JOURNAL OF COMPUTATIONAL PHYSICS 108, 27–37 (1993)

A FCT Method for Staggered Mesh

P. M. VELARDE

Instituto de Fusión Nuclear, Madrid, Spain

Received April 25, 1990; revised September 21, 1992

A finite difference method on a staggered mesh for shock hydrodynamics with diffusion controlled via flux corrected transport is described. The algorithm is conservative, free streaming invariant and well behaved around shocks. This method is second order in accuracy in the worse case with several third- and fourth-order features. The algorithm is designed in Lagrangian coordinates and an arbitrary mesh can be used when remapping with piecewise parabolic method.

1. INTRODUCTION

Lagrangian finite difference codes have been succesfully used over many years for hydrodynamic simulations with strong shocks. The artificial viscosity introduced is usually a first- or second-order [1] diffusion term, with high enough coefficients chosen to assure low amplitude oscillations behind shocks. This leads to excessive spreading of the shock profile. To reduce the amount of added numerical diffusion, while maintaining the positivity of the algorithm, flux corrected transport (FCT) [2, 3] is often used. The resulting scheme is less sensitive to the numerical value of the diffusion coefficient, it is higher in order and, in general, more accurate. This article introduces FCT in classical Lagrangian codes with a staggered mesh. Unfortunately, attempts to use FCT directly have not worked properly for all the cases we have tested. With FCT alone, an explicit diffusion has to be introduced, which fails the FCT criterion of no new maximum or minimum when a staggered mesh is used. This explicit diffusion can be chosen to be second or third order away from shocks. Shock detectors are used to determine when to apply the explicit diffusion. This is a common practice in several codes. The way to determine the low order scheme is to assure damping of oscillations in smooth flow and in contact discontinuities. Simple coefficients, such as in Ref. [3], are not suitable for Lagrangian equations, so the Zalesak method is used with appropriate diffusion coefficients. The high order scheme is designed to produce a minimum computing time overhead. The final algorithm must be conservative and free stream invariant (and with some other obvious symmetries) both in the Lagrangian and in the remapping phase. Next we describe such a Lagrangian FCT staggered-mesh scheme (RMFCT) and we apply the algorithm to 1D gasdynamics equations in Lagrangian form. The Eulerian solution is obtained from the Lagrangian one by applying a high order remapping step, such as in the piecewise parabolic method (PPM) [4] scheme. Contrary to the original formulation, we make use of FCT for the non-convective terms. Moreover, we design RMFCT to give the correct results in Lagrangian coordinates, where the remapping phase is a different problem whose aspects will not be treated in detail in this article.

The hybridization procedure requires low and high order schemes and this article is divided accordingly. Section 2 defines the low order scheme for smooth flow, obtaining the scaling of the diffusion coefficients. Section 3 treats the additional diffusion needed near the shocks. This diffusion introduces enough spreading to limit strongly the oscillations. Section 4 describes the high order algorithm and the overall method. A simpler alternative to the PPM remapping scheme is pointed out. Tests are analysed in the last section. The covergence rate for each test is given in the last section as well.

2. LOW ORDER FLUX

Let us consider the one-dimensional Lagrangian equations in planar geometry $\partial_t V = \partial u$, $\partial_t u = -\partial P$, and $\partial_t E = -\partial uP$, being V, u, P, and $E = e + \frac{1}{2}u^2$ the specific volume, velocity, pressure, specific total energy, and e the specific internal energy, with $\partial_t \equiv \partial/\partial t$ and $\partial \equiv \partial/\partial \xi$ and ξ the mass coordinate. We write these equations as

$$\partial_t U + A\,\partial U = 0 \tag{1}$$

with

$$A = \begin{pmatrix} 0 & -1 & 0 \\ a & 0 & b \\ 0 & P & 0 \end{pmatrix}$$

0021-9991/93 $5.00

and $U = {}^t(V, u, e)$. FCT is independently applied to each conservation law. That simplifies the method, but the flux limitation on one equation could produce oscillations in another equation. To prevent this problem, diffusive fluxes will be obtained from the scalar equations giving an appropriate scaling of the diffusive coefficients and the form of the low order scheme. The transformation of the Lagrange equations into three independent scalar equations [5] is performed by locally diagonalising the matrix A of the Lagrangian system 1 by means of the matrix R composed by the right eigenvectors of A. Explicitly,

$$R = \begin{pmatrix} 1 & b & 1 \\ c & 0 & -c \\ -P & -a & -P \end{pmatrix}$$

$$R^{-1} = \frac{1}{2c^2}\begin{pmatrix} -a & c & -b \\ 2P & 0 & 2 \\ -a & -c & -b \end{pmatrix}$$

