JOURNAL OF COMPUTATIONAL PHYSICS 108, 218-225 (1993)

Improved FCT Algorithm for Shock Hydrodynamics

DUŠAN ODSTRČIL

Geophysical Institute, Slovak Academy of Sciences, Dúbravská cesta 9, 842 28 Bratislava, Czechoslovakia

Received April 29, 1991; revised April 2, 1992

The new YDFCT algorithm is presented for the numerical solution of hydrodynamic flow problems with steep gradients. This algorithm is an explicit finite-difference scheme based on the flux-corrected transport technique and it is an improvement of the ETBFCT and the XDFCT algorithms. The erosion of sharp edges, the phase distortion of curved profiles and the terracing effect and velocity oscillations are eliminated or significantly reduced. These improvements are obtained by following action of the limiter at edges, more detailed linear numerical analysis of the difference scheme, and considering the velocity gradient in the diffusion and antidiffusion coefficients. Properties of the new algorithm are also demonstrated by the solution of three test problems.

1. INTRODUCTION

In the preceding paper [1], we have introduced the XDFCT algorithm for numerical solution of hydrodynamic flow with shocks. The XDFCT algorithm is an explicit finite-difference scheme based on the flux-corrected transport (FCT) technique. This approach is general and enables wide applications in hydrodynamics with steep gradients. No knowledge about the properties of the considered fluid is required in advance. This enables us to solve also problems where the equation of state is non-ideal (e.g., gas at high temperatures or gas with chemical reactions) or where more characteristic velocities exist (e.g., magneto-hydrodynamics).

The XDFCT algorithm enables, with respect to the original ETBFCT algorithm [2], to use twice the timestep with a comparable accuracy. This possibility significantly reduces the amount of computer time or enables us to perform computations on a finer difference mesh for the same amount of computer time. However, the price of this possibility is slight "erosion" of sharp edges of passively moving structures (e.g., convection of "square waves" or propagation of contact discontinuities). Further, the terracing effect is slightly more significant due to the use of timesteps with larger Courant numbers.

In this paper, we introduce the YDFCT algorithm as an improvement of the ETBFCT and XDFCT algorithms. YDFCT has the underlying difference scheme similar to XDFCT a full description of which is given in the preceding paper [1]. The new YDFCT algorithm removes the erosion effect, reduces the phase distortion, and suppresses the terracing effect in the numerical solution. These improvements are obtained by following action of the limiter at edges, more detailed linear numerical analysis of the difference scheme, and consideration of the velocity gradient in the diffusion and antidiffusion coefficients. Results of the solution of two convection problems, as well as an application to the interaction of two strong blast waves are also presented. We note that these results can be directly compared with the results obtained by the ETBFCT and XDFCT algorithms given in the preceding paper [1].

2. IMPROVEMENTS OF THE FCT ALGORITHM

Here we try to remove erosion of edges, reduce the phase distortion, and suppress the terracing effect which can occur in the numerical solution. All these improvements are verified using three test problems in the next section. The new improved algorithm is named as the YDFCT.

2.1. *Edge Erosion*

The possibility of using twice the timestep significantly reduces the amount of computer time. However, its price is a slight erosion near convected sharp edges. As can be seen in the figures shown in [1], this effect occurs for passively convected profiles with sharp edges and for the propagation of contact discontinuities. Shocks do not suffer by this effect due to their self-steepening mechanism.

Originally, we have assumed [1] this effect to be caused by the application of a strong limiter. For this purpose, we tried to use the less diffusive solutions as references for the limiter, however, without success. The real problem, in fact, consists in small antidiffusive fluxes which are not able to improve the transported-diffused solution near sharp edges. We try to illustrate this in a simplified situation shown in

0021-9991/93 $5.00

Fig. 1. In the case of the original ETBFCT algorithm, the diffusive fluxes are $f^d_{j+1/2} = \nu(\rho^o_{j+1} - \rho^o_j)$ and the antidiffusive fluxes are $f^a_{j+1/2} = \mu(\rho^t_{j+1} - \rho^t_j)$, where ρ^o is the initial value at the old time level, ρ^t is the transported value, ν and μ are the diffusion and antidiffusion coefficients, and j is the cell number of the grid.

