix is created as g is updated. Then by doing the C matrix multiplications, we find the final transition matrix which describes how the initial 3D cell's densities change throughout the "long" step. By noting that all the different V_P cells with a certain (Z'', V''_Z) all transform the same way, we can then update the full 3D distribution from just a matrix–vector multiplication.

Storing all the intermediate transition matrices is prohibitive due the large memory requirements. Instead, during the update of g, an *outer-product* is done between a column vector of transition matrix elements in the current 2D move (which consists of the group of final cells for a particular initial cell of the 2D move) and a row of the previous transition matrix (the row corresponding to the initial cell of the overall move, whereas the column corresponds to the final cells). This is added to the running sum of the new transition matrix. Only two matrices are required regardless of the number of short steps.

The matrix solution, as described above, is exact. With efficient matrix-handling capabilities, this offers an increase in speed. Alternatively, approximate matrix solutions, which refer to a fraction of the phase-space mesh, can be used. For example, we can simplify the updating of the V_P distribution $h(V_P)$ at a given V_Z, by assuming that $h(V_P)$ is

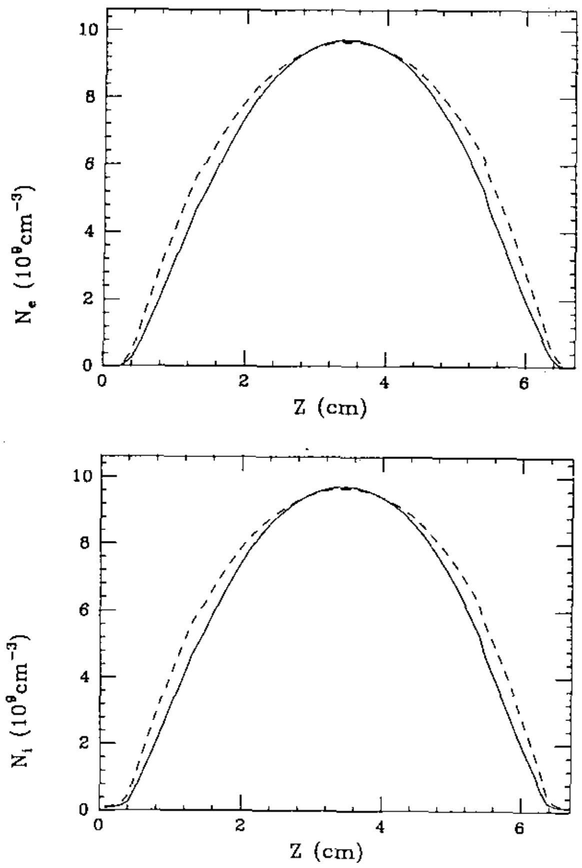


FIG. 3. Time averaged electron, N_e, and ion, N_i, densities; solid lines indicate accelerated results, dashed lines indicate "basic" results.

the same at all $|V_Z|$ values greater than $V_{Z_{\max}}$, for some suitable value of $V_{Z_{\max}}$. The matrix need not be evaluated for $|V_Z| > V_{Z_{\max}}$, in this case. This approach was used in the reults shown in Figs. 3–7. In addition, very small terms in the transition matrix were dropped. Depending on the application, the transition matrix elements from some cells need not be calculated but may be interpolated from those of adjacent cells, further reducing the computational effort. In our calculations, with no special capability to perform matrix manipulations, but using a "linked-list" representation to store the transition matrix, "breakeven" in terms of the speed came when the matrix was evaluated for between $\frac{1}{3}$ and $\frac{1}{2}$ of the phase-space cells. When $\frac{1}{3}$ of the cells were used, a 50% increase in speed resulted. The maximum possible in our case is a fivefold increase.

Various techniques are used to speed the calculation of the evolution of the 2D distribution. Since there are no additional operations between successive moves using g (other than calculation of an intermediate electric field), we store the 2D distribution in a three-dimensional matrix with indices given by (Z_i, V_{Z_i}, *Current*), where *Current* is an index with value 1 or 2. By toggling the index, one can avoid reading the new values of the distribution into the matrix after each move, to replace the old, as one would have to do if *Current* were not there.