$$\Lambda = \operatorname{diag}(-c, 0, c) = R^{-1}AR$$

with $a = \partial P/\partial V$ and $b = \partial P/\partial e$. The transformed of ∂U is obtained from

$$\partial\alpha = R^{-1}\,\partial U = \frac{1}{2c^2}\begin{pmatrix} -a\,\partial V + c\,\partial u - b\,\partial e \\ 2P\,\partial V + 2\,\partial e \\ -a\,\partial V - c\,\partial u - b\,\partial e \end{pmatrix},$$

where c is the Lagrangian sound speed defined as $c^2 = bP - a$. If $\partial_t R = 0$ then the equation for α is $\partial_t\alpha + \partial\alpha = 0$. The low order scheme is the upwind flux associated to the signals with velocities $\pm c$, 0. By transforming back to the original representation we obtain the diffusive fluxes for vector U

$$F = \frac{1}{2}\,\delta t R\,|\Lambda|\,\partial\alpha = \frac{\delta t}{2c}\begin{pmatrix} -a\,\partial V - b\,\partial e \\ c^2\,\partial u \\ aP\,\partial V + bP\,\partial e \end{pmatrix}.$$

The finite difference formulation of ∂U is $(U_{i+1} - U_i)/\delta_{i+1/2}$, where $\xi_{i+1/2}$ is the mass coordinate of the point $x_{i+1/2}$, $\delta_i = \xi_{i+1/2} - \xi_{i-1/2}$, $\xi_i = \frac{1}{2}(\xi_{i+1/2} + \xi_{i-1/2})$, $\delta_{i+1/2} = \frac{1}{2}(\delta_i + \delta_{i+1})$. The fluxes for each U component are

$$F^{(V)}_{i+1/2} = \nu_{i+1/2}\,\delta t c_{i+1/2}(V_{i+1} - V_i - \beta_{i+1/2}(e_{i+1} - e_i)) \quad (2)$$

$$F^{(u)}_i = \mu_i\,\delta t c_i(u_{i+1/2} - u_{i+1/2}) \quad (3)$$

$$F^{(e)}_{i+1/2} = -P_{i+1/2}F^{(V)}_{i+1/2}$$

with the coefficients ν and β given by

$$\nu = -\frac{a}{2c^2} = -\frac{1}{2c^2}\frac{\partial P}{\partial V}$$

$$\beta = b/a = -\frac{\partial P/\partial e}{\partial P/\partial V}$$

and $\mu = \frac{1}{2}$. For a perfect gas equation of state $(P = (\gamma - 1)\,e/V)$ ν and β are

$$\nu = 1/2\gamma$$

$$\beta = V/e.$$

The simplest way to determine these coefficients is assuming $\nu_{i+1/2}$ to be a constant and

$$\beta_{i+1/2} = (V_{i+1} + V_i)/(e_{i+1} + e_i).$$

Equations (2)–(3) define the diffusive fluxes to be added to the high order fluxes in order to lead the low order scheme. Since the velocity in the flux of the energy equation is the corrected flux of the specific volume equation, no diffusive flux seems to be needed in this energy equation. However, if the flux of total energy is composed with the fluxes of V and u, the entropy will be calculated without damping (up to second order) with the low order scheme. To show that, we suppose V_i^1 and E_i^1 (temporal updated values of V_i and E_i) are obtained from

$$V_i^1 = V_i + \lambda_i(F_{i+1/2} - F_{i-1/2}) \quad (4)$$

$$E_i^1 = e_i - \lambda_i(P_{i+1/2}F_{i+1/2} - P_{i-1/2}F_{i-1/2}). \quad (5)$$

Up to first order we can write $V_i \approx V + \delta V_i$, $E_i \approx E + \delta E_i$, $F_{i+1/2} \approx F + \delta F_{i+1/2}$, and $P_{i+1/2} \approx P + \delta P_{i+1/2}$. Because of the translation invariance, we can take $u = F = 0$ and then $\delta E = \delta e$. Multiplying (4) by P and adding (5) results in

$$\delta e_i^1 + P\,\delta V_i^1 = \delta e_i + P\,\delta V_i. \quad (6)$$

The change in entropy S_i can be expressed by $\delta S_i \propto \delta e_i + P\,\delta V_i$ and, according to (6), $S_i^1 = S_i$ up to second order. In order to include some diffusion of the entropy at low order, we add a flux to the internal energy equation:

$$F^{(S)}_{i+1/2} = \nu_{i+1/2}\lambda_{i+1/2}c_{i+1/2} \times ((e_{i+1} - e_i) + \tfrac{1}{2}(P_{i+1} + P_i)(V_{i+1} - V_i)). \quad (7)$$

