JOURNAL OF COMPUTATIONAL PHYSICS 80, 61–71 (1989)

An Implicit Flux-Corrected Transport Algorithm

P. STEINLE* AND R. MORROW†

*Department of Applied Mathematics, The University of Adelaide, and
†CSIRO Division of Applied Physics, Sydney, Australia 2070

Received May 15, 1987; revised January 20, 1988

An implicit flux-corrected transport algorithm is developed which gives accurate, non-negative results for Courant numbers $c > 1$, and retains high accuracy for $c < 1$. The new algorithm can give a threefold increase in overall speed over existing methods with the same order of accuracy, or give greater accuracy for the same computational effort; the accuracy decreases slowly as c increases beyond unity. The method has been developed for application in discharge physics problems where a high order of accuracy is required for solving the continuity equations for electrons and ions under the influence of dominant space-charge effects. We follow Zalesak's approach by computing both a high-order solution and a low-order solution and then using some flux-limiter to determine what proportion of each solution is used at any given point in space and time [*J. Comput. Phys.* **31**, 335 (1979)]. For the high-order solution we use a fourth-order time- and space-centred scheme. For the lower-order solution we use upwind differences. The fourth-order scheme presented is more accurate than the REVFCT algorithm proposed by Boris and Book [*J. Comput. Phys.* **20**, 397 (1976)] and guarantees positive results.

1. INTRODUCTION

Equations describing the drift of charged particles in an electric field form the basis of most theoretical studies of gaseous discharges. In many situations the electric field is controlled by space-charge effects, and must therefore be obtained by solving Poisson's equation. In these cases precise numerical algorithms are needed to take proper account of charge cancellation in the evaluation of the net charge density. The prototype equation to be solved accurately for the movement of charge is

$$\frac{\partial \rho}{\partial t} + \frac{\partial (w\rho)}{\partial x} = 0, \tag{1}$$

where ρ is the particle density and w the drift velocity. A comparison [1] of several methods used to solve this equation indicated that the explicit flux-corrected transport (FCT) algorithms of Boris and Book [2] have many advantages. Examples of the application of such an algorithm to studies of gaseous discharges where space charges dominate have been given by Morrow [3, 4]. These examples involve a very large number of computational steps [3] since we are restricted to values of Courant number $c < 0.5$, where $c = w\,\Delta t/\Delta x$, Δt is the time step, and Δx is the mesh

0021-9991/89 $3.00

581/80/1-5

size. This restriction severely limits the time step and the kind of problem which can be solved. Moreover, in such calculations near the cathode the electron density may be low and not controlling the space-charge field, yet the time step is limited by the largest value of c for electrons in this region.

Thus it is of vital importance to examine the possibility of performing accurate calculations using $c \gg 1$, particularly if the electron density is often low and extreme accuracy not as important as stable non-negative results. Note that the SHASTA FCT method [6] is limited to $c < 0.5$. Clearly the way to perform calculations with $c \gg 1$ is to develop an implicit form of FCT. Previous implicit forms of FCT have not given a non-negative treatment for the mass continuity equation [5, 6]. Patnaik *et al.* [5] treat the energy and momentum equations implicitly, but use an explicit form for the mass continuity equation, while the REVFCT scheme of Boris and Book [6] gives negative results by their own admission (Ref. [6, Table 2]). Thus in this paper we present an original implicit treatment of advection using a true FCT method.

In this paper we explore the properties of implicit forms of FCT which combine the inherent advantages of implicit methods with the benefits endowed by the flux-correction process, and we apply the method to the mass continuity equation. Further, we develop and test ways of using the implicit schemes with an arbitrary Courant number; it is not possible to use an arbitrary Courant number with the explicit forms of the FCT algorithm. In so doing we follow Zalesak's method [7] and compute both a high-order flux and a low-order flux, and use the flux-correction process to determine what fraction of each solution is used at a given point in space and time. For the high-order scheme we use a fourth-order space- and time-centred method [8]. For the low-order scheme we use upwind differences, a simple explicit technique which guarantees non-negative results with minimal numerical diffusion [9]. (Initially tests were made using the Crank–Nicolson implicit method for the high-order scheme. However, while flux-correction worked, the results for $c = 0.8$ and 200 time steps were distorted in a similar manner to the results for "donor-cell ZRD" shown in Fig. 5a of Ref. [2], and the computer time taken was comparable to that for the present method.)

