Towards a More Accurate Flux Corrected Transport Algorithm

E. E. KUNHARDT AND C. WU

Weber Research Institute,
Polytechnic Institute of New York, Route 110, Farmingdale, New York 11735

Received July 1, 1985; revised April 11, 1986

An algorithm is discussed for solving first-order hyperbolic equations whose solution may exhibit very steep and changing gradients (shock-like) and large dynamic range. In developing the algorithm, emphasis has been placed on obtaining more accurate solutions. The algorithm is based on Flux Corrected Transport techniques. © 1987 Academic Press, Inc.

I. INTRODUCTION

The work presented in this paper has been motivated by a need to simulate the expansion of an ionized region of a neutral gas due to drift, diffusion and ionization growth; that is, the propagation of ionizing potential waves [1]. This problem can be modelled by the set of equations [2]

$$\partial_t n_e + \nabla \cdot (n_e \mathbf{v}_e) = \nu_i n_e \tag{1}$$

$$\partial_t n_i + \nabla \cdot (n_i \mathbf{v}_i) = \nu_i n_e \tag{2}$$

$$\nabla \cdot \mathbf{E}_{sc} = \frac{e}{\varepsilon_0} (n_i - n_e), \tag{3}$$

where

$$n_e \equiv \text{electron density}$$

$$n_i \equiv \text{ion density}$$

$$\mathbf{v}_e, \mathbf{v}_i \equiv \text{electron and ion fluid velocities}$$

$$\nu_i \equiv \text{ionization rate}$$

$$\mathbf{E}_{sc} \equiv \text{space-charge electric field.}$$

The fluid velocities and the ionization rate are, in general, nonlinear functions of the local electric-field-to-neutral-density ratio, E/N ($\mathbf{E}$ is the total electric field which is the sum of the applied field, $\mathbf{E}_0$, and $\mathbf{E}_{sc}$). These functional forms are assumed to be known [2]. Ionization growth is caused by electron impact (a local process), and by photoionization (non-local process). This last process has not been included in Eqs. (1) and (2) (see, for example, Ref. [3] for details on how to do so).

0021-9991/87 $3.00

Equations (1) and (2) are the continuity equations for the electron and ion densities, respectively, and Eq. (3) is the differential form of Gauss' law. There has been a considerable and continuous interest in the numerical solution of equations of the above type. Recently, Kunhardt and Williams [4] presented a technique for solving Eq. (3). We refer the reader to that paper and references therein for discussions of numerical techniques that can be used to solve Eq. (3). In this paper we focus on techniques for solving first-order hyperbolic equations, that is, Eqs. (1) and (2).

The work on the solution of equations of this type has been mainly connected with compressible gas dynamics [5–13]. A major difficulty encountered in this area is that of accurately simulating the evolution of shock and contact discontinuities which can develop in the flow. This problem is most severe when finite difference techniques are used. When first-order differencing is used, sharp gradients in the solution tend to be reduced by dissipation; moreover, when higher-order schemes are used, the solution develops dispersive ripples near the sharp gradients [14].

A number of finite difference schemes have been devised to address this problem [5, 8–13]. Since these schemes are not tailored to compressible gas dynamics, they can be used to numerically solve Eqs. (1) and (2). However, greater demands are placed on the accuracy of the numerical method when considering the coupled set of Eqs. (1)–(3). This is because small errors in the computation of the electron and ion densities result in errors in the space-charge density and consequently the space-charge field. Since the ionization rate is an exponential function of the local electric field, subsequent calculations of the electron and ion densities may be greatly in error. This is particularly a problem near the boundaries of the electron and ion distributions (the sheaths) where large electric fields exist, and ionization growth of the charged particle densities is large [3, 15–17]. These densities exhibit very steep gradients (shock-like behavior) and change by orders of magnitude in the sheaths. For example, the electron density in the sheath of an avalanche in N_2 at atmospheric pressures and $E_0 = 50\,\text{kV/cm}$ can vary from 10^0 to $10^{15}\,\text{cm}^{-3}$. The ionization rate v_i, in the sheath may be as high as $7 \times 10^{10}\,\text{sec}^{-1}$. Because of the strong ionization growth, accuracy must be maintained over a very large dynamic range of densities. Moreover, ripples in the solution cannot be tolerated, since they are amplified with time by ionization feedback.

We found it necessary to develop a numerical scheme that can be used to solve Eqs. (1) and (2) when the solution has very steep and changing gradients (shock-like) and large dynamic range. The scheme uses the Flux Corrected Tranport (FCT) method [5–7, 11] as a basis. This method was chosen because of versatility and guaranteed monotonicity in the particle densities.

In the next two sections, we discuss the FCT scheme. In Section IV, we demonstrate its capabilities by solving the initial value problem for a 1-D first-order hyperbolic equation, namely,

$$\begin{aligned} \partial_t n + \partial_x(nu) &= 0 \\ n(x, 0) &= g(x), \end{aligned} \tag{4}$$

where n and u are the fluid density and velocity, respectively. The velocity is space-time dependent, either explicitly or through a dependence on n. In either case, the functional form of u is assumed to be known. Results are presented for two different initial density distributions ($g(x)$) and fives choices for u. Comparisons are made with results obtained using the FCT algorithm of Zalesak (FCTZ) [11] and upwind FCT (UFCT) [12]. Some final comments are given in Section V.

II. The Flux Corrected Transport Technique

The essence of the numerical scheme we have developed is the FCT method introduced by Boris and Book [5–7] and later generalized by Zalesak [11]. We refer the reader to their papers for a more complete description of the FCT. The FCT version we have used is that of Zalesak [11]. This algorithm (FCTZ) is described below.

Assume for simplicity that the x-domain is divided into cells of equal width, Δx. Take n_i and u_i to be the density and fluid velocity at the center of the ith cell. At time $m\,\Delta t$, these variables are known on the grid. The objective is to compute the density one time-step later, n_i^{m+1}, by numerically solving Eq. (4) (the superscript refers to the time step). This is accomplished as follows.

First, a "diffused" density is calculated at $(m+1)\,\Delta t$ using a low-order scheme for computing the spatial derivative. This step must not introduce ripples into the solution. An example of a low-order scheme is the donor-cell method. Using this method, the "diffused" density, n_i^{D}, is obtained from the expression

$$n_i^{\mathrm{D}} = n_i^m - \frac{\Delta t}{\Delta x}(f_{i+1/2} - f_{i-1/2}), \tag{5}$$

where

$$f_{i+1/2} = \begin{cases} u_{i+1/2}^m n_i^m, & u_{i+1/2}^m \geqslant 0 \\ u_{i+1/2}^m n_{i+1}^m, & u_{i+1/2}^m < 0 \end{cases}$$

and

$$u_{i+1/2}^m = \frac{u_i^m + u_{i+1}^m}{2}.$$

$f_{i+1/2}$ is the "low-order flux" at the boundary between the i and $i+1$ cells.

Second, a higher-order scheme is used to compute the spatial derivative in Eq. (4), i.e., using "high-order fluxes." Using the Leapfrog–Trapezoidal scheme [14] to obtain the higher-order fluxes, an intermediate density is calculated with the expresson

$$n_i^{\mathrm{int}} = n_i^{m-1} - \frac{2\,\Delta t}{\Delta x}(F_{i+1/2} - F_{i-1/2}),$$

where, for example, in fourth order

$$F_{i+1/2}(f)=\frac{7}{12}(f_{i+1}+f_i)-\frac{1}{12}(f_{i+2}+f_{i-1}).$$

The high-order flux $F^{\mathrm{H}}_{i+1/2}$ is then obtained as follows,

$$F^{\mathrm{H}}_{i+1/2}=F_{i+1/2}(f^S),$$

where

$$f^s_i=n^s_i u^m_i$$

and

$$n^S_i=\frac{n^m_i+n^{\mathrm{int}}_i}{2}.$$

The density profile computed using the higher-order fluxes, $F^{\mathrm{H}}_{i+1/2}$, contains ripples (due to dispersion).