For the XDFCT algorithm, the antidiffusive fluxes are $f^a_{j+1/2} = \mu(\rho^h_{j+1} - \rho^h_j)$ at the full step, where ρ^h is the solution obtained during the half step. Application of diffusive fluxes smoothes values of the solution and application of antidiffusive fluxes attempts to remove the introduced diffusion (with simultaneous increasing accuracy of the solution and control by the limiter to avoid non-monotonicity). As can be seen in Fig. 1, this is not possible at cell interfaces 3–4 and 6–7 because $\rho^h_{j+1} - \rho^h_j = 0$ here; thus $f^a_{j+1/2} = 0$. That is why the erosion of sharp edges begins. Because passively convected edges do not have a self-steepening mechanism (in contrast to shocks) this erosion is preserved throughout the computation.

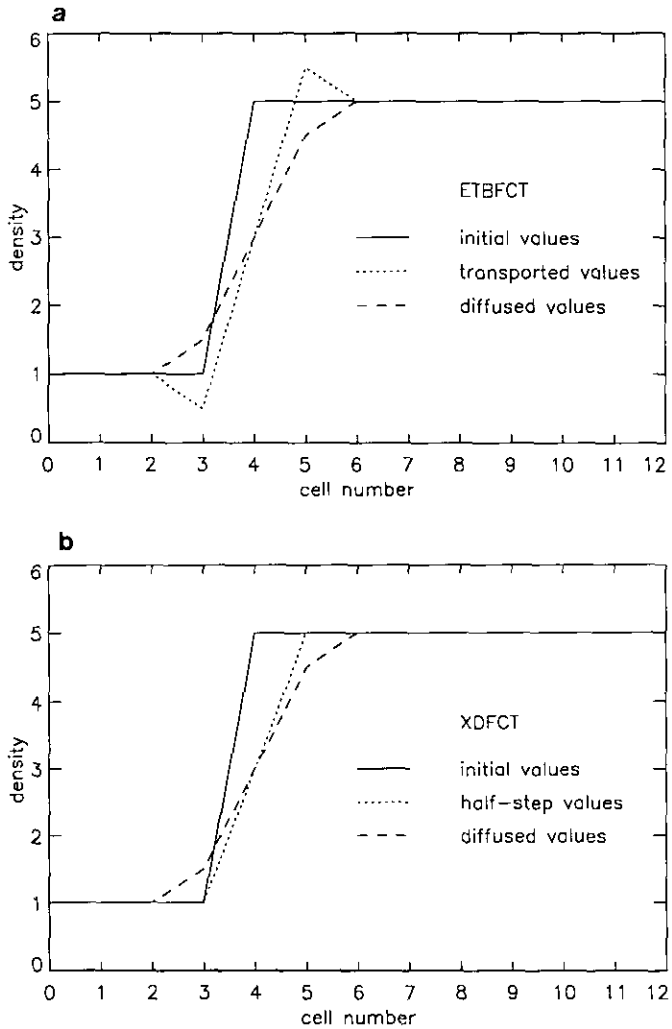

FIG. 1. *Simplified illustration of the solution procedure near a sharp* edge: (a) the antidiffusive fluxes $f^a_{j+1/2} = \mu(\rho^t_{j+1} - \rho^t_j)$ are non-zero and thus they can remove the introduced diffusion; (b) the antidiffusive fluxes $f^a_{j+1/2} = \mu(\rho^h_{j+1} - \rho^h_j)$ are zero and thus they cannot remove the introduced diffusion.

Taking into account the action of the limiter, we can improve the numerical solution at the edges. Using ρ^h in the antidiffusive fluxes is necessary to obtain a solution with monotonic properties and high amplitude and phase accuracy. This was shown by numerical analysis of the difference scheme [1]. On the other hand, as was shown above, using ρ^h cannot guarantee the removal of all excessive diffusion at the edges. Therefore, we propose to use antidiffusive fluxes,

$$f^a_{j+1/2} = \mu \max[(\rho^h_{j+1} - \rho^h_j), \tfrac{1}{2}(\rho^t_{j+1} - \rho^t_j)], \tag{1}$$

to overcome this conflict. This form preserves monotonicity and high accuracy of the numerical solution because $\rho^h_{j+1} - \rho^h_j$ is larger than $\frac{1}{2}(\rho^t_{j+1} - \rho^t_j)$ in normal situations. Further, this form avoids erosion of sharp edges because $\frac{1}{2}(\rho^t_{j+1} - \rho^t_j)$ is larger than $(\rho^h_{j+1} - \rho^h_j)$ in these extreme situations (often due to overshoots and undershoots in the transported solution ρ^t near sharp edges). Thus, the proposed form of the antidiffusive fluxes (1) preserves the optimized properties of the difference scheme and simultaneously enables us to avoid the erosion of edges. This form is used in the YDFCT algorithm.