A second time-saving technique involves the calculation of the motion of the cell faces. We move each of the faces independently according to the local electric field. Now, the left face of a given cell has the same motion as the right face of the cell with the same initial V_Z in the spatial location just left of the original cell. We therefore store the final spatial locations of the right faces of the cells with coordinates (Z_k, V_Z), to be used for the left faces of cells with coordinates (Z_{k+1}, V_Z). Since finding the final location of the cell faces can take appreciable time, we achieve a factor of two in computation time, compared to moving all the faces separately.

The accelerated time step is typically three or four times longer in the first method and up to six times longer in the matrix method than the basic step in the case we present below. This reflects the sizes of T_C and T_P; if $T_C \gg T_P$, a much greater acceleration results.

As an alternative to the above accelerated schemes, a Newton's method would be employed in steady state, with constraints that the change in density in every phase-space cell and the change in electric field at each spatial cell are zero. The independent variables are the density in all phase-space cells and the electric field at all spatial cells. Then the above constraints may be expressed as $H(u) = 0$. The Jacobian matrix $\mathscr{J} = [\partial H/\partial u]$ can readily be set up, during the calculations which are normally done in order to take a time step. At each iteration of the Newton's method, $\mathscr{J}$ must be re-evaluated, since some of the smaller derivatives are capable of changing significantly. Sparse matrix handling methods must be used to solve the resulting system.

Finite Mesh Effects in an Energy Conserving Scheme

We now consider the numerical accuracy of a Boltzmann solution on a mesh, and in particular of the CS on our mesh.

In any calculation which employs a mesh, the size of the mesh imposes a limit on the accuracy of the results. In this section we consider the interaction of the mesh spacing ΔZ in the spatial coordinate Z, and the spacing ΔV_Z in V_Z, with the requirement of energy conservation.

As explained above, the distance moved is $\delta z = V_Z \Delta t + \frac{1}{2} a \Delta t^2$, where a is the acceleration, $-eE/m_e$, for electrons. For simplicity, the electric field is assumed to be constant along the trajectory of the electron. However, energy conservation is used to determine the final value of V_Z, that is, the V_Z cells in which particles are placed and the fraction of particles placed in each. In particular, particles which stay in the same spatial cell at $t + \Delta t$ as they were at time t, will have the same V_Z, since $\Phi(Z)$ is not changed until the densities are updated and V_P is not changed in collisionless motion. This is not unique to our approach, of course. Any mesh-based technique which has energy conservation will have this feature.

If V_Z does not change until particles leave the initial cell, the question arises as to how big an error this causes. The time to cross the cell is roughly $t_{\text{cross}} = \Delta Z/V_Z$, and the change in V_Z which is not accounted for is $\Delta V_Z = at_{\text{cross}} = (-e/m_e)\, E\, \Delta Z/V_Z$. The fractional error in V_Z is thus $\Delta V_Z/V_Z = -eE\,\Delta Z/m_e V_Z^2$, so provided $eE\,\Delta Z \ll mV_Z^2$, twice the kinetic energy due to the "Z" motion, the error is small. This criterion is easily satisfied in the bulk region and marginally satisfied at the electrode surfaces, with $\Delta Z = 10^{-3}$ m in the cases presented here. If the criterion is not met, the time to cross the cell is overestimated, artificially lowering the mobility.

When particles enter another spatial cell, energy conservation gives the correct V_Z. Provided ΔV_Z, the spacing of cells in V_Z, is small compared to V_Z itself, V_Z is adequately resolved. At each time step there is still numerical diffusion in V_Z as well as Z due to the finite size of ΔZ and ΔV_Z, however. Further, there are always cells of the mesh with energy well below the average. For these low energy cells, at_{cross}/V_Z is large. As the time step increases, fewer particles will remain in the initial cell and, consequently, return to their initial velocity. Bigger time steps decrease this effect and also decrease numerical diffusion.