In the final step of the FCTZ, the high-order fluxes and the diffused solution, n^{D}_i, are used in conjunction with a nonlinear filter to obtain the density profile at time $(m+1)\,\Delta t$. The objective is to adjust the fluxes in and out of a cell so as to decrease the diffusion introduced by the low-order scheme without introducing ripples into the density profile. The density at $(m+1)\,\Delta t$ is obtained from the equation

$$n^{m+1}_i=n^{\mathrm{D}}_i-\frac{\Delta t}{\Delta x}\left[A^c_{i+1/2}-A^c_{i-1/2}\right], \tag{6}$$

where

$$A^c_{i+1/2}=C_{i+1/2}A_{i+1/2}$$

with

$$0\leqslant C_{i+1/2}\leqslant 1 \qquad \text{and} \qquad A_{i+1/2}=F^{\mathrm{H}}_{i+1/2}-f_{i+1/2}.$$

$A_{i+1/2}$ is the diference between the high- and low-order fluxes. As can be seen from Eq. (6), $A_{i+1/2}$ corresponds to the maximum amount of "anti-diffusion" flux available for reducing the diffusion introduced by the low-order scheme. The coefficients $C_{i+1/2}$ determine the amount of anti-duffision fluxes to be used. In Zalesak's version of the FCT [11], the fluxes $A^c_{i+1/2}$ are determined as follows. If $A_{i+1/2}(n^{\mathrm{D}}_{i+1}-n^{\mathrm{D}}_i)<0$ and either $A_{i+1/2}(n^{\mathrm{D}}_{i+2}-n^{\mathrm{D}}_{i+1})<0$, or $A_{i+1/2}(n^m_i-n^m_{i-1})<0$ then

$$A_{i+1/2}=0.$$

The purpose of this criterion is to ignore any anti-diffusive flux that tends to decrease the slope of continuously increasing or decreasing portions of the density profile. That is, the high-order fluxes can only steepen a profile. The coefficients $C_{i+1/2}$ are obtained from the recipe

$$C_{i+1/2} = \begin{cases} \min(R_{i+1}^{+}, R_i^{-}) & \text{if} \quad A_{i+1/2} \geqslant 0 \\ \min(R_i^{+}, R_{i+1}^{-}) & \text{if} \quad A_{i+1/2} < 0, \end{cases} \tag{7}$$

where

$$R_i^{+} = \begin{cases} \min(1, Q_i^{+}/P_i^{+}) & \text{if} \quad P_i^{+} > 0 \\ 0 & \text{if} \quad P_i^{+} = 0, \end{cases} \tag{8a}$$

$$R_i^{-} = \begin{cases} \min(1, Q_i^{-}/P_i^{-}) & \text{if} \quad P_i^{-} > 0 \\ 0 & \text{if} \quad P_i^{-} = 0, \end{cases} \tag{8b}$$

and

$$P_i^{+} = \max(0, A_{i-1/2}) - \min(0, A_{i+1/2}) \tag{9a}$$

$$Q_i^{+} = (n_i^{\max} - n_i^{\mathrm{D}}) \frac{\Delta x}{\Delta t} \tag{9b}$$

$$P_i^{-} = \max(0, A_{i+1/2}) - \min(0, A_{i-1/2}) \tag{9c}$$

$$Q_i^{-} = (n_i^{\mathrm{D}} - n_i^{\min}) \frac{\Delta x}{\Delta t} \tag{9d}$$

with

$$n_i^{\max} = \max(n_{i-1}^{a}, n_i^{a}, n_{i+1}^{a}) \tag{10a}$$

$$n_i^{\min} = \min(n_{i-1}^{b}, n_i^{b}, n_{i+1}^{b}) \tag{10b}$$

$$n_i^{a} = \max({}_i^{m}, n_i^{\mathrm{D}}) \tag{10c}$$

$$n_i^{b} = \min(n_i^{m}, n_i^{\mathrm{D}}). \tag{10d}$$

The + and − superscripts refer to fluxes into and out of a cell, respectively. Equations (10) define upper ($n_i^{\max}$) and lower bounds ($n_i^{\min}$) for the density in the ith cell at time $(m+1)\,\Delta t$. These bounds are determined from the densities at time $m\,\Delta t$ and the "diffused" densities in the $i-1$, i, and $i+1$ cells. This criterion has been selected to prevent the generation of ripples by requiring that local maxima or minima not be created. Thus, the maximum anti-diffusion fluxes that can be used (Q_i^{+} and Q_i^{-} in Eqs. (9)) are then obtained from the difference between the bounds and the diffused density at time $(m+1)\,\Delta t$. Note that these fluxes cannot be greater than the maximum anti-diffusion fluxes available, namely, the A_i's. The R_i's as defined in Eqs. (8) insure this condition. They represent the fraction of the total anti-diffusion flux available in the ith cell that is to be used in correcting n_i^{D} without generating ripples in the final solution. This fraction is defined in each cell. At the boundary between two adjacent cells, i and $i+1$, only one of the anti-diffusion fluxes defined in each of the cells is allowed to flow. The smaller of the two anti-diffusion fluxes is selected. This criteria leads to Eq. (7) for $C_{i+1/2}$.

III. A More Accurate FCT

The difficulty in the numerical solution of Eq. (4) lies in the computation of the second term. The problem is in defining the correct fluxes at the cell boundaries. The more important errors that an be made in calculating these fluxes are illustrated below:

(a) Interpolation of the product of n and u. Consider the triangular density distribution shown in Fig. 1. The actual flux $f_{i+1/2}$ at the boundary between the i and $i+1$ cells is $n_{i+1/2}u_{i+1/2}$. Assuming $u=n$ (i.e., the inviscid Burgers' equation),

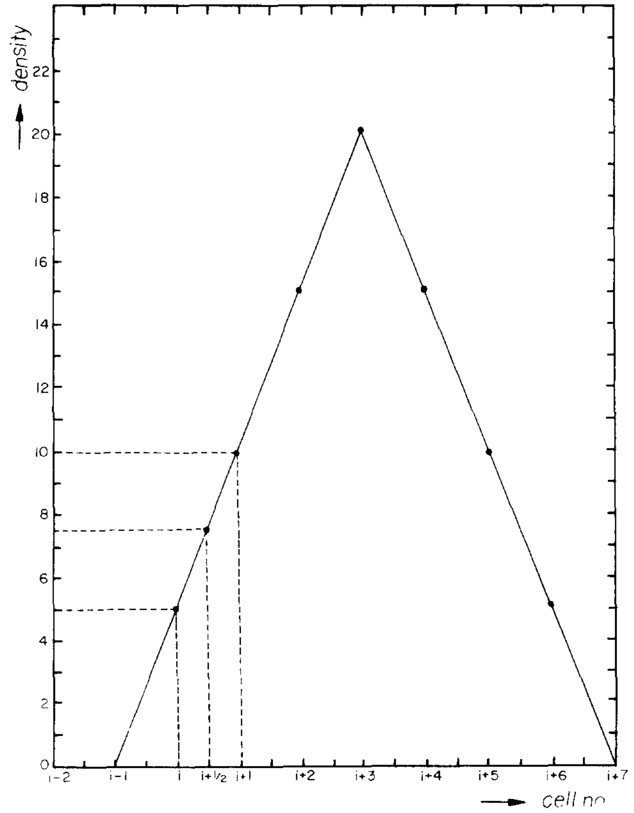

Fig. 1. Illustration of the dependence of the flux at the boundary of a cell on the interpolation of the product "nu."

this flux equals 56.25. Interpolating the product nu, and using second-order central differencing,

$$f_{i+1/2}=\frac{10^2+5^2}{2}=62.5,$$

for second-order upwind difference,

$$f_{i+1/2}=1.5\times 5^2-0.5\times 0^2=37.5,$$

and the accurate $f_{i+1/2}=56.25$. Although the FCTZ flux-limiter (Eq. (7)) can correct the error incurred by the high-order scheme, this places an undue burden on the limiter and may result in the wrong limits in the next time step. By interpolating n and u independently and then taking the product, the accuracy of the high-order scheme is greatly improved. That is, by obtaining values for the density and velocity at the boundary, we can more accurately calculate the mass flow through the boundary.