2.2. *Phase Distortion*

Basic properties of the difference scheme and errors in the resulting solution can be studied by means of the linear numerical analysis. Details about such an analysis for the XDFCT algorithm are given in the previous paper [1]. More detailed analysis, by considering higher-order terms, shows that the XDFCT algorithm has, in fact, a sixth-order amplitude and fourth-order phase accuracy. This explains the slightly sharper steep gradients, as observed, for example, in the square wave convection.

For simplicity, the continuity equation

$$\frac{\partial}{\partial t}(\rho) + \frac{\partial}{\partial x}(\rho v) = 0, \tag{2}$$

where ρ is the density and v is the velocity, will be used in what follows. The solution of Eq. (2) by the FCT technique may be symbolically written as

$$\rho^n_j = \rho^0_j - \frac{f^t_{j+1/2} - f^t_{j-1/2}}{\Delta x} + \frac{f^d_{j+1/2} - f^d_{j-1/2}}{\Delta x} - \frac{f^a_{j+1/2} - f^a_{j-1/2}}{\Delta x}, \tag{3}$$

where f^t are the transport fluxes which reflect the real trans-

port of mass, f^d are the diffusive fluxes which introduce a numerical diffusion to the solution, and f^a are the antidiffusive fluxes which eliminate excessive numerical diffusion where possible [1]. Following Book [3], we try to improve the phase properties of the XDFCT algorithm by splitting the diffusive fluxes in two,

$$f^d_{j+1/2} = \nu(\rho^o_{j+1} - \rho^o_j) + \gamma(\rho^o_{j+1} - \rho^o_j), \tag{4}$$

where ν and γ are the diffusive coefficients. The first part is applied to the transported solution before its use for antidiffusive fluxes. The second part completes the transported-diffused solution as obtained originally. In the other words, instead of transported values, partially diffused transported values are used for the antidiffusive fluxes. This approach is, of course, possible only for the half step, because the full step uses different antidiffusive fluxes [1].

Linear numerical analysis of this modified difference scheme gives the amplification function

$$\begin{aligned} g = {} & 1 + 2(N-M)(\cos\beta - 1) \\ & - 2M(\gamma + \nu - \mu)(\cos 2\beta - 4\cos\beta + 3) \\ & + 2M\mu\gamma(\cos 3\beta - 6\cos 2\beta + 15\cos\beta - 10) \\ & - i\varepsilon\left[\left(1 + M + \frac{5M\mu}{2}\right)\sin\beta \right. \\ & \left. - \left(\frac{M}{2} + 2M\mu\right)\sin 2\beta + \frac{M\mu}{2}\sin 3\beta\right], \end{aligned} \tag{5}$$

the relative amplitude error,

$$\begin{aligned} A = {} & [2(N-M) - \varepsilon^2]\,\beta^2 + \left[-(N-M)^2 - \frac{N-M}{6}\right. \\ & \left. + 2M(\nu + \gamma - \mu) + \frac{\varepsilon^2}{3} - M\varepsilon^2\right]\beta^4 + O(\beta^6), \end{aligned} \tag{6}$$

and the relative phase error,

$$\begin{aligned} R = {} & \left[N - \frac{M}{2} - \frac{1}{6} - \frac{\varepsilon^2}{3}\right]\beta^2 + \left[(N-M)^2 + \frac{M}{2}(N-M)\right. \\ & + M\left(\nu + \gamma - \frac{\mu}{2}\right) + (M-2N)\left(\frac{1}{8} + \frac{\varepsilon^2}{2}\right) \\ & \left. + \frac{\varepsilon^2}{6} + \frac{\varepsilon^4}{5} + \frac{1}{120}\right]\beta^4 + O(\beta^6), \end{aligned} \tag{7}$$

where $\varepsilon = v\,\Delta t/\Delta x$. The second-order error terms vanish when

$$N - M = \frac{\varepsilon^2}{2} \tag{8}$$

and

$$\begin{aligned} N &= \frac{1}{3} + \frac{\varepsilon^2}{6}, \\ M &= \frac{1}{3} - \frac{\varepsilon^2}{3}. \end{aligned} \tag{9}$$