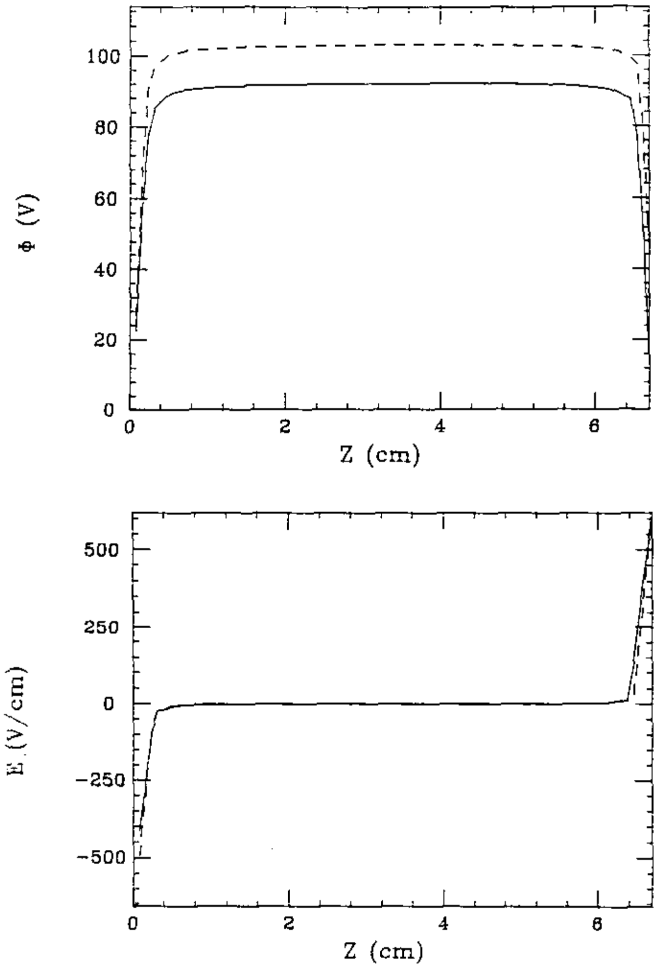


FIG. 4. Instantaneous electrostatic potentials and electric fields at zero applied voltage.

III. RESULTS FOR AN RF-DISCHARGE IN HELIUM

We now present results for a He rf-discharge and conclusions. The case considered is at a helium pressure of 300 mtorr, with peak plasma density of about $9.7 \times 10^{10}\,\text{cm}^{-3}$. In this run the long step was $\Delta t_{\text{long}} = 4T_P/13$ for the first method, and $\Delta t_{\text{long}} = 6T_P/10$ in the matrix method, which ensure that $\Delta t_{\text{long}} < T_C/5$. The first method gave approximately a factor of two increase in speed. The matrix approach, which neglected high V_Z cells in the transition matrix, is very machine dependent, but in the case examined here it gives an increase of between 1.5 and 5. The results of using this longer step are shown in Figs. 3–7.

We now compare the results of the non-accelerated scheme (with a damped field, referred to as the "basic" scheme) to the reduced matrix approach. (The first accelerated scheme gives similiar results to the matrix approach within the error of the simulation.) It is interesting to compare the discharge parameters predicted by the two calculations described above. As can be seen, the densities are only slightly altered, (Fig. 3), the bulk region being somewhat narrower in the reduced matrix case. The electric field and potential are changed very slightly, since the low field bulk region has contracted, (Fig. 4). The biggest difference is in the ionization rate S (Fig. 5), which has dropped by about 25%. This in turn leads to a slight drop in average electron energy $\frac{3}{2}k_B T_e$ (Fig. 6), since the product of density N_e and $\frac{3}{2}k_B T_e$ is expected to scale with ionization rate S, according to a simple ambipolar diffusion model.