(b) Very steep gradients with large dynamic range. Consider the density distribution sketched in Fig. 2, and assume that u is constant and equal to 1. Using second-order central difference, the fluxes at the boundary

$$f_{i+1/2}=\left(\frac{10^{10}+10^9}{2}\right)u=5.05\times 10^9$$

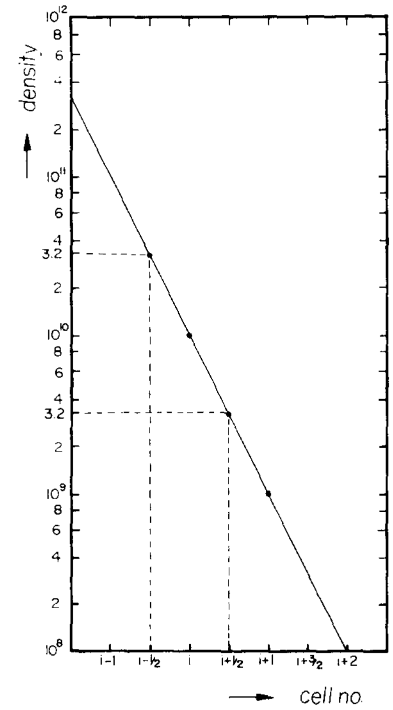

FIG. 2. Illustration of the consequence of profiles with large dynamic range on the computation of the flux at a cell boundary.

and

$$f_{i\quad 1/2}=\left(\frac{10^{11}+10^{10}}{2}\right)u=5.05\times 10^{10}.$$

Fourth-order central difference yields

$$f_{i+1/2}=\frac{7}{12}(10^{10}+10^{9})-\frac{1}{12}(10^{11}+10^{8})=-\frac{3}{12}10^{10}.$$

This flux is high and negative. Thus, for very steep gradients, higher-order schemes introduce more error in the solution. Moreover, the flux-limiter in FCTZ allows the density in cell i to ecrease to the value of the density in the $i+1$ cell. This results in a staircase front (see next section). After the formation of the staircase, the flux-limiter can no longer prevent the formation of ripples. In two dimensions, FCTZ inherently allows the formation of ripples. These ripples are caused by the limiter used in FCTZ. Because of this, we suggest the use of time-splitting techniques to avoid these inherent ripples in the solution, although the staircase behavior would still be present.

Using upwing differencing, the flux at the boundary is

$$f_{i+1/2}=1.5\times 10^{10}-0.5\times 10^{11}=-3.5\times 10^{10}.$$

This also is not the correct flux. When this differencing scheme is used with Zalesak's flux-limiter, the results are again incorrect and also lead to staircase formation. Although the Boris and Book [5] limiter has more numerical diffusion than Zalesak's, it may also lead to staircase formation (see next section).

The problem with both of these "low-order" limiters is that they use the values of the densities at the $i-1$ and $i+1$ cells to determine the upper and lower bounds for the density on the ith cell (for example, see Eqs. (10)). In very steep gradients and when $u\,\Delta t\leqslant\Delta x/2$, these values are too far removed from the density in the ith cell to serve as accurate limiters. They introduce too much "play" in the value of the density in the ith cell. Since the errors introduced at each time step may compound, this "play" in the density is intolerable for long simulation times.

(c) Constant density in a cell. When the gradients are large, the density variation in the cell may have to be taken into account to compute the correct fluxes. As illustrated in Fig. 3, the flux out of the ith cell is greater than it should be. This tends to steepen the font if one uses Zalesak's limiter. Moreover, for large gradients, one must differentiate between the average density in a cell $\bar{n}_i$ and the density at the center of the cell, n_i, since each plays a different role. This role depends on the interpolation scheme and the computation of the fluxes at cell boundaries.

(d) Interpolation technique. The method used for interpolating the wavefront becomes very important when the gradients are steep. In general, high-order Lagrangian interpolating schemes tends to steepen the profile when used with

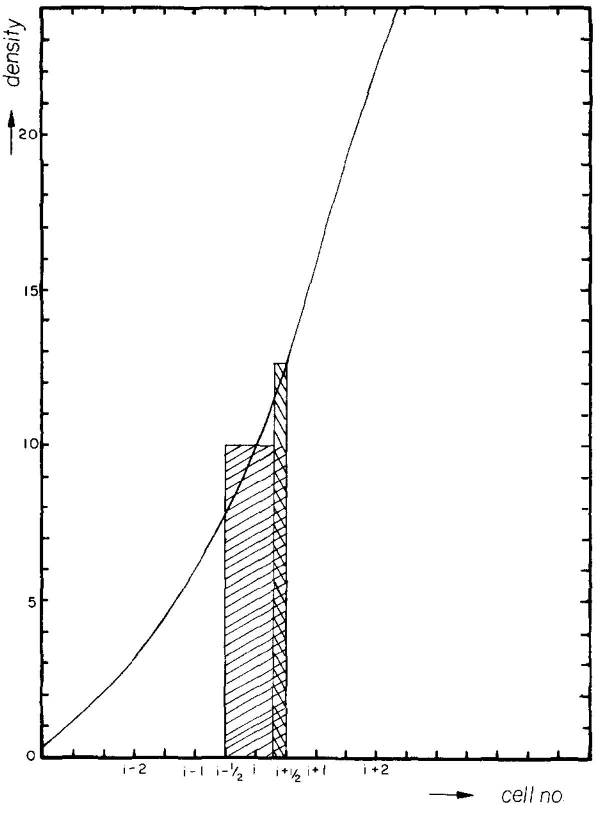

FIG. 3. Illustration of the consequence of assuming constant density in cell on the calculation of the flux at a cell boundary.

Zalesak's flux-limiter. For example, assume u is constant and consider the profiles sketched sketched in Fig. 4 at two time steps, Δt apart. In second order, $f_{i+1/2} = 5 \times 10^9 u$ and $f_{i-1/2} = 5 \times 10^8 u$. Let $u\,\Delta t = 0.2$ and $\Delta x = 1$. After Δt, the density in cell i is given by

$$n_i^{t+\Delta t} = n_i^t - \frac{\Delta t}{\Delta x}(f_{i+1/2} - f_{i-1/2}) = 10^9 - 0.2(5 \times 10^9 - 5 \times 10^8) \approx 10^8.$$

$n_i^{t+\Delta t}$ is much smaller than it should be. Since 10^8 is greater than 8×10^7 (the lower bound on the density), the FCTZ limiter allows the density to decrease to 10^8, i.e., it cannot stop this overestimated flow. Thus the profile steepens.

Our thrust has been to develop a more accurate algorithm. To do this, we have (1) obtained better representation for the fluxes at the boundary of a cell, and (2) reduced the range between the upper and lower bounds of the densities in the flux-limiter (Eqs. (10)).

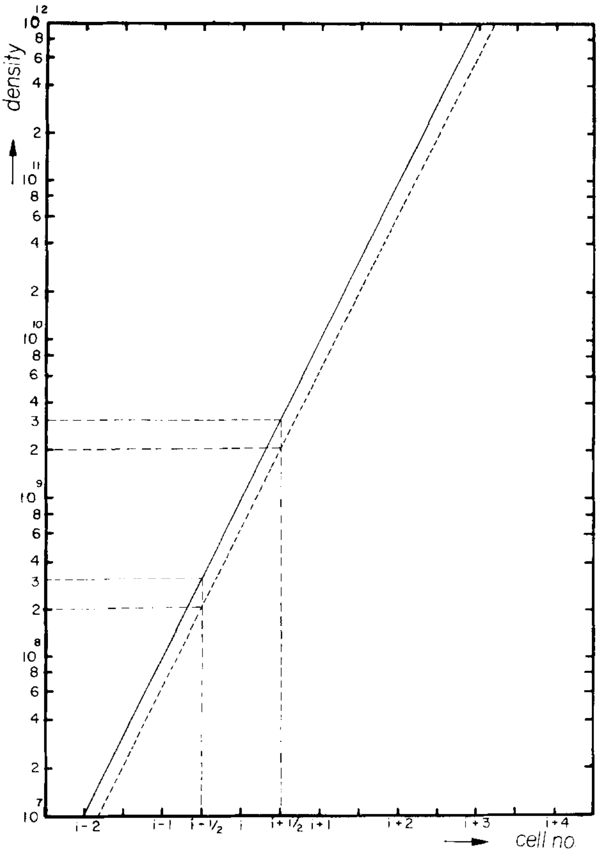

FIG. 4. Effect of the interpolation technique on the calculation of the flux at a cell boundary.

To obtain a better representation for the fluxes at the boundary, we (a) obtain the flux by independent interpolations of the fluid velocity, u, and the density n, (b) obtain the velocity and density at the boundary by interpolating functions of these quantities which have a slower variation in space and time than the quantities themselves, and (c) consider their variation inside a cell to calculate the mass flow in and out of the cell.