After developing and testing the implicit method for $c < 1$ we then explore the possibility of using the method under the condition $c > 1$. For the lower-order method we use multiple steps of upwind differences within the larger implicit step. The fourth-order method is theoretically stable for $c > 1$ [8], but the matrix inversion is not stable if standard techniques such as the Thomas algorithm are used. However, we develop a marching method which allows us to quickly and accurately solve the equations for $c > 1$.

In this paper we concentrate on developing an accurate and efficient algorithm with a uniform spatial mesh and constant positive velocity. Thus, the partial differential equation we are solving in this paper is more strictly written as

$$\frac{\partial \rho}{\partial t} + w \frac{\partial \rho}{\partial x} = 0. \tag{2}$$

We test the algorithms with the standard square-wave test and also with a new semicircle test suggested by Zalesak [10] which can often reveal faults that the standard square-wave test fails to detect.

2. Numerical Methods and Results

2.1. *Fourth-Order Scheme*

For a high-order solution we use the fourth-order implicit scheme

$$(2-3c+c^2)\,\bar{\rho}_{j-1}+(8-2c^2)\,\bar{\rho}_j+(2+3c+c^2)\,\bar{\rho}_{j+1}$$
$$=(2+3c+c^2)\,\rho^n_{j-1}+(8-2c^2)\,\rho^n_j+(2-3c+c^2)\,\rho^n_{j+1}. \tag{3}$$

The simplest low-order non-negative method is the upwind-difference scheme

$$\tilde{\rho}_j=\rho^n_j-c(\rho^n_j-\rho^n_{j-1}), \tag{4}$$

where ρ^n_j is the density at point j and time step n, $\bar{\rho}_j$ is the high-order solution to the implicit scheme at time level $n+1$, and $\tilde{\rho}_j$ is the lower-order solution at time level $n+1$.

From Eqs. (3) and (4) we can write down the high- and low-order fluxes, $\phi^{\mathrm{H}}_{j+1/2}$ and $\phi^{\mathrm{L}}_{j+1/2}$, respectively, which represent the flux of particles across the cell boundary between mesh point j and $j+1$ due to the application of the high- and low-order schemes. We then derive the antidiffusive flux $\phi_{j+1/2}$ required to transform the lower-order solution to the high-order solution, and effectively remove the numerical diffusion introduced by the low-order scheme, by noting that

$$\phi^{\mathrm{H}}_{j+1/2}=\frac{(2+c^2)}{12}(\bar{\rho}_{j+1}-\bar{\rho}_j-\rho^n_{j+1}+\rho^n_j)+\frac{c}{4}(\bar{\rho}_{j+1}+\bar{\rho}_j+\rho^n_{j+1}+\rho^n_j), \tag{5}$$

$$\phi^{\mathrm{L}}_{j+1/2}=c\rho^n_j, \tag{6}$$

and hence

$$\phi_{j+1/2}=\phi^{\mathrm{H}}_{j+1/2}-\phi^{\mathrm{L}}_{j+1/2} \tag{7}$$

$$=\frac{(2+c^2)}{12}(\bar{\rho}_{j+1}-\bar{\rho}_j-\rho^n_{j+1}+\rho^n_j)+\frac{c}{4}(\bar{\rho}_{j+1}+\bar{\rho}_j+\rho^n_{j+1}-3\rho^n_j). \tag{8}$$