Specifying the same coefficients,

$$\begin{aligned} \nu &= \frac{1}{6} + \frac{\varepsilon^2}{12}, \\ \mu &= \frac{1}{6} - \frac{\varepsilon^2}{24}, \end{aligned} \tag{10}$$

as used for the ETBFCT and XDFCT at the half step [1], the remaining relative amplitude error is

$$A = \left[\gamma\left(\frac{2}{3} - \frac{2\varepsilon^2}{3}\right)\right]\beta^4 + O(\beta^6), \tag{11}$$

and the remaining relative phase error is

$$R = \left[\gamma\left(\frac{1}{3} - \frac{\varepsilon^2}{3}\right) + \frac{\varepsilon^2}{144} - \frac{\varepsilon^4}{720} - \frac{1}{180}\right]\beta^4 + O(\beta^6). \tag{12}$$

There remains one free parameter which can be chosen so as to build some desirable property into the algorithm (this is the reason for introducing γ). Letting $\gamma = 0$ as in the XDFCT [1] yields an algorithm with sixth-order amplitude accuracy. The choice

$$\gamma = \frac{1}{60} + \frac{\varepsilon^2}{240} \tag{13}$$

yields an algorithm with sixth-order phase accuracy. We note that the coefficients for the half step use the parameter $\varepsilon = \Delta t/\Delta x$, i.e., ε is specified for the full step. If we use $\varepsilon = \Delta t/(2\,\Delta x)$ as is computed at the half step, then we have $\nu = 1/6 + \varepsilon^2/3$, $\mu = 1/6 - \varepsilon^2/6$, and $\gamma = 1/60 - \varepsilon^2/60$ instead of the above given forms.

As was stated by Boris and Book [4], phase properties of an algorithm are more important than the amplitude ones. Damping generally leaves the long wavelengths untouched while removing the very short ones. Since these short wavelength harmonics of the solution generally suffer the most dispersion anyway, damping at the short wavelengths can sometimes reduce the overall errors in conjunction with dispersion. The phase properties are practically important because phase errors grow secularly when the velocity is predominant in one direction. The difference in position between the correct phase front and the numerically computed one increases in time. From the above it follows that

if we have a choice to reduce either amplitude or phase errors to the sixth order than the better choice is the second one. Therefore, we have chosen the split diffusion with the diffusive coefficients given above to reduce phase errors in the YDFCT algorithm.

When realizing such an algorithm, the split diffusion can be applied either at the half step or at the full step. The first possibility is straightforward; however, the solution loses the fourth-order amplitude accuracy at the half step and the values are more diffusive. These half step values are used for the centered transport fluxes and the source terms at the full step and thus they can negatively influence the resulting solution. The latter possibility uses no split diffusion at the half step; i.e., the algorithm is the same as the ETBFCT with its fourth-order amplitude and phase accuracies. The source terms and the transport fluxes are computed for the given equation using these half-step values. Then, just before the full-step completion, the half-step values are corrected by the diffusion associated with the γ coefficient. Finally, the full step is completed; i.e., the algorithm used is the same as the XDFCT, but the resulting solution has fourth-order amplitude and sixth-order phase accuracies. We have performed some test computations to show that application of the split diffusion at the full step yields slightly more accurate results. Therefore, we have chosen this possibility to realise the split diffusion in the YDFCT algorithm.

2.3. *Terracing Effect*

Although the computed profiles of hydrodynamic flow are accurate and physically correct on average, they can exhibit numerous flat shelves and accompanying sharp drops. This phenomenon is known as the "terracing effect" and occurs especially in regions with changing velocity. It is clear that the diffusion and antidiffusion coefficients must take into account gradients of the velocity. In the next, we try to derive their forms empirically and verify them by test calculations.

The system of partial differential equations describing the 1D HD ideal gas problems written in conservative form is [1]:

$$\begin{aligned} \rho^{\circ} + (\rho v)' &= 0, \\ (\rho v)^{\circ} + (\rho v^2)' + p' &= 0, \\ E^{\circ} + (Ev)' + (pv)' &= 0, \end{aligned} \tag{14}$$

where ρ is the mass density $[\mathrm{kg}\cdot\mathrm{m}^{-3}]$, v is the velocity $[\mathrm{m}\cdot\mathrm{s}^{-1}]$, p is the thermal pressure $[\mathrm{N}\cdot\mathrm{m}^{-2}]$, E is the total energy density $[\mathrm{J}\cdot\mathrm{m}^{-3}]$, and dot ($^{\circ}$) and prime ($'$) denote the time and space derivations, respectively.