This flux spreads contact discontinuities whenever the high order algorithm produces peaks in entropy. It is observed in the simulations that the value of μ can be lower

than 0.5. In all the test performed we take $\mu = \nu = 0.18$ ($\approx 1/4\gamma$ for $\gamma = 1.4$) and $c_{i+1/2} = \min(c_i, c_{i+1})$. With this choice the diffusion coefficients depend linearly on the time step δt, where the amount of diffusion introduced between two fixed times is almost constant. That produces results with slight dependence on the time step. The low order scheme is the addition of high order and dissipative fluxes multiplied by the mesh spacing.

Similar expressions are obtained from Roe's linearization [6] of the Riemann problem. In that case, the coefficients are the same that those previously calculated with $a = -\frac{1}{2}(\gamma - 1)(e_i + e_{i+1})/(V_i V_{i+1})$, $b = \frac{1}{2}(\gamma - 1)(V_i + V_{i+1})/(V_i V_{i+1})$, and $P = \frac{1}{2}(P_i + P_{i+1})$, changing the value of ν.

3. ADDITIONAL DIFFUSION

With the spatial centering chosen for V, u, and e a shock wave cannot be represented monotonically in one zone, nor even as a initial condition. Because FCT is independently applied to each conservation law, an additional term is needed for spreading any shock wave on two or three zones. First we define a shock detector f_i that is at least second order in smooth flow and one near shocks. Useful detectors are

$$f_i^{(1)} = |P_{i+1} - 2P_i + P_{i-1}|/(P_{i+1} + 2P_i + P_{i-1}) \approx \frac{1}{4P}\delta^2 \frac{\partial^2 P}{\partial \xi^2}$$

or

$$f_i^{(2)} = \max(0, \min(1, 2(P_{i+1} - P_{i-1})/(P_{i+2} - P_{i-2}) - 1)) \approx -\frac{1}{2}\delta^2 \frac{\partial^3 P/\partial \xi^3}{\partial P/\partial \xi}.$$

The former equation is that for adaptive diffusion [7] and the latter one, more selective on shocks, is similar to the one used in the PPM scheme [4]. When the flow is in expansion we explicitly fix $f_i = 0$. In fact, f_i can be dropped whenever weak shock waves are present. In all the performed tests the results do not change if we set $f_i = 0$ when $u_{i-1/2} - u_{i+1/2} \leqslant 0.1 C_i$. Both f_i coefficients work out well in our case but the first one is better when two shocks collide and it is simpler to implement. The additional diffusion coefficient, used in the examples of Section 5, is fixed to

$$g_i = \tfrac{3}{4} f_i^{(1)} \rho_i (2C_i + \max(0, u_{i-1/2} - u_{i+1/2})), \tag{8}$$

where C_i is the sound speed. Similar results are obtained with $f_i^{(2)}$ when the coefficient is chosen:

$$g_i = 2 f_i^{(2)} \rho_i \max(0, u_{i-1/2} - u_{i+1/2}).$$

Finally, the actual coefficient for the velocity equation is

$$g_i^{(u)} = \max(g_{i-1}, g_i, g_{i+1}) \tag{9}$$

and for the specific volume equation

$$g_{i+1/2}^{(V)} = \min(g_i^{(u)}, g_{i+1}^{(u)}) \tag{10}$$

where $g_{i+1/2}^{(e)} = g_{i+1/2}^{(V)}$. This diffusion term is added to the low and high order fluxes before FCT is applied. With this shock detector, the continuous compression errors [8] are reduced up to second order or they are not present at all.

4. HIGH ORDER FLUXES

The first version of the algorithm was formulated with a low phase error scheme with second order in space and time. This type of algorithms are very unstable, where the correction coefficients are $-\frac{1}{12}$ times the diffusion of fluxes and require a special time splitting. In consequence they are a strong test on the procedure of hybridization. Details and results of this method have been described elsewhere [9]. In order to increase the accuracy of the method, we composed a third-order approximation to the fluxes on an arbitrary mesh. It is not possible to do that by a Taylor expansion; furthermore, there is inconsistency between the requested point values (fluxes) and the data of the averaged magnitudes (mass, momentum, and energy). In fact, the right procedure is to obtain a third-order approximation of the fluxes (velocity and pressure) with the mean values of V, u, and e [4, 10]. With the integral values, $\sum V_i \delta_i$, $\sum e_i \delta_i$, and $\sum u_{i-1/2} \delta_{i-1/2}$, we interpolate a third-order polynomial around ξ_i for pressure and around $\xi_{i-1/2}$ for velocity. The derivative of each polynomial in ξ_i and $\xi_{i-1/2}$ respectively gives the fluxes with third-order approximation. These values are