The solution of Eq. (3) requires the inversion of a tridiagonal matrix, which remains diagonally dominant for $c\leqslant 1$. We set the density at the boundaries to zero and do not seek solutions near the boundaries as the treatment of the boundary conditions depends on the particular problem. We can use the Thomas algorithm [11] to invert the matrix efficiently. Once the high- and low-order solutions and the antidiffusive flux have been determined we use either Boris and Book's flux limiter [2] or Zalesak's limiter [7] without his peak-preserver. (We omit Zalesak's

peak preserver as this has been found to produce overshoots.) After using a flux-limiter to prevent the growth of unwanted maxima and minima, and to maintain positivity, we obtain a corrected flux $\bar{\phi}_{j+1/2}$ and compute a new solution

$$\rho_j^{n+1} = \tilde{\rho}_j + \bar{\phi}_{j-1/2} - \bar{\phi}_{j+1/2}. \tag{9}$$

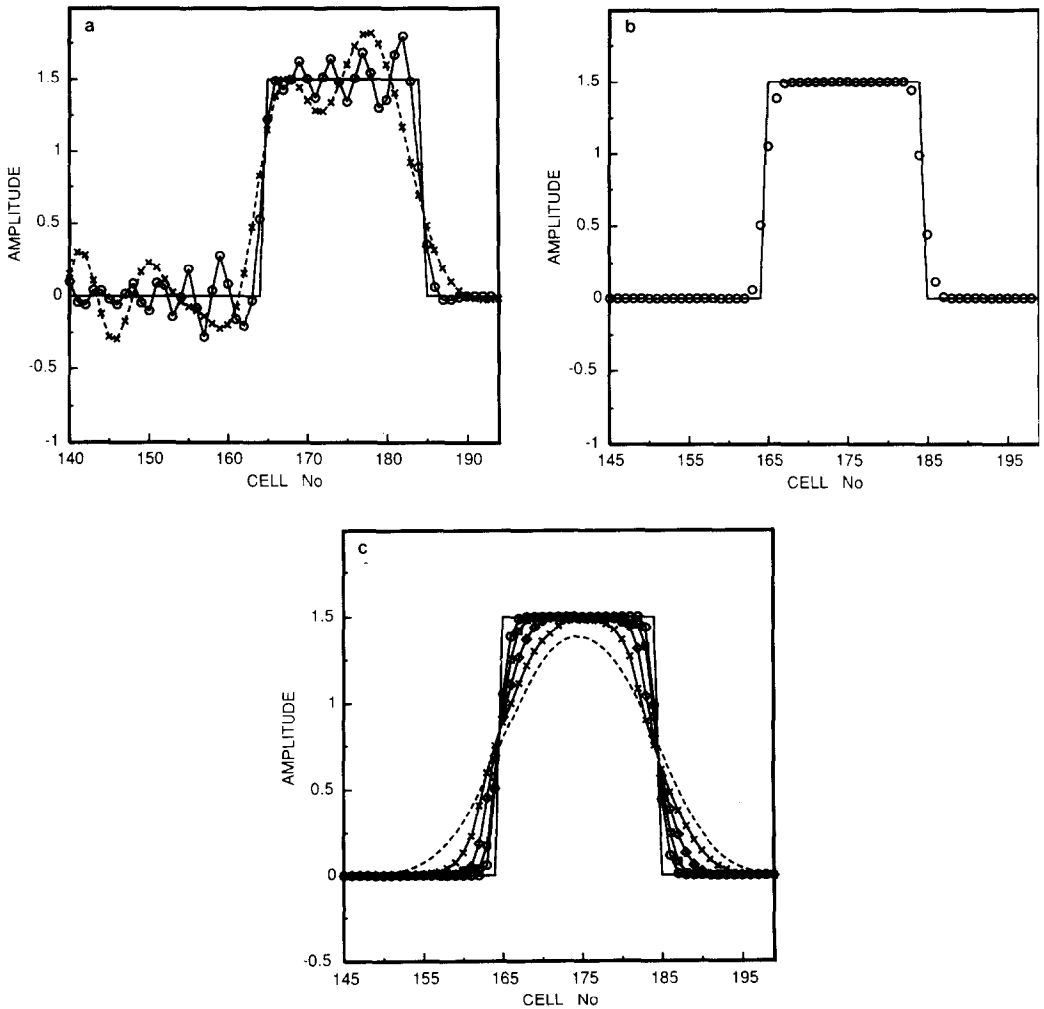

FIG. 1. Fourth-order algorithm solutions for the square-wave test. (a) Fourth-order algorithm alone: ○, $c = 0.8$, using 200 time steps; ×, $c = 4.0$, using 40 time steps; —, the exact solution. (b) Flux-corrected fourth-order algorithm, using Boris and Book's limiter: ○, $c = 0.8$, using 200 time steps; —, the exact solution. (c) Flux-corrected fourth-order solutions, using Boris and Book's limiter: ○, $c = 0.8$, 200 time steps; □, $c = 2.5$, 64 time steps; ◇, $c = 4.0$, 40 time steps; ×, $c = 8.0$, 20 time steps; ---, upwind-differences, $c = 0.8$, 200 time steps; —, exact solution.

Use of upwind differences alone results in considerable numerical diffusion (see [1]), as shown in Fig. 1c. The fourth-order method can be shown to be stable for all values of c, and to produce a diagonally dominant system of linear equations for $c \leqslant 1$, which must be solved. The solution using the fourth-order method alone for the square-wave test with 200 steps and $c = 0.8$ is shown in Fig. 1a. A large improvement is obtained when the flux-correction technique is incorporated: as shown in Fig. 1b the result is then excellent. The average error, computed using Boris and Book's formula [2], is A.E. = 0.032 as compared with a square wave amplitude of 1.5. This result is very good since we have used a larger value of c than that employed by Boris and Book for most of their