Instead of performing a complex numerical analysis, we note that the linear numerical analysis uses the same constant velocity for each equation. This means that only a portion of the information about the actual distribution of the velocity is used. In addition, the energy is transported with the velocity

$$v^e = v + \frac{pv}{E} \tag{15}$$

which is parallel to and somewhat larger than the ordinary flow velocity v. The consistency between the velocity actually solved by the system of Eqs. (14) and its form used for the numerical analysis could be achieved using the following "modified velocity"

$$v^m = v + v^g, \tag{16}$$

where v^g is the "gradient velocity term" which is

$$v^g = \rho v'/\rho' \tag{17}$$

for the equation of continuity,

$$v^g = \frac{\rho v v'}{(\rho v)'} \tag{18}$$

for the equation of motion, and

$$v^g = \frac{Ev' + (pv)'}{E'} \tag{19}$$

for the equation of total energy. Thus, we should compute the diffusive and antidiffusive coefficients using the above "modified velocities" before solving each equation (originally, these coefficients were computed once before solving the system of equations).

Because, in certain situations, the "gradient velocity term" can reach high values, it is necessary to limit the "modified velocities." The maximum modified velocity which is used for the diffusion and antidiffusion coefficients must be less than $\Delta x/\Delta t$ as is required by the stability condition. In addition, we impose the limit to the gradient velocity term

$$v^g < \frac{|v| + c}{2}, \tag{20}$$

where c is the characteristic velocity such that $|v| + c$ represents the maximum propagation velocity of information. For hydrodynamic problems c is the sound velocity. This condition enables us to stabilize the numerical solution (suppress non-physical oscillations) in regions with low flow and high characteristic velocities. At the same time, it avoids too large a numerical diffusion at regions with high flow and low characteric velocities. We note that the maxi-

mum propagation velocity of information is also used for timestep control during the whole computation. Modified velocities and their limits, as given above, are used in the YDFCT algorithm.

3. RESULTS OF THE TEST PROBLEMS

We have chosen two convection problems to show the basic properties of our improvements and one shock hydrodynamics problem to show their properties under complex circumstances. The first test problem consists of advecting a square wave 20 grid points wide for 800, 400, and 200 time steps at the Courant numbers of 0.2, 0.4, and 0.8, respectively. The second test problem consists of advecting a semicircle 30 grid points wide for 300, 150, and 75 time steps at the Courant numbers of 0.2, 0.4, and 0.8, respectively. All these calculations were performed on a uniform grid of 100 points with periodic boundary conditions. The third test problem consists of the interaction of two strong blast waves generated in a shock tube with reflecting boundaries filled initially by a gas with uniform density and with two opposing pressure jumps of 10^5 and 10^4 ratio. Calculations were performed on a uniform grid of 200 and 800 grid points.

Results of these test problems can be directly compared with the results obtained by the ETBFCT and XDFCT which are given in the previous paper [1]. All computations were performed on the PC-AT with the use of single precision (32 bits in a word). We note that the results may be slightly different due to the different representation of floating point numbers with respect to the previously used IBM computer. Therefore, we have repeated the computations of convection problems and determined their relative errors for the all compared algorithms.

3.1. *Convection of a Square Wave*

The results of the square wave convection problem obtained by the YDFCT are shown in Fig. 2. It can be seen that there is a distinct improvement, with respect to the XDFCT, by the complete elimination of erosion near the edges of the square wave. The average absolute errors (0.0400, 0.0334, and 0.0241) are comparable to the errors in solutions obtained by the ETBFCT (0.0388, 0.0369, and 0.0417) and by the XDFCT (0.0365, 0.0379, and 0.0276). The XDFCT with the modified antidiffusive fluxes (as the YDFCT uses) has these errors (0.0312, 0.0302, and 0.0235) that are slightly smaller due to its formally higher amplitude accuracy as shown in Section 2.2.