Since both the fluid velocity and the density may have very steep gradients, it is desirable to interpolate functions of these quantities which have a slower variation in space than the quantities themselves. We have chosen to interpolate the natural logarithm of these quantities in regions where the gradients are large. Where the gradients are small, a linear interpolating scheme is used. Thus in regions of steep gradient, we find the density in the ith cell by expanding the function $\ln[1+n'(x_i+x, t)]$, where x is the distance from x_i, and $n'(x_i+x, t) = n(x_i+x, t)/n_0$. n_0 is a scale factor. This factor is chosen so that the logarithmic interpolation scheme is satisfactory for any density range. Thus,

$$\ln[1+n'(x_i+x, t)] = \ln[1+n'(x_i, t)] + \partial_x \ln[1+n'(x, t)]\,|_{x=x_i}\, x + \cdots, \quad (11)$$

from which we find

$$n'(x_i + x, t) = [1 + n'(x_i)]\, e^{a_i x} - 1, \tag{12}$$

where

$$a_i = \partial_x \ln[1 + n'(x, t)] \,|_{x = x_1}$$

and is determined computationally from the expression

$$a_i = \frac{\ln[1 + n'(x_{i+1}, t)] - \ln[1 + n'(x_i, t)]}{\Delta x}. \tag{13}$$

The reason for using $1 + n'(x, t)$, instead of $n'(x, t)$, is to avoid taking the logarithm of zero. Note that $n'(x, t) \geqslant 0$. Obtaining the density from Eq. (12) may require considerably more time than by linear interpolation of the density, if the logarithm of a function is not done by either hardware or expedient software. This point will be addressed again in the next section, when we discuss some examples. An expression similar to Eq. (12) is used for the velocity $u'(x, t)$, where $u'(x, t) = u(x, t)/u_0$, and u_0 is a scale factor (see Appendix 1). However, since $u'(x, t)$ may be positive or negative, we expand $\ln[1 + |u'(x, t)|]$ for $u' < 0$. The flux at the boundary, $f_{i+1/2}$, can then be computed from equations for n and u (Eqs. (12) and (A1)) evaluated at $x = \Delta x/2$. From these equations, we can also compute the amount of mass that flows in and out of a cell through the boundary during time Δt. An expression for this mass flow is given in Appendix 2. We have used this mass flow to compute both the low- and high-order fluxes. Expressions for the high- and low-order schemes are given in Appendices 3 and 4, respectively.

To further reduce the error in the solution, we have restricted the range of values that the density can have at $(m+1)\Delta t$ by defining "high-order limiters." This is most important when the density gradient is large and the time step satisfies the condition $u_i \Delta t \leqslant \Delta x/2$. For this case, the values of the density at $i-1$ and $i+1$ are too far removed from the values at either i or $i + 1/2$ to serve in limiting the flux.

To obtain suitable upper and lower bounds for the density in the ith cell at time $(m+1)\Delta t$, note that when the velocity is uniform and the criterion $u \Delta t \leqslant \Delta x/2$ is satisfied, the density at $x = x_i$ is bounded by $n^m_{i+1/2}$ or $n^m_{i-1/2}$ (depending on the flow direction). At the maximum allowed velocity, $\Delta x/(2\Delta t)$, the density at x_i becomes either $n^m_{i+1/2}$ or $n^m_{i-1/2}$ (depending on the flow direction) at time $(m+1)\Delta t$.

Thus, instead of Eqs. (10), we have

$$n_i^{\max} = \max(n^m_{i+1/2}, n^m_i, n^{\mathrm{D}}_i, n^m_{i-1/2}) \tag{14}$$

$$n_i^{\min} = \min(n^m_{i+1/2}, n^m_i, n^{\mathrm{D}}_i, n^m_{i-1/2}). \tag{15}$$

$n^m_{i+1/2}$ and $n^m_{i-1/2}$ are obtained from Eq. (12), respectively. Note that when $u_{i+1/2} < 0$ and $u_{i-1/2} > 0$, or $u_{i+1/2} > 0$ and $u_{i-1/2} < 0$, the reference chosen is n^{D}_i. Equations

(14) and (15) form the basis of the "high-order" limiter. Using these expressions in Eqs. (7), (8) and (9), we obtain a better representation for the maximum anti-diffusion flux that can be used in correcting the low-order scheme.

IV. Test Results

To test the modified FCT algorithm described in the previous section, we have numerically solved Eq. (4) for the following cases

$$\text{Case 1: } \begin{cases} g_1(x) = 1 \times 10^{14} e^{-((x-30)/3)^2} \\ u(x,t) = 1 \times 10^{-7} n(x,t) \end{cases} \qquad \Delta t = 2 \times 10^{-9}\ \text{sec}$$

$$\text{Case 2: } \begin{cases} g_2(x) = g_1(x) \\ u(x,t) = 1 \times 10^{7} - 0.50 \times 10^{-7} n(x,t) \end{cases} \qquad \Delta t = 2 \times 10^{-9}\ \text{sec}$$

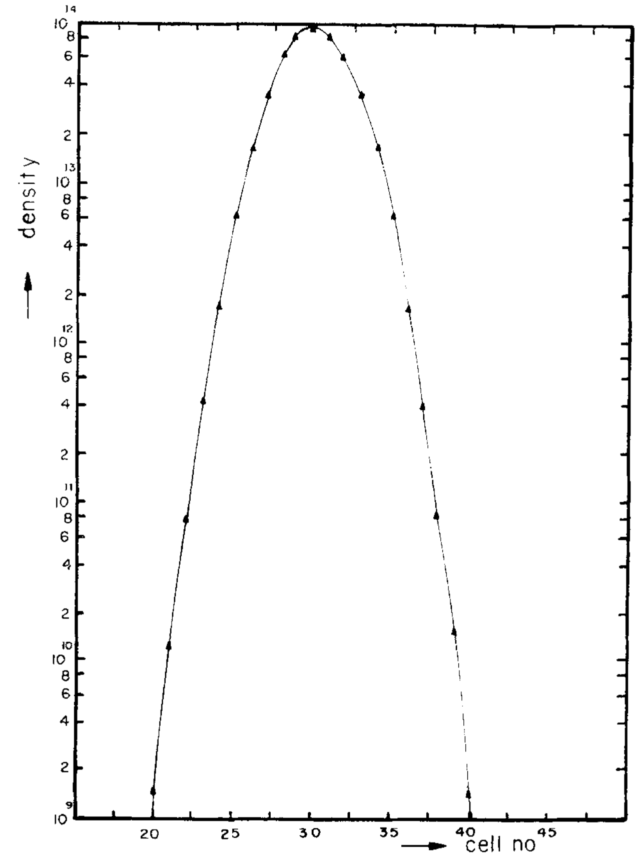

FIG. 5. Initial density distribution used is tests (1), (2), (3), and (4).

$$\text{Case 3: } \begin{cases} g_3(x) = g_1(x) \\ u(x, t) = 0.5 \times 10^7 - 1 \times 10^{-7} n(x, t) \end{cases} \qquad \Delta t = 1 \times 10^{-9} \text{ sec}$$

$$\text{Case 4: } \begin{cases} g_4(x) = g_1(x) \\ u(x, t) = \text{constant} \end{cases} \qquad \Delta t = 1 \times 10^{-9} \text{ sec}$$

$$\text{Case 5: } \begin{cases} g_5(x) = 1 \times 10^{14}, & 10 \leqslant x \leqslant 50, \ \Delta t = 2 \times 10^{-9} \text{ sec} \\ \qquad = 0 & \text{otherwise} \\ u(x, t) = \text{constant.} & \end{cases}$$

In all cases, the x-domain is uniformly sampled, with $\Delta x_i = 1$. The sampled waveform contains 200 points. The initial waveforms for the cases investigated are shown in Figs. 5 and 6. The objective of the tests has been to ascertain the robustness and accuracy of the technique for positive and negative slope waveform-steepening, and for initial conditions where the function and its slope are rapidly

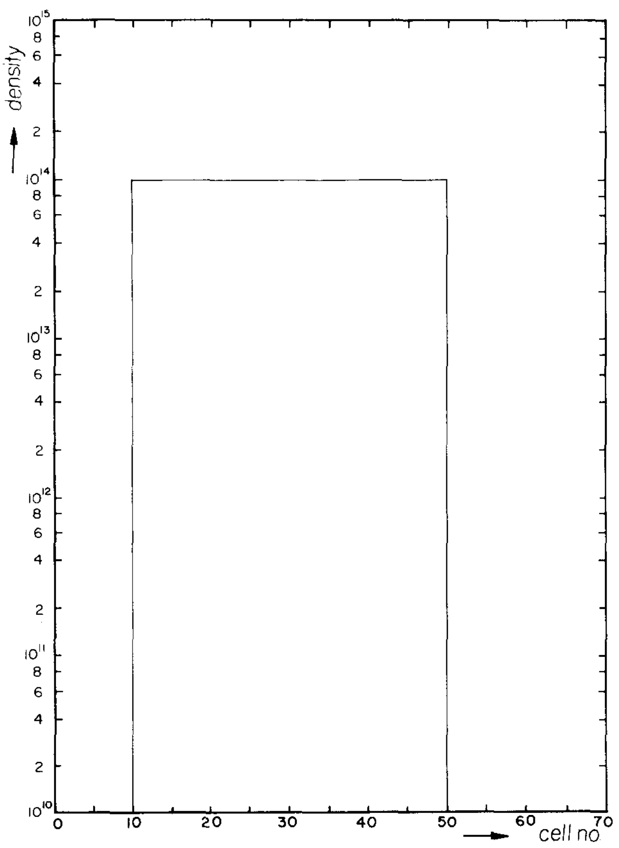

FIG. 6. Initial density distribution used in test (5).