tests. In fact Boris and Book's schemes could not be used for such a Courant number. Another valuable test to apply in such cases is to transport a density wave with a semi-circular profile; the result of using the fourth-order method alone, for a semicircle of radius 10 mesh spacings, is shown in Fig. 2a and with flux correction in Fig. 2b. Again the results are close to the exact solution, although the flux corrector is unable to remove the slight distortion introduced by the high-order scheme.

Although it can be shown that for $c > 1$ the fourth-order method is stable, the inversion of the tridiagonal matrix of coefficients by normal methods is no longer stable. However, for $c > 1$ we can rewrite Eq. (3) to give the recursion relation

$$\begin{aligned}(2-3c+c^2)\,\bar{\rho}_{j-2} + (8-2c^2)\,\bar{\rho}_{j-1} + (2+3c+c^2)\,\bar{\rho}_j \\ = (2+3c+c^2)\,\rho^n_{j-2} + (8-2c^2)\,\rho^n_{j-1} + (2-3c+c^2)\,\rho^n_j.\end{aligned} \tag{10}$$

This recurrence equation can be shown to be stable if and only if $c > 1$. This equation is not diagonally dominant for $c > 4$ and so if treated as a matrix inversion problem it is not clear that the method would be stable. However, by treating it as a recurrence equation it can be shown to be stable as follows. The errors introduced by Eq. (10) will grow according to the homogeneous form of the equation

$$(2-3c+c^2)\,\xi_{j-2} + (8-2c^2)\,\xi_{j-1} + (2+3c+c^2)\,\xi_j = 0, \tag{11}$$

where ξ_j is the error in $\bar{\rho}_j$ due to the marching scheme. This equation has complex eigenvalues λ that lie within the unit circle if and only if $c > 1$. So the values of ξ_j are bounded for all j and the scheme is therefore stable.

This equation must be solved in the direction of the flow and needs two value at the upstream boundary for initialization; these values may be found using some other algorithm and need not concern us here. Once the high- and low-order solutions are known, the antidiffusive fluxes may be determined by computing in he direction of the flow, away from the upstream boundary, using

$$\phi_{j+1/2} = \tilde{\rho}_j - \bar{\rho}_j + \phi_{j-1/2}. \tag{12}$$

This is the most rapid method of computing the antidiffusive fluxes. For $c > 1$ we can use just one time-step by marching the fourth-order method and multiple upwind steps with $c = 0.8$.