3.2. *Convection of a Semicircle*

The results of the semicircle convection problem obtained by the YDFCT are shown in Fig. 3. It can be seen that there is a distinct improvement, with respect to both the ETBFCT and the XDFCT, by the significant elimination of distortion of the curved profile. The average absolute errors (0.0113, 0.0097, and 0.0071) are smaller than the errors in the solutions obtained by the ETBFCT (0.0208, 0.0198, and 0.0163) and by the XDFCT (0.0186, 0.0173, and 0.0117). The XDFCT with the modified antidiffusive fluxes (as the YDFCT uses) has these errors (0.0122, 0.0112, and 0.0075) which are slightly larger, with significant distortions of the curved profiles, due to its formally lower phase accuracy as shown in Section 2.2.

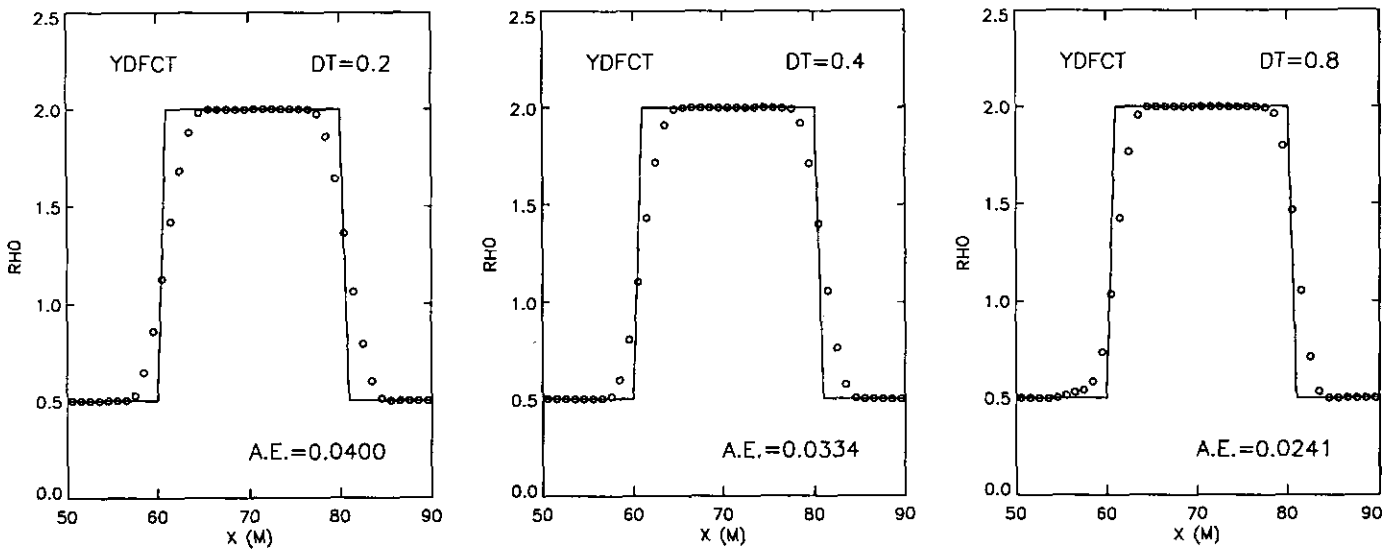

FIG. 2. Solution of the square wave convection problem by the YDFCT at 160 s. The solution without edge erosion and an increase of the overall accuracy with respect to the ETBFCT or XDFCT can be seen.