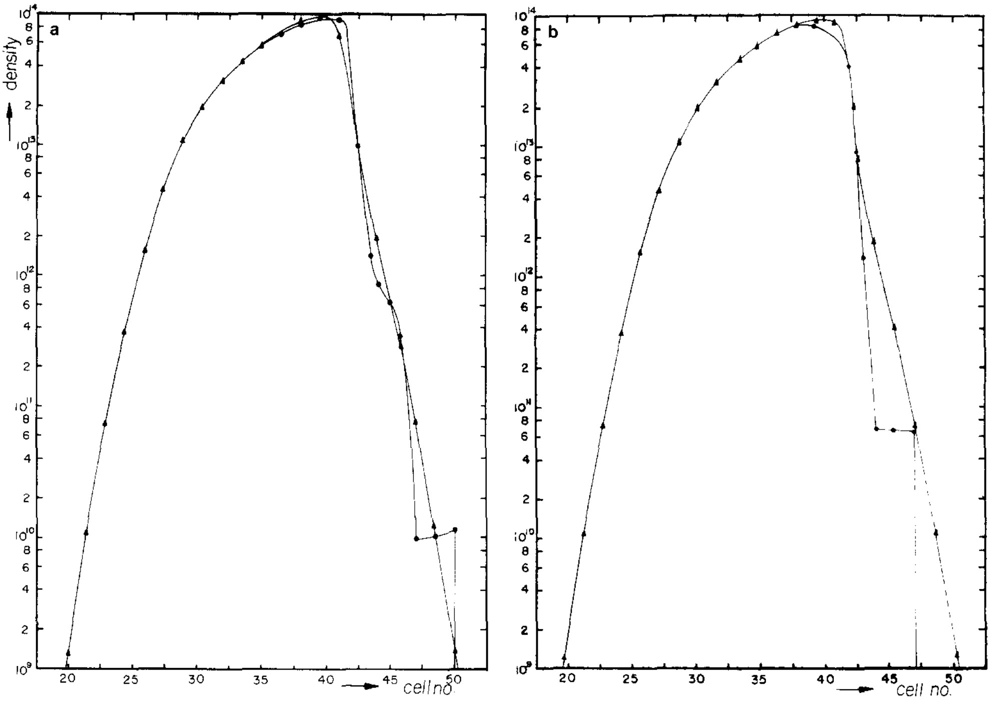

FIG. 7. Density distribution after $m = 93$ time steps: (a) FCTZ, (b) UFCT, and (c) our results. Note the staircase behavior obtained with both FCTZ and UFCT. In time, the staircase develops into oscillations. The curves with triangles correspond to the exact solution.

changing in space and have a wide dynamic range. For comparison, we have also obtained results for each of the above cases using FCTZ [11] and UFCT [12]. The UFCT makes use of the Boris and Book (BB) flux-limiter [5]. The FCTZ high-order scheme is eighth order, whereas, the UFCT is second order.

The results we have obtained after m time steps, for each of the test cases, are shown in Figs. 8–11. The number of time steps, m, is given in the figure captions. The corresponding analytic solution to the equation after the m time steps is also shown in each figure. For Cases (1)–(4), our results are more accurate than those obtained with either FCTZ or UFCT. For Case (5), FCTZ produces the best results. The reason for this, as discussed in the previous section, is that the higher-order fluxes in the FCTZ tend to steepen the waveform (the fluxes are higher than the correct values), and the flux-limiting filter allows it. The square wave initial condition, Case (5), is not useful in testing the accuracy of the results because of its singuar (infinite) slope at the edges.

For the examples we have considered, the UFCT with the BB limiter yields the highest error levels (see Fgs. 7–10). The reason is that to compute the fluxes in and out of a cell, both the upwind differencing technique and the BB limiter, utilize

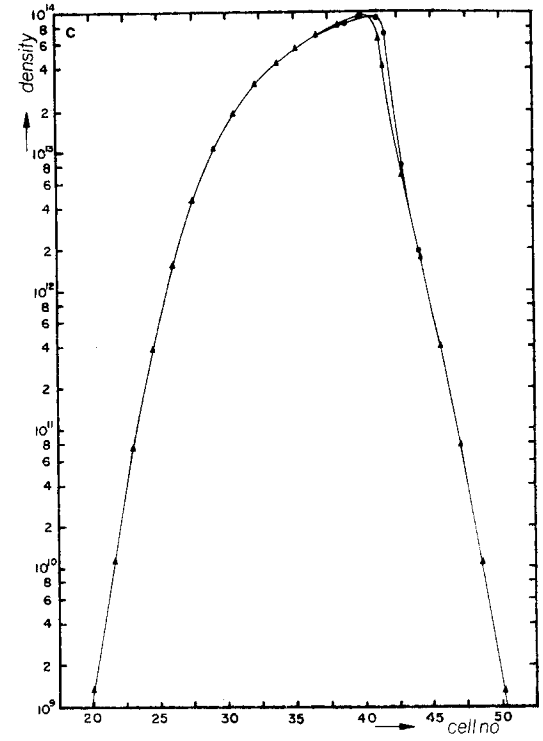

FIG. 7—*Continued.*

information that is "biased" towards one side of the cell and too far removed from it. This results, as illustrated below, in the computation of the wrong fluxes. Upwind differencing tends to steepen the waveform; whereas, the BB limiter tends to diffuse. The Zalesak algorithm, on the other hand, is more symmetric about a cell which results in a more realistic calculation of the fluxes.

To illustrate the effect of the BB limiter, assume a constant flow velocity, u, of 10^7 and consider the monotonically increasing density distribution

n_{i-1}	n_i	n_{i+1}	n_i+2
2.3×10^{10}	1×10^{11}	5×10^{11}	1.7×10^{12}

The high-order flux at the boundary $F_{i+1/2}=1.5f_i-0.5f_{i-1}\approx1.4\times10^{18}$. The low-order flux, $f_{i+1/2}$ is 1×10^{18}. The difference between these fluxes, $\delta f_{i+1/2}$, is 0.4×10^{18}. Using the BB limiter [5]; namely,

$$\delta f_{i+1/2}\rightarrow\operatorname{sgn}(\delta f_{i+1/2})\max\left\{0,\min\left[|\delta f_{i+1/2}|,\operatorname{sgn}(\delta f_{i+1/2})\right.\right.$$
$$\left.\left.\times\frac{\Delta x}{\Delta t}(n^{\mathrm{D}}_{i+2}-n^{\mathrm{D}}_{i+1}),\operatorname{sgn}(\delta f_{i+1/2})\frac{\Delta x}{\Delta t}(n^{\mathrm{D}}_{i}-n^{\mathrm{D}}_{i-1})\right]\right\}, \qquad (16)$$

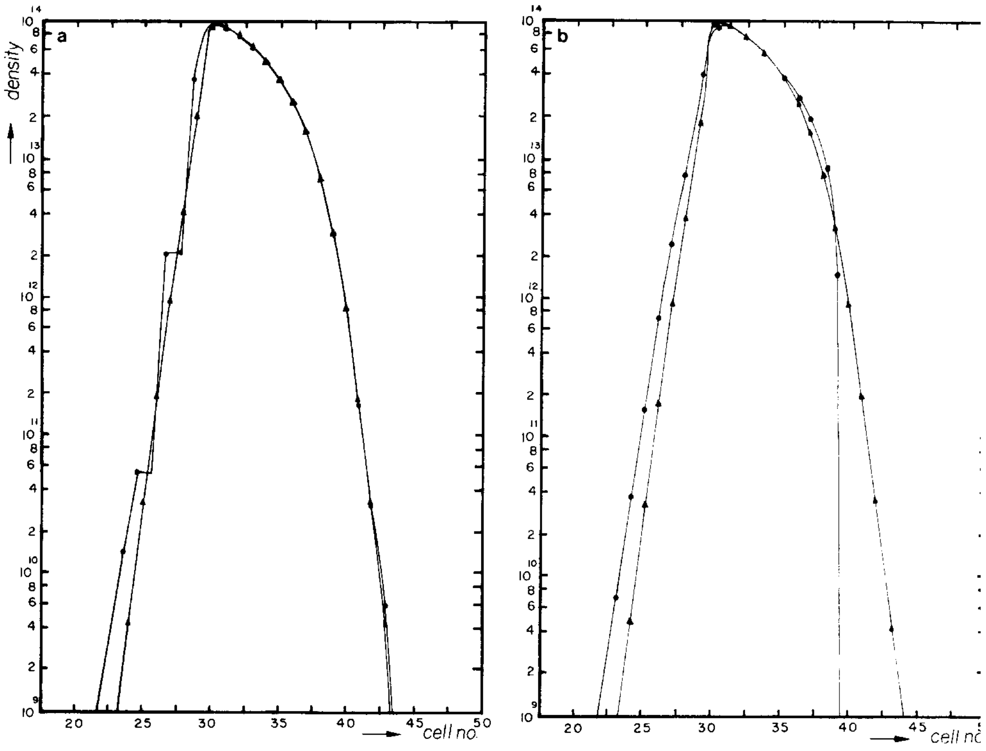

FIG. 8. Density distribution after $m+175$ time steps for test Case (2): (a) FCTZ, (b) UFCT, and (c) our results. In this case, the waveform moves to the right, but the steepening occurs in the trailing edge. The curves with triangles correspond to the exact solution.