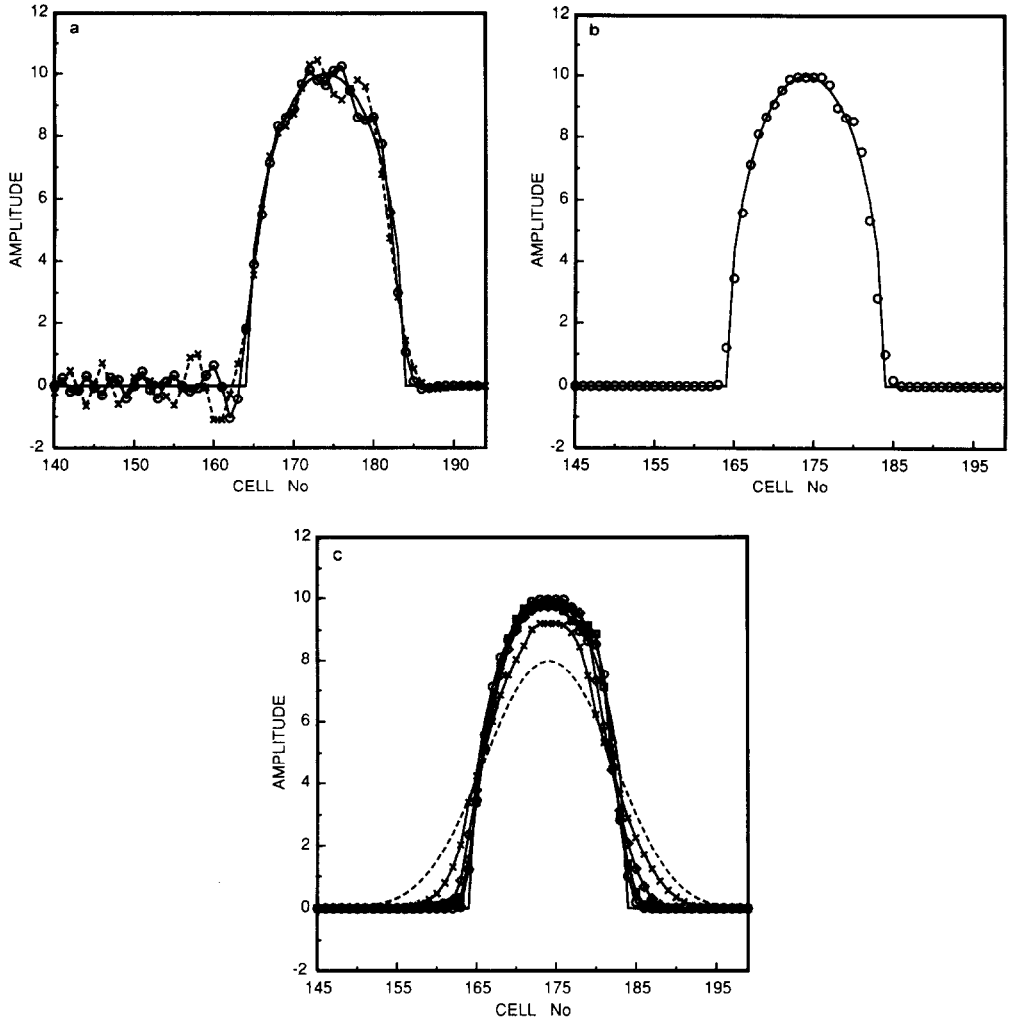

FIG. 2. Fourth-order algorithm solutions for the semicircle test. (a) Fourth-order algorithm alone: ○, $c = 0.8$, using 200 time steps; ×, $c = 2.5$, using 64 time steps; —, the exact solution. (b) Flux-corrected fourth-order algorithm, using Boris and Book's limiter: ○, $c = 0.8$, using 200 time steps; —, the exact solution. (c) Flux-corrected fourth-order solution, using Boris and Book's limiter: ○, $c = 0.8$, 200 time steps; □, $c = 2.5$, 64 time steps; ◇, $c = 4.0$, 40 time steps; ×, $c = 8.0$, 20 time steps; ---, upwind-differences, $c = 0.8$, 200 time steps; —, exact solution.