$$\bar{u}_{i-1/2} = u_{i-1/2} - \frac{1}{\delta_{i-2} + 2\delta_{i-1} + 2\delta_i + \delta_{i+1}} \left[\delta_{i-1}\delta_i \times \left(\frac{u_{i+1/2} - u_{i-1/2}}{\delta_{i-1} + 2\delta_i + \delta_{i+1}} - \frac{u_{i-1/2} - u_{i-3/2}}{\delta_{i-2} + 2\delta_{i-1} + \delta_i} \right) + (\delta_i - \delta_{i-1}) \left(\frac{2\delta_{i-1} + \delta_{i-2}}{\delta_{i-1} + 2\delta_i + \delta_{i+1}} (u_{i+1/2} - u_{i-1/2}) + \frac{2\delta_i + \delta_{i+1}}{\delta_{i-2} + 2\delta_{i-1} + \delta_i} (u_{i-1/2} - u_{i-3/2}) \right) \right] \tag{11}$$

$$\bar{P}_i = P_i - \frac{\delta_i^2}{4(\delta_{i-1} + \delta_i + \delta_{i+1})} \times \left(\frac{P_{i+1} - P_i}{\delta_i + \delta_{i+1}} - \frac{P_i - P_{i-1}}{\delta_{i-1} + \delta_i} \right). \tag{12}$$

These values are the high order fluxes. For an uniform mesh, with $Df_i \equiv f_{i+1} - 2f_i + f_{i-1}$, the $\bar{u}_{i+1/2}$ and $\bar{P}_i$ are reduced to

$$\bar{u}_{i+1/2} = u_{i+1/2} - \tfrac{1}{24} Du_{i+1/2}$$
$$\bar{P}_i = P_i - \tfrac{1}{24} DP_i,$$

giving fourth-order accuracy. Because the pressure is calculated from averaged values of the density and internal energy, this function is actually second-order accurate. For the energy equation, values for the pressure at zone boundaries are requested. These values must be consistent with the definition of kinetic energy so that the overall algorithm should be free-stream invariant. The second-order option is

$$P_{i+1/2} = (\delta_i \bar{P}_{i+1} + \delta_{i+1} \bar{P}_i)/(\delta_i + \delta_{i+1}) \tag{13}$$
$$E_i = e_i + \tfrac{1}{4}(u^2_{i+1/2} + u^2_{i-1/2}) \equiv e_i + K_i. \tag{14}$$

Finally, the energy equation is written as

$$e^1_i = e_i - (K^1_i - K_i) - \frac{\delta t}{\delta_i}(P_{i+1/2}\bar{u}_{i+1/2} - P_{i-1/2}\bar{u}_{i-1/2}).$$

The full Lagrangian step is described in the Appendix. It is worthwhile to point out that in the interpolation with Eqs. (11)–(12) no monotonicity constraint is applied, and we only need point values and not the exact distribution. The cpu time consumed by this step is almost negligible and makes its implementation in full two-dimensional codes easier.

If remapping is performed (as in the following tests), the projection on a new mesh must be consistent with the distribution chosen for the fluxes resulting the interpolating function a parabola. Two solutions are considered. The first one is to calculate third-order values of ρ, u, and p at boundaries from neighbour zones. The simplest form is

$$P^{\pm}_i = P_i \pm \frac{\delta_i}{\delta_{i-1} + \delta_i + \delta_{i+1}} \left(\frac{\delta_{i-1}}{\delta_i + \delta_{i+1}} (P_{i+1} - P_i) + \frac{\delta_{i+1}}{\delta_{i-1} + \delta_i} (P_i - P_{i-1}) - \frac{\delta_i}{\delta_i + \delta_{i\pm1}} (P_i - P_{i\pm1}) \right) \tag{15}$$