In summary, techniques have been discussed for speeding up solutions of the Boltzmann equation. Exploiting the

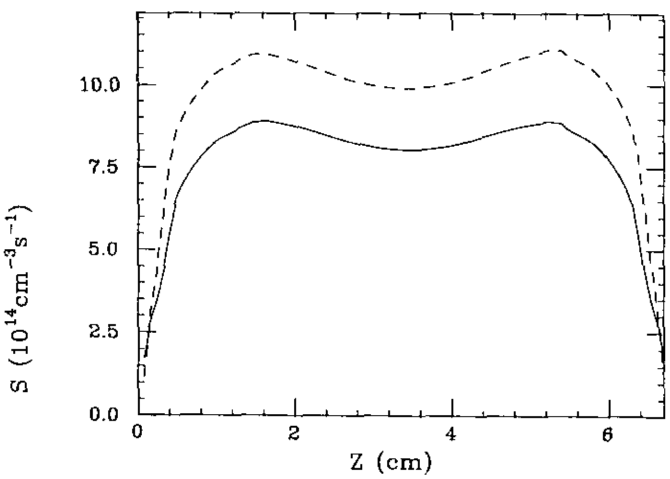


FIG. 5. Time-averaged ionization rates per unit volume, S.

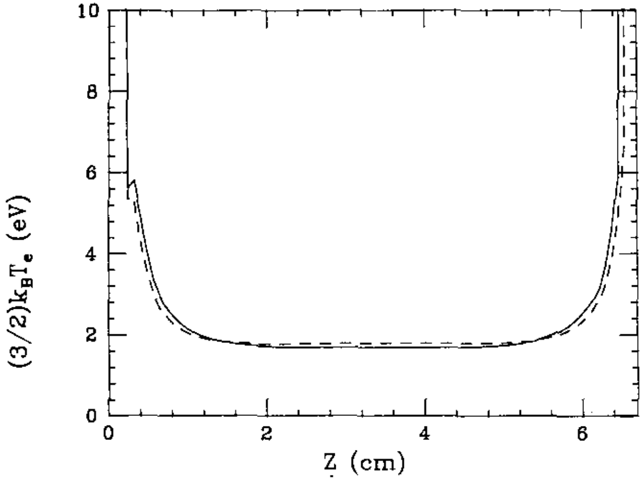


FIG. 6. Time averaged mean electron energy, $\frac{3}{2}k_B T_e$.

ignorability of the perpendicular direction, in the variables (Z, V_Z, V_P), was extensively examined (in addition to damping the electric field). Different schemes were identified, offering reasonable accuracy, with speed increases in the range 1.5 to 5. Similar techniques could also be devised in systems with two or more spatial coordinates. A truly implicit method was outlined for steady state problems, employing a Newton's method.

A detailed examination of the numerical formulation employed here has been given elsewhere, emphasizing the ways in which numerical errors should be handled [11]. The main sources of error were outlined here. It is shown in [11] how they can be reduced so that they do not effect the solution to the same problem treated here, involving an rf discharge. Applications to an rf discharge were presented, and the small differences between the accelerated and "basic" results were discussed.

REFERENCES

1. A. B. Langdon, B. I. Cohen, and A. Friedman, *J. Comput. Phys.* **51**, 107 (1983).
2. R. J. Mason, *J. Comput. Phys.* **41**, 233 (1981).
3. J. Denavit, *J. Comput. Phys.* **42**, 337 (1981).
4. J. U. Brackbill and B. I. Cohen (Eds.), *Multiple Time Scales* (Academic Press, Orlando, FL, 1985).
5. W. N. G. Hitchon, D. J. Koch, and J. B. Adams, *J. Comput. Phys.* **83**, 79 (1989).
6. T. J. Sommerer, W. N. G. Hitchon, and J. E. Lawler, *Phys. Rev. A* **39**, 6356 (1989).
7. T. J. Sommerer, W. N. G. Hitchon, R. E. P. Harvey, and J. E. Lawler, *Phys. Rev. A* **43** (8), 4452 (1991).
8. J. W. Eastwood, *Comput. Phys. Commun.* **44**, 73 (1987).
9. A. Robert, *Atmos. Oceans* **19**, 35 (1981).
10. J. R. Bates, F. H. M. Semazzi, R. W. Higgins, and S. R. M. Barros, *Mon. Weather Rev.* **118**, 1615 (1990).
11. W. N. G. Hitchon, G. J. Parker, and J. E. Lawler, to be published in *IEEE Trans. Plas. Sci.*