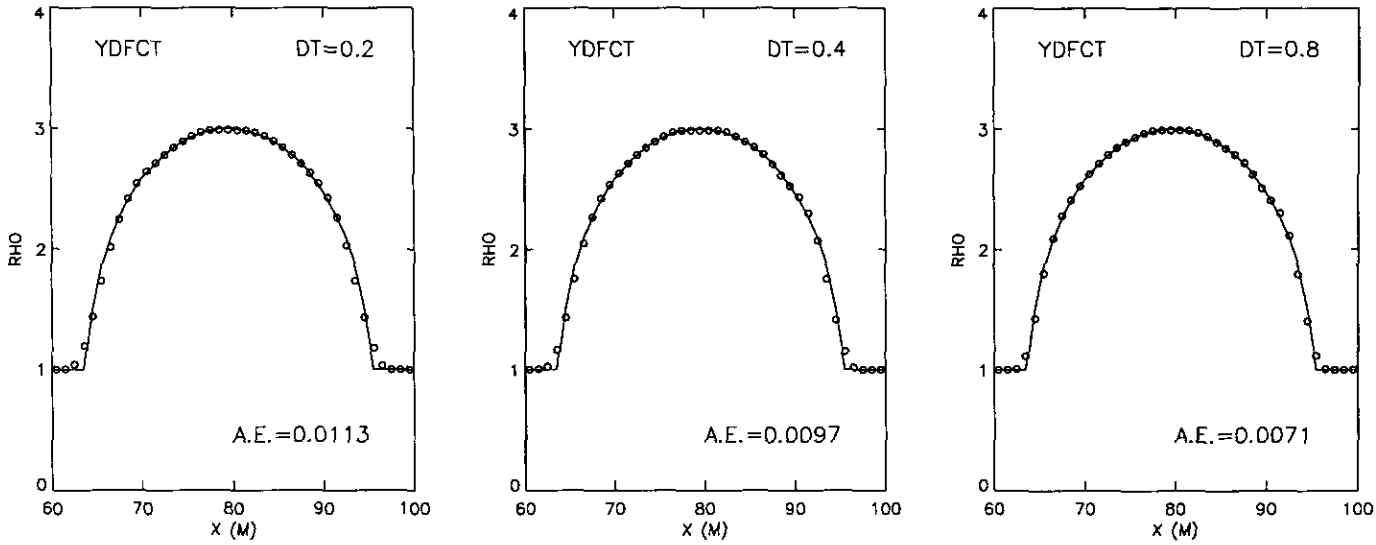

FIG. 3. Solution of the semicircle convection problem by the YDFCT at 60 s. Significant elimination of phase distortion of the curved profiles and an increase of the overall accuracy with respect to the ETBFCT or XDFCT can be seen.

3.3. *Interaction of Two Blast Waves*

Figure 4 shows the distribution of the density at times 0.026 and 0.038 s for the solution on 200 and 800 point grids. Positions of the shocks and contact discontinuities are calculated very accurately. Further, the shocks are sharply resolved with a width of one or two cells only. However, the YDFCT gives more smeared contact discontinuities. Especially, it is difficult to accurately resolve the contact discontinuity which is overtaken by a reflected rarefaction wave with a steep density slope on a 200-point grid. Its position is approximately at 0.5 and 0.6 m at times 0.026 and 0.038 s, respectively. Finally, the YDFCT has problems to represent the compressed density slab on a 200-point grid. Its position is approximately between 0.75 and 0.8 m at time 0.038 s. All these results are comparable with the results obtained by the ETBFCT and XDFCT previously.

The YDFCT eliminates the terracing effect in density profiles which occurs in solutions obtained by the ETBFCT and XDFCT. Further, it avoids the non-physical ripple of the density at 0.7 m at 0.038 s which is significant for the ETBFCT and which still is apparent for the XDFCT. Finally, the YDFCT suppresses the velocity oscillations as can be seen in Fig. 5. We note that the XDFCT gives even more oscillations than the ETBFCT due to the double Courant numbers used in computation. If the XDFCT is run with the same timestep, these oscillations are smaller and comparable with the ETBFCT ones. Some oscillations remain at the point of the contact discontinuity (approximately at 0.45 m at 0.026 s). Although some other oscillations can persist in velocity profiles they are very small and the YDFCT represents a distinct improvement.

On the other hand, the YDFCT gives slightly lower density peaks and slightly more diffusion of the less curved profiles. Examples of the latter property can be seen on the density profile between 0.5 and 0.65 m at time 0.026 s in Fig. 4 and on the velocity profile between 0.0 and 0.4 m at time 0.026 s in Fig. 5 (both on a 200-point grid). This is the price for almost complete elimination of the terracing effect and velocity oscillations.

Finally, we should mention that despite a slight improvement these test results are still not so impressive as those obtained by the Godunov-type algorithms [5]. This difference can be decreased when FCT computations are performed on finer meshes, i.e., when comparisons are made for similar computational efforts. Further, the FCT approach is general and no knowledge about the properties of the considered fluid is required in advance. This also enables us to solve problems where the equation of state is non-ideal (e.g., gas at high temperatures or gas with chemical reactions) or where more characteristic velocities exist (e.g., magnetohydrodynamics).