We do not use $c = 1$ for the upwind steps, since this would give an exact solution and a false indication of accuracy; such accuracy would not hold for the more general case of variable velocity and mesh size. Furthermore, it is desirable that the low-order solution should have more diffusion than the high-order solution for the flux-correction process to operate efficiently. Generally we use $c = 0.8$ for the

multiple upwind steps, with the last step adjusted to make up the implicit time step. The low-order flux is derived as the sum of the low-order fluxes from each upwind step

$$\phi^{L}_{j+1/2} = \sum_{i=1}^{k} \phi^{i}_{j+1/2}, \tag{13}$$

where k is the number of upwind time steps and $\phi^{i}_{j+1/2}$ is the flux due to each upwind-difference step.

We used this approach, with multiple upwind steps, to obtain the results for $c > 1$ shown in Fig. 1c. Clearly the method works very well within the range $1 < c < 3$, but, when c increases above 3, more diffusion must be added this degrades the solution. The results are always non-negative and no spurious wiggles develop. For the semicircle test we also get excellent results, as shown in Fig. 2c. We can demonstrate the deterioration of the solution as c increases in a semiquantitative way by plotting the average error as a function of c, as shown in Fig. 3a for the square-wave test and in Fig. 3b for the semicircle test. Note that the distorted high-order solution shown in Fig. 1a never occurs when the flux-correction process is used, since such solutions are suppressed in favour of a more correct approximation to the exact solution of Eq. (1). It should also be realised that the average errors are only qualitative so that the values for the semicircle test should not be compared with those for the square-wave test.

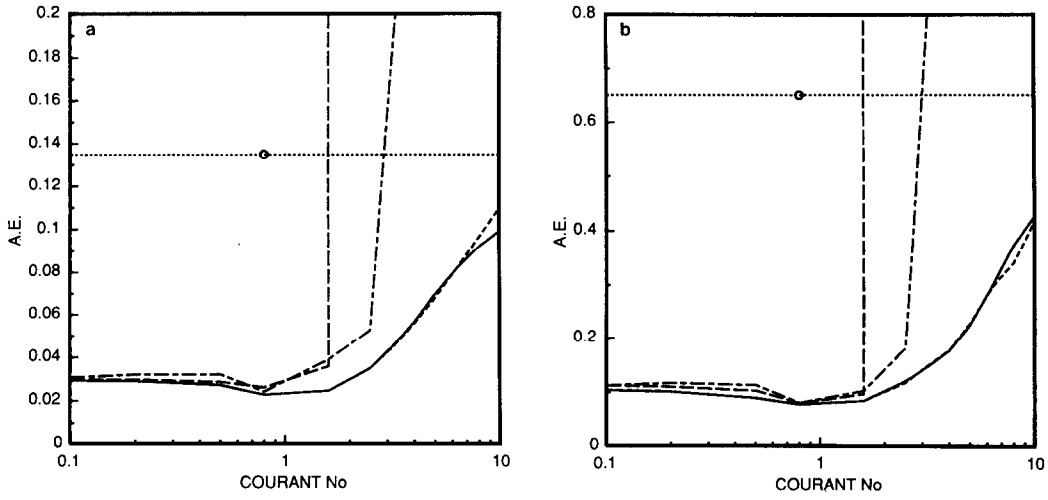

FIG. 3. Average error A.E. computed as a function of Courant number c using Boris and Book's formula [2]. (a) Square-wave test: —, fourth-order method with Boris and Book's limiter; ---, fourth-order method with Zalesak's limiter; –·–, REVFCT with Boris and Book's limiter; – – –, REVFCT with Zalesak's limiter; ⊙, upwind-differences alone with $c = 0.8$. (b) Semicircle test: same notation as in (a).