we calculate for the anti-duffusion flux 0.4×10^{18}. Thus, the computed flux (the sum of the low-order and anti-diffusion flux) is 1.4×10^{18}. This value is lower than the actual flux through the boundary at $i+1/2$. This flux is approximately 2×10^{18}. Because of this, mass tends to lag behind in the cell; that is, the antidiffusion fluxes are not sufficient to compensate for the diffusion introduced by the low-order scheme.

Next consider the monotonically decreasing density distribution

n_{i-1}	n_i	n_{i+1}	n_{i+2}
1.7×10^{12}	5×10^{11}	1×10^{11}	2.3×10^{10}

and again assume a positive and constant flow velocity of 10^7. The high-order flux $F_{i+1/2}$ using UFCT [12], is calculated to be -1×10^{18}, whereas, the low-order flux, $f_{i+1/2}$, is found to be 5×10^{18}. The difference between these two fluxes, $\delta f_{i+1/2}$, is -6×10^{18}. Using the BB limiter (Eq. (16)), we find for the anti-diffusion flux -6×10^{18}, and for the computed flux -1×10^{18}. Note that the computed flux

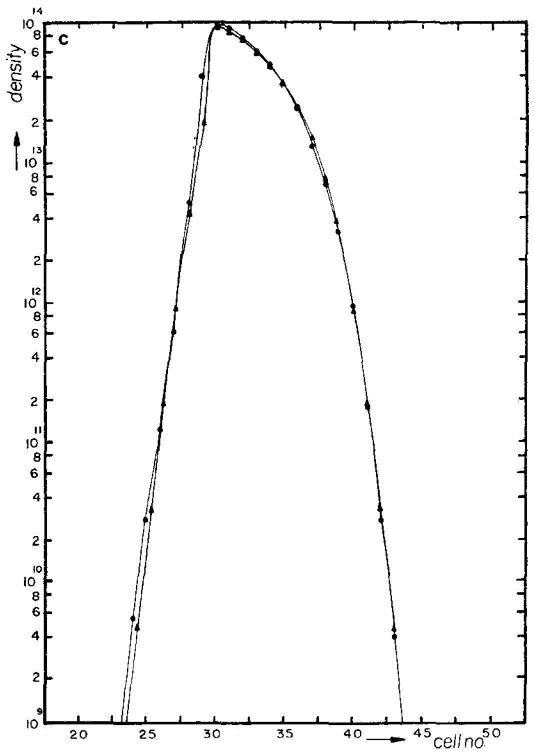

FIG. 8—*Continued.*

through the boundary at $i + 1/2$ is negative; the flow through that boundary is into cell i. The flow should obviously be out of the cell. This effect tends to steepen the profile. The staircase behavior obtained with the FCTZ (see Fig. 8) has been noted before [10]. The staircase further evolves into ripples. A similar behavior is also observed with the UFCT, when used in conjunction with Zalesak's flux-limiter.

V. DISCUSSION AND CONCLUSIONS

An objection to using the algorithm discussed in Section IV is the time required to compute the ln of the function at each position x_i. Considering the number of times that this has to be done, the algorithm may be very time-consuming if this tep is not done with expedience. Either hardware or microsoft techniques for calculating the ln should yield the fastest results. These investigations were carried out on a Gould 6005 minicomputer, where the ln of a number is done by software. Our simulation times are of the same order as the FCTZ, but much greater than the UFCT. Recall that the high-order UFCT scheme used is second order, whereas the high-order FCTZ is eighth order, so that increasing the order of the UFCT

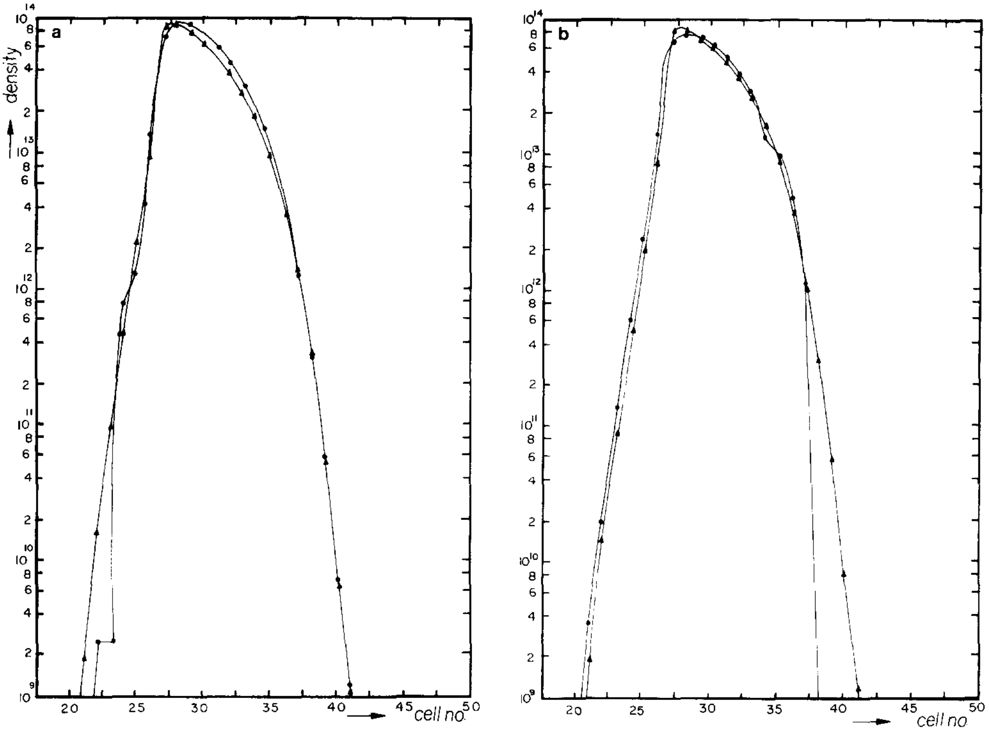

FIG. 9. Density distribution after $m = 175$ time steps for test Case (3): (a) FCTZ, (b) UFCT, and (c) our results. In this case, the steepening occurs at both ends. The curves with triangles correspond to the exact solution.

would result in comparable computation times. The question arises as to whether it would be more economical to increase the number of grid points and use another less time-consuming interpolating scheme. To obtain the accuracy we have achieved (with the large dynamic range of the waveform), the number of points would have to be increased by at least a factor of 10. (We have not considered non-uniform griding techniques.) The computation time in this case using either FCTZ or UFCT (eighth order) would be considerably greater than for our case. In adition, since in our applicaton we also have to solve Eq. (3) and include in Eqs. (1) and (2) non-local effects (such as photoionization [3]), the increase in the number of points would make the simulation impossible (these additions take considerable more time than calculating the flow). As mentioned in the Introduction, accuracy is necessary at each step if the results are to have any physical meaning. Errors in the calculation of the electron and ion densities in one time step result in errors in the calculation of the space-charge electric field. Both of these errors are greatly amplified with time because of ionization growth of the densities. This growth is exponential with v_i which, on the hand, exponentially depends on the total electric

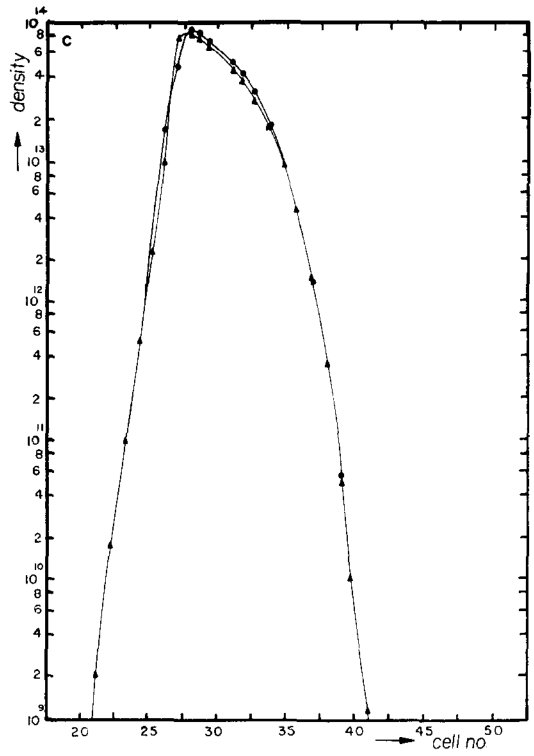

FIG. 9—*Continued.*

field. The accuracy we have achieved and the time taken to run this algorithm has allowed us to simulate the evolution of Eqs. (1)–(3) in two dimensions. We have used time-splitting to achieve two-dimensionality. As mentioned in Section II, this avoids the formation of ripples. However, the time step must be sufficiently small to avoid the introduction of artificial compressibility effects [11]. This condition is satisfied in the case of Eqs. (1)–(3). In this case, the time step is limited by the more stringent condition that the field be nearly constant in a time step. These results will be presented in a future paper.