4. CONCLUSIONS

In this paper, we have considered the problem of reducing the errors in numerical solutions of hydrodynamic flow problems with steep gradients. The phase and amplitude errors of the difference scheme were minimized by proper use of the diffusive and antidiffusive coefficients. The resulting fourth-order amplitude and sixth-order phase accuracies enable sufficient resolution of steep gradients and significantly reduce phase distortion of curved profiles. Erosion of sharp edges was removed by following the action of the limiter and using the modified antidiffusive fluxes

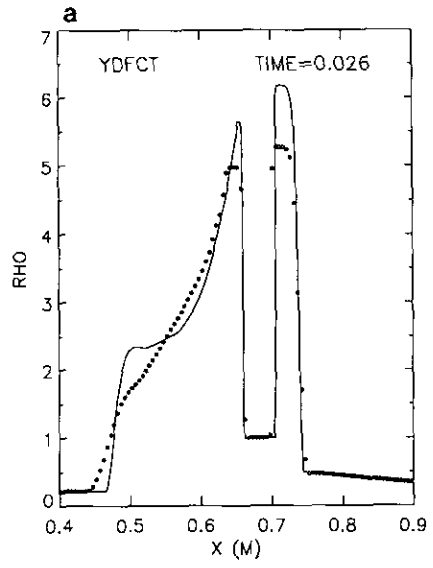

FIG. 4. Solution of the two interacting blast waves problem by the YDFCT on a 200-point grid (circles) and on an 800-point grid (solid line) at 0.026 s (a) and 0.038 s (b) for the mass density.

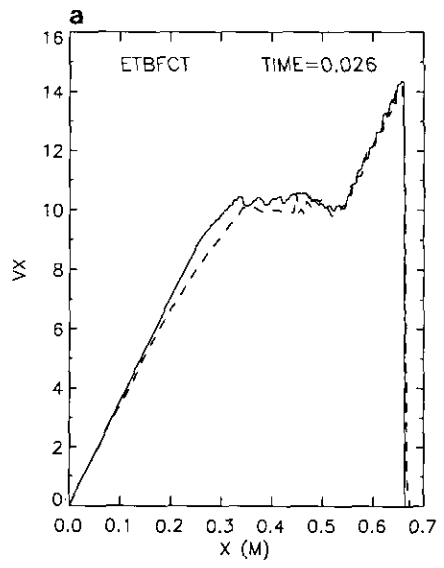

FIG. 5. Solution of the two interacting blast waves problem by the ETBFCT (a) and YDFCT (b) on a 200-point grid (dashed lines) and on an 800-point grid (solid lines) at 0.026 s for the velocity.

here, if necessary. The terracing effect was removed or significantly suppressed by considering of the velocity gradient in the diffusion and antidiffusion coefficients. This modification also suppresses velocity oscillations that could otherwise occur, for example, in rarefaction waves. Because we apply increased diffusion, gradually curved profiles can be smoothed more. Generally, these results represent a distinct improvement with respect to the original ETBFCT algorithm or to its modification of the XDFCT algorithm.

Computation of the two-blast interaction test problem (on a 200-point grid, up to 0.038 s) requires 1208, 655, and 883 s for the ETBFCT, the XDFCT, and the YDFCT algorithms, respectively. The XDFCT and the YDFCT both use twice the timestep that the ETBFCT does. Therefore, the XDFCT is approximately twice as fast as the ETBFCT because it performs the same amount of computations. The YDFCT uses the modified antidiffusive fluxes, split diffusive fluxes, and the modified velocity term for computations of the diffusion and antidiffusion coefficients before solving each equation. Therefore, the YDFCT is slower than the

XDFCT; however, it is still significantly faster than the ETBFCT. We conclude that the increased computation requirements with respect to the XDFCT are more than compensated by its improved properties and that the YDFCT is faster and gives generally better solutions than the ETBFCT. Finally, we want to point out that our FCT algorithm is sufficiently robust and independent of any specific system of partial differential equations. We note that the maximum propagation velocity of information, which is used to impose limits on the modified velocity term, is also used for timestep control during the whole computation.

REFERENCES

1. D. Odstrčil, *J. Comput. Phys.* **91**, 71 (1990).
2. D. L. Book, J. P. Boris, and S. T. Zalesak, in *Finite-Difference Techniques for Vectorized Fluid Dynamics Calculations*, edited by D. L. Book (Springer-Verlag, New York, 1981).
3. D. L. Book, in *Computational Techniques and Applications: CTAC-83*, edited by J. Noye and C. Fletcher (Elsevier, North-Holland, New York, 1984).
4. J. P. Boris and D. L. Book, *J. Comput. Phys.* **20**, 397 (1976).
5. P. Woodward and P. Colella, *J. Comput. Phys.* **54**, 115 (1984).