with upper and lower signs for right and left boundaries. The same formula is applied for the velocity, changing δ_i by $\delta_{i+1/2}$. These interface values must be limited by the average values of the adjacent zones. After that, these values are monotonized, and a parabola is reconstructed with the boundary and mean values. The other approach is the same as that used in the PPM method. Both share the same monotonicity constraint. PPM is more accurate in all the performed tests and is globally monotonic. Because of this better performance, the PPM projection is presently used in every 1D test presented here. Density is interpolated in volume coordinates; pressure and velocity are interpolated in mass coordinates. The mass flux across zone boundaries is determined from the density distribution and the internal energy profile is obtained from the density and pressure ones. We do not apply any interpolation or monotonicity constraint to the total energy. Instead of that, kinetic energy is independently remapped with the square of the velocities without any additional correction. This procedure is commonly used with that in other staggered mesh codes [11]. With this interpolation, the algorithm is globally free-stream invariant and it is in conservation form (i.e., a transformation as $u \mapsto u + a$, $x \mapsto x + at$, with left-invariant ρ and e). In fact, the definitions in Eqs. (13) and (14), and the remapping of the kinetic energy are chosen to verify this property. Finally the contact discontinuity detection algorithm is not applied whenever shock waves are detected, i.e., $f_i > 0$.

5. RESULTS

Tests on RMFCT have been performed on one-dimensional cartesian problems. In the three cases analysed, the time step is fixed to 0.4 times the Courant limit and the diffusion coefficients and contact discontinuity parameters are also common.

Solutions for the Riemann problem (Sod's test) are shown in Fig. 1 for 100 Eulerian numerical zones at time 0.141 (44 time steps). The exponent of the convergence rate (L^1 error) is reported for higher times (2.820) with wall boundary conditions at $x = 0$ and $x = 1$. It allows several interactions between waves, making the test more significant. The reference solution is the Lagrangian one with a

TABLE I

Initial Conditions for the Each Test

Test	Initial densities	Internal energy	Problem time	Convergence rate
Sod	1.0 0.125	2.5 2.0	0.141	1.05
Blast wave	1.0 1.0 1.0	2500 0.025 250	0.038	0.98
Explosion	1.29 1.29 1.29	0.01 $2.7349/\delta \times 10^6$ 0.01	7.1718×10^{-5}	0.97

Note. All conditions are performed with constant mesh space δ, time step of 0.4 times the Courant limit, wall boundary conditions, and perfect gas EOS with $\gamma = 1.4$. The exact solution to measure the convergence rate (L^1 error) is the Lagrangian one with a high number of zones (≥ 1200).

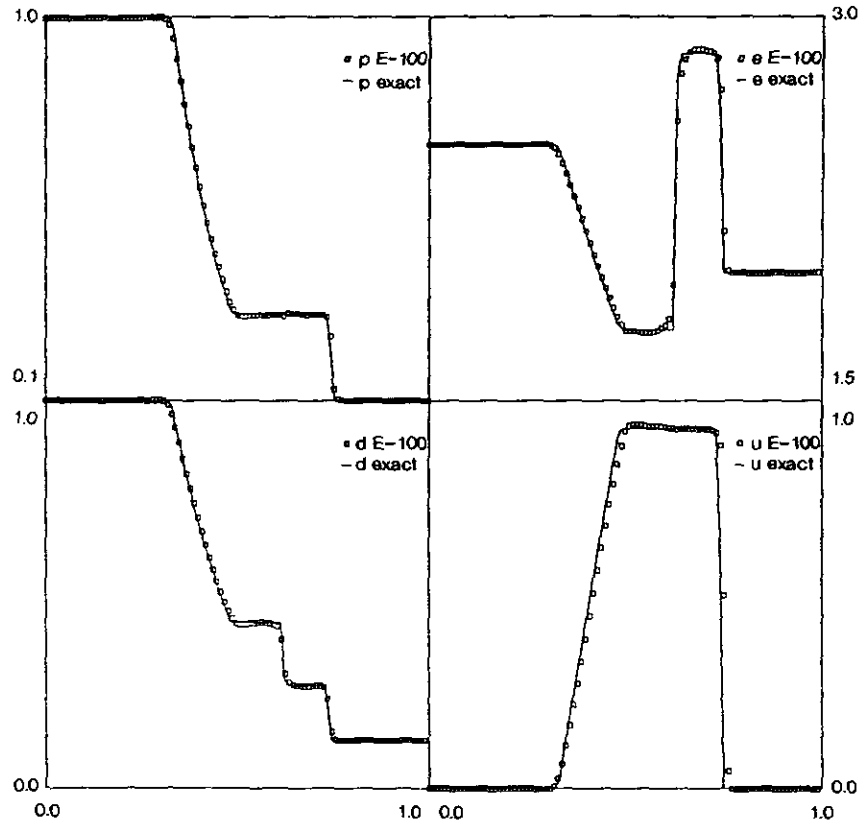

FIG. 1. Results on the Sod's test with 100 Eulerian zones. Exact zone averaged solution is plotted in continuous line.