We have not obtained a stability criterion for the numerical scheme presented. However, since the transport scheme (not taking into account the flux limiter) has been derived by considering the mass flow through the boundary between the i and $i+1$ cell, only using the information at mesh points i and $i+1$ (see Appendix 2 and Section III), the time step must be taken to be $\leqslant \Delta X/(2U_{\max})$, where $U_{\max}$ is the maximum velocity in the space mesh. This condition guarantees that the calcuation of the mass flow through the boundary during a time Δt does not involve information coming from mesh points further than i, $i+1$. This is in essence the Courant–Friedricks–Lewey condition [14]. In all our simulations the above condition was always satisfied. In such cases, the scheme was found to be stable.

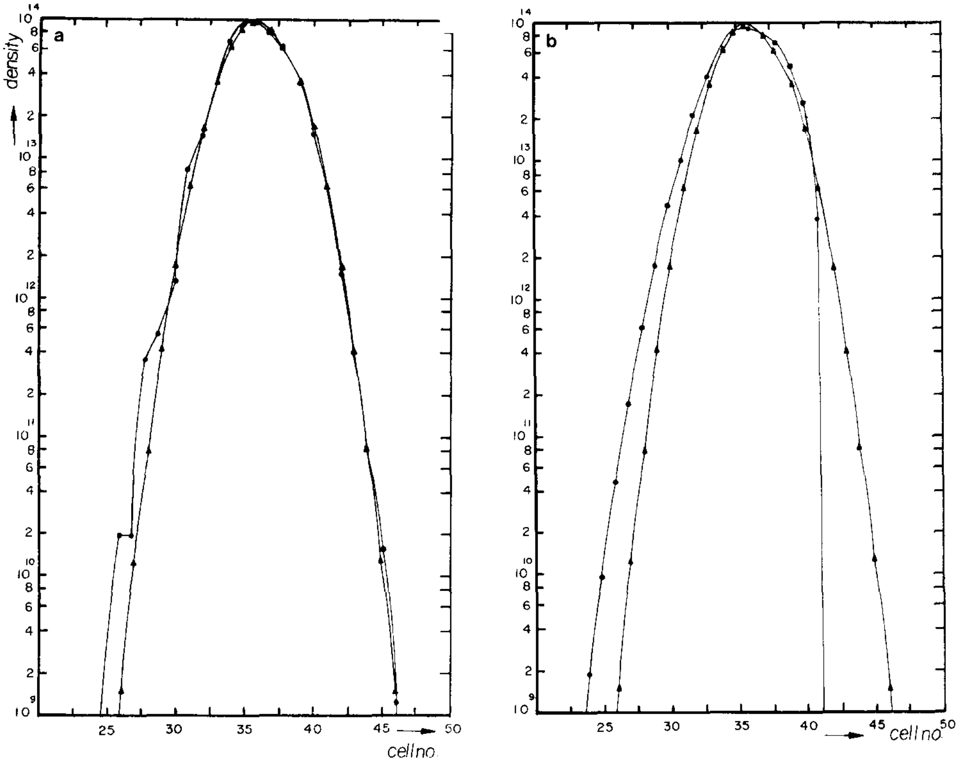

FIG. 10. Density distribution after $m = 600$ time steps for test Case (4): (a) FCTZ, (b) UFCT, (c) our results. This case corresponds to the uniform displacement of the initial distribution. The curves with triangles correspond to the exact solution.

APPENDIX 1: CALCULATION OF THE FLOW VELOCITY

Let $u_i' = u'(x_i)$, $u_{i+1}' = u'(x_{i+1})$. As in the calculation of the density, the velocity in the ith cell is obtained from the expression

$$u'(x_i + x) = S[(1 + |u_i'| + S_2 \cdot 2\,|u_{i+1}'|)\,e^{b_i x} - 1] - S_1 \cdot 2\,|u_i'| - S_2 \cdot 2\,|u_{i+1}'|], \qquad \text{(A1)}$$

where

$$b_i = \frac{1}{\Delta x}\,[\ln(1 + |u_{i+1}'| + S_1 \cdot 2\,|u_i'|) - \ln(1 + |u_i'| + S_2 \cdot 2\,|u_{i+1}'|)]$$

and

$$\begin{array}{llll} S_1 = 0, & S_2 = 0, & S = 1 & \text{when} \quad u_i' \geqslant 0, \quad u_{i+1}' \geqslant 0 \\ S_1 = 0, & S_2 = 0, & S = -1 & \text{when} \quad u_i' < 0, \quad u_{i+1}' < 0 \\ S_1 = 0, & S_2 = 1, & S = 1 & \text{when} \quad u_i' \geqslant 0, \quad u_{i+1}' < 0 \\ S_1 = 1, & S_2 = 0, & S = 1 & \text{when} \quad u_i' < 0 \quad u_{i+1}' \geqslant 0. \end{array}$$

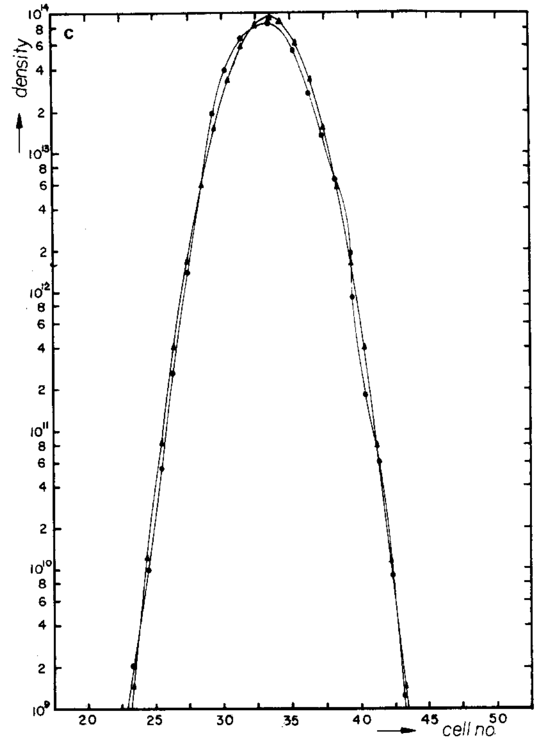

FIG. 10—*Continued.*

APPENDIX 2: CALCULATION OF THE MASS FLOW

Let xt be the position at time t which will reach $x_i + \Delta x/2$ at time $t + t_1$. Then, the mass between $x = xt$ and $x = x_i + \Delta x/2$ will flow into the $i+1$ cell. Using the expressions for $u(x_i + x)$ and $n(x_i + x)$ (Eqs. (A1) and (12), respectively), we find

$$xt = \frac{\Delta x/2 - u_i t_1}{1 + b_i(1 + u_i)\, t_1}.$$

t_i is the duration of the time interval. The mass transferred to the $i+1$ cell is

$$\Delta N = \int_{xt}^{\Delta x/2} [(1 + n_i)\, e^{a_i x} - 1]\, dx,$$

i.e.,

$$\Delta N = \left\{ \frac{(1 + n_i)}{a_i} (e^{a_i(\Delta x/2)} - e^{a_i xt}) - \frac{\Delta x}{2} + xt \right\}.$$