2.2. *Comparison with the REVFCT Method*

Apart from the specification of the antidiffusive flux the fourth-order scheme described above is almost identical to REVFCT [2]. For REVFCT the antidiffusive flux is defined as

$$\phi_{j+1/2} = \frac{(2+c^2)}{12}(\bar{\rho}_{j+1} - \bar{\rho}_j). \tag{14}$$

When we compare Eq. (8) with Eq. (14) we note that several terms are missing from Eq. (14). Since the antidiffusive flux is the difference between the high- and low-order fluxes, these terms have been incorporated into the low-order fluxes of REVFCT. These different low-order fluxes as used in REVFCT no longer guarantee positive results, and so REVFCT will not preserve positivity, as noted in Table 2 of Ref. [6].

2.3. *Comparison of Computational Times for Different Schemes*

In Table I we compare the times required for computing the different solutions shown in Figs. 1–3 on a Cyber 840 computer. We used Courant numbers from 0.2 to 10 for the fourth-order scheme and $c = 0.8$ for the upwind-difference scheme alone. We also present results for the LPE–Shasta scheme for $c = 0.2$ and 0.5. The flux limiter used was that of Boris and Book, except for one case where Zalesak's limiter was used for comparison. The number of implicit time steps is tabulated, as well as the number of upwind-difference steps taken within each implicit step.

TABLE I

Comparison of Computational Times for Various FCT Methods

Courant number	Time steps	Multiple low-order steps	Fourth-order (Boris–Book limiter)[a]	Fourth-order (Zalesak limiter)[a]	Upwind-differencing (alone)[a]	LPE–Shasta (Boris–Book limiter)[a]
0.2	800	1	19.9	30.7	4.5	7.9
0.5	320	1	8.0	12.2	1.9	3.1
0.8	200	1	5.0	7.6	1.1	—
1.6	100	2	1.7	3.1	0.9	—
2.5	64	4	1.5	2.4	1.0	—
3.2	50	4	1.2	1.9	0.8	—
4.0	40	5	1.1	1.6	0.8	—
5.0	32	7	1.1	1.5	0.8	—
6.4	25	8	0.9	1.3	0.7	—
8.0	20	10	0.9	1.2	0.7	—
10.	16	13	0.9	1.1	0.7	—

[a] Central processor times (seconds) on a Cyber 840.

3. Discussion

We have derived a highly accurate implicit finite-difference scheme that produces non-negative results for any Courant number. The freedom allowed in the choice of the size of the time step makes the algorithm very efficient. For example, the fourth-order scheme, with Boris and Book's limiter, and $c = 3.2$, gives results in one third of the time required to obtain comparable results using LPE–Shasta.

For values of Courant number $c < 2$ the fourth-order scheme is more accurate than any of the practical schemes described by Boris and Book [2]. As the Courant number increases the computational time decreases although, above $c = 2$ the average error also increases. As the Courant number approaches 10 the errors become greater than those obtained with other FCT algorithms. However, although the time taken becomes comparable to that for upwind differencing alone, the results are significantly better than for upwind differencing and remain non-negative. Indeed, for $c = 10$, 84% of the time goes into calculating the low-order solution. Thus using our method at $c = 3.2$, instead of upwind differencing, could give three times more accuracy for an expenditure of only 20% more computing time.

The semicircle test demonstrates that the algorithm can faithfully transport all density gradients, not just square waves. This test acts as an important check, since slight errors in coding can lead to any arbitrary waveform being transformed into a square wave.

For $c > 1$ we find no difficulty in using either Boris and Book's flux-limiter or that of Zalesak, since the antidiffusive fluxes are defined locally, irrespective of how far material moves across the cells. However, Zalesak's limiter allow one the freedom to choose any densities to set the maximum and minimum density limits. These densities are traditionally taken as the maximum and minimum of six density values defined at the mesh point and its nearest neighbours. Three density values are from the final solution last obtained, ρ^n, and three values are from the newly computed lower-order solution $\tilde{\rho}$. Thus for $c > 1$, if we use ρ^n values from the region from which the material is advected, we make the limiter sensitive to steep gradients in the original solution which should be transported, and so obtain a considerable improvement in the results. We can determine the region from which the three ρ^n values should be obtained by moving back, against the direction of flow, the number of mesh points equal to the integer value of c.

The fourth-order scheme used has many advantages over other possible high-order schemes. It gives the highest possible accuracy when using information from only three mesh points and has the advantage that it is stable when marched with $c > 1$, which makes it very efficient. Schemes using more than three mesh points have very complex coefficients [8], require more computation, special treatment at the boundaries, cannot be marched, and in many cases are not stable for $c > 1$.

Use of multiple upwind steps for $c > 1$ is a fast and accurate method of developing the low-order solution and this procedure guarantees stable positive results. Thus at high Courant numbers the fourth-order method acts as a high-order

correction to upwind differences. However, if a suitable implicit low-order scheme, which gives non-negative results, can be developed then further improvement in overall efficiency and accuracy will be achieved.

In gas discharges the electrons are the fastest moving species and therefore the value of c for electrons, c_e determines the maximum time step permissible. In a discharge both c_e and the electron density N_e can vary greatly from one point in the discharge to another [3]. Near the cathode, the value of c_e, for a given time step, can be high and the algorithm approximate, since N_e is very low and does not control the space–charge effects. Further from the cathode, for the same time step, the value of c_e is low and the algorithm very accurate, and this is where electrons accumulate and dominate the space–charge effects. Thus the algorithm we have developed is ideal for studying gas discharge problems.

4. Conclusions

We have succeeded in developing an implicit flux-corrected transport algorithm applicable for Courant numbers $c > 1$. This algorithm can give a threefold increase in overall speed over previous methods with the same order of accuracy, or give greater accuracy for the same computational effort, while returning strictly non-negative results. The algorithm is ideally suited to the solution of gas discharge problems where the electric field and electron density vary greatly throughout the discharge. Further development of the present algorithm will be necessary to adapt it for use on a non-uniform mesh and to include real diffusion in implicit form. Further work on improving the algorithm should concentrate on the development of a more efficient low-order scheme, since this is where most of the computational time is spent for high Courant numbers. Some effort could also be directed towards the development of a more efficient and perhaps more appropriate flux limiter. The clipping of sharp peaks that is typical of FCT algorithms could also be reduced if a better low-order scheme was used, but as yet none have been found that give sharper peaks and produce smooth results.

Acknowledgments

The authors acknowledge many helpful discussions with Dr. John Noye and thank him for his support. We also thank Vivienne Bowers, Dr. Ralph Loughhead, Dr. Noel Barton, and Dr. Gerry Haddad for their help in preparing this manuscript. The first author acknowledges the support from the Commonwealth Postgraduate Research Award Scheme and thank the CSIRO for travel support.

References

1. R. Morrow, *J. Comput. Phys.* **43**, 1 (1981).
2. J. P. Boris and D. L. Book, in *Methods in Computational Physics*, edited by J. Killeen (Academic Press, New York, 1976), p. 85.

3. R. MORROW, *Phys. Rev. A* **32**, 1799 (1985).
4. R. MORROW, *Phys. Rev. A* **35**, 1778 (1987).
5. G. PATNAIK, R. H. GUIRGUIS, J. P. BORIS, AND E. S. ORAN, *J. Comput. Phys.* **71**, 1 (1987).
6. J. P. BORIS AND D. L. BOOK, *J. Comput. Phys.* **20**, 397 (1976).
7. S. T. ZALESAK, *J. Comput. Phys.* **31**, 355 (1979).
8. J. NOYE, in *Computational Techniques and Applications: CTAC-85*, edited by J. Noye and R. May (North–Holland, Amsterdam, 1986), p. 137.
9. S. K. GODUNOV, *Math. Sb.* **47**, 271 (1959).
10. S. T. ZALESAK, Naval Research Laboratory, Washington, DC 20735, U.S.A., private communication (1982).
11. P. J. ROACHE, *Computational Fluid Dynamics* (Hermosa, Albuquerque, NM, 1972).