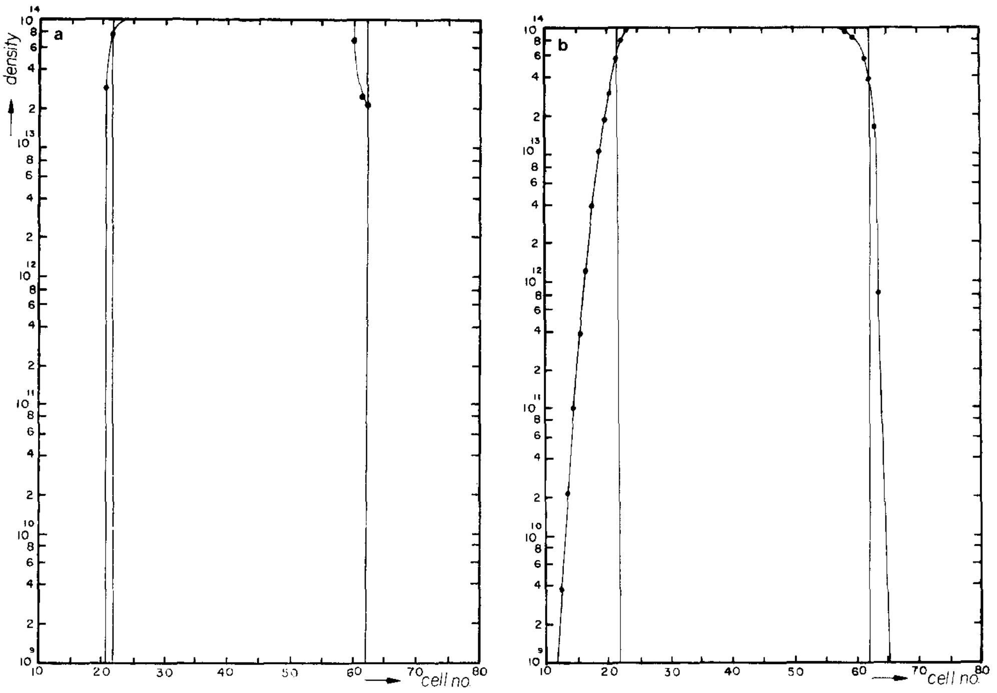

FIG. 11. Density distribution after $m = 600$ time steps for test Case (5): (a) FCTZ, (b) UFCT, (c) our results.

APPENDIX 3: THE LOW-ORDER SCHEME

The flow velocity at the boundary of the ith cell is obtained from Eq. (A1) by letting $x = \Delta x/2$, i.e., $u_{i+1/2} = u(x_i + \Delta x/2)$. The flux at the boundary for $u_{i+1/2} \geqslant 0$ is

$$f_{i+1/2} = u_{i+1/2} n_i, \qquad \text{where} \quad n_i = n(x_i) \text{ and } u_{i+1/2} \text{ is from (A1);}$$

otherwise $f_{i+1/2} = u_{i+1/2} n_{i+1}$. The density at $(m+1)\,\Delta t$ is then obtained from the expression

$$n_i^{\mathrm{D}} = n_i^m - \frac{\Delta t}{\Delta x}\,(f_{i+1/2} - f_{i-1/2}).$$

APPENDIX 4: THE HIGH-ORDER SCHEME

The high-order scheme for computing the flux through the boundary at $i + 1/2$, $F_{i+1/2}$, is described below. The flux is computed by using the results from Appendixes 1 and 2 and recognizing that the average density in a cell need not be equal to

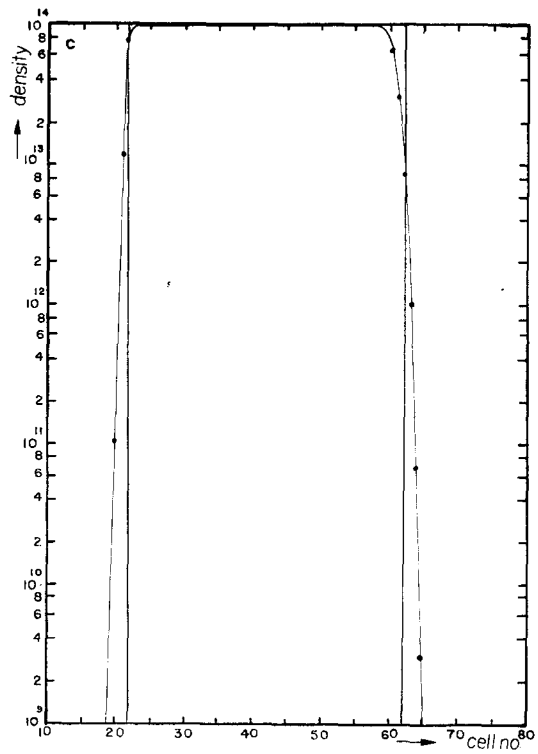

FIG. 11—*Continued.*

the value of the density at the center. When the gradients are small, this difference can be neglected. However, in the region of high gradients, we must differentiate between the two values in order to calculate the correct flux at the boundary. In such regions, and when u^m_{i+1} and $u^m_i > 0$, let

$$F_{i+1/2}(n^c_i, u^c_i) = \frac{(1+n^c_i)((e^{a_i(\Delta x/2)} - e^{a_i xt})/a_i) - \Delta x/2 + xt}{t_1}, \tag{A2}$$

where

$$xt = \frac{\Delta x/2 - u^c_i t_1}{1 + (1 + u^c_i)\, t_1 b_i} \tag{A3}$$

$$n^c_i = \frac{a_i}{2}\,\frac{(n^m_i \Delta x + \Delta x)}{\sinh((a_i/2)\,\Delta x)} - 1 \tag{A4}$$

$$u^c_i = \frac{b_i}{2}\,\frac{(u^m_i \Delta x + \Delta x)}{\sinh((b_i/2)\,\Delta x)} - 1 \tag{A5}$$

and t_1 has been defined in Appendix 2. Equations (A2)–(A5) get modified depending on the sign of u_i^m and u_{i+1}^m as discussed in Appendix 2. When the gradients are small (i.e., $w_{i+1}/w_i < 5$ or $w_{i+1}/w_i > 0.2$ and $w_{i+1}, w_i < 1$ in our code, where w_i represents either n_i or u_i,) then the average value in a cell is very close to the center value. In this case

$$n_i^c = n_i^m, \qquad u_i^c = u_i^m.$$

The flux of the high-order scheme is obtained in two steps. First, an intermediate state density is calculated using

$$n_i^i = n_i^m - \frac{\Delta t}{2} \frac{(F_{i+1/2}(n_i^c, u_i^c) - F_{i-1/2}(n_i^c, u_i^c))}{\Delta x}$$

and

$$n_i^{\text{int}} = \frac{n_i^t + n_i^c}{2}.$$

Moreover, if $n_i^{\text{int}} < 0$, then $n_i^{\text{int}} = 0$. The flux of the high-order scheme is

$$F_{i+1/2}^{\text{H}} = F_{i+1/2}(n_i^{\text{int}}, u_i^c).$$

Acknowledgment

This work was supported by the Office of Naval Research.

References

1. L. B. Loeb, *Science* **148**, 1417 (1965).
2. I. P. Shkarofsky, T. W. Johnston, and M. P. Bachynski, *The Particle Kinetics of Plasmas* (Addison–Wesley, Reading, Mass., 1966), p. 46.
3. L. E. Kline, *J. Appl. Phys.* **46**, 1994 (1975).
4. E. E. Kunhardt and P. F. Williams, *J. Comput. Phys.* **57**, 403 (1985).
5. J. P. Boris and D. L. Book, *J. Comput. Phys.* **11**, 38 (1973).
6. J. P. Boris and D. L. Book, *J. Comput. Phys.* **20**, 397 (1976).
7. J. P. Boris and D. L. Book, *Methods of Computational Physics*, edited by John Killeen (Academic Press, New York, 1976).
8. B. Van Leer, *J. Comput. Phys.* **23**, 276 (1977).
9. P. Colella and P. R. Woodward, *J. Comput. Phys.* **54**, 174 (1984).
10. M. Ben-Artzi and J. Falcovitz, *J. Comput. Phys.* **55**, 1 (1984).
11. S. T. Zalesak, *J. Comput. Phys.* **31**, 335 (1979).
12. B. E. McDonald and J. Ambrosiano *J. Comput. Phys.* **56**, 448 (1984).
13. P. Woodward and P. Colella, *J. Comput. Phys.* **54**, 115 (1984).
14. D. Potter, *Computational Physics* (Wiley, Chichester, New York, 1977), p. 65.
15. A. J. Davies, C. J. Evans, P. Townsend, and P. M. Woodison, *Proc. IEE* **124**, 171 (1977).
16. K. Yoshida, T. Taniguchi, and H. Tagashira, *J. Phys. D* **12**, L3 (1979).
17. R. Morrow and J. J. Lowke, *J. Phys. D* **14**, 2027 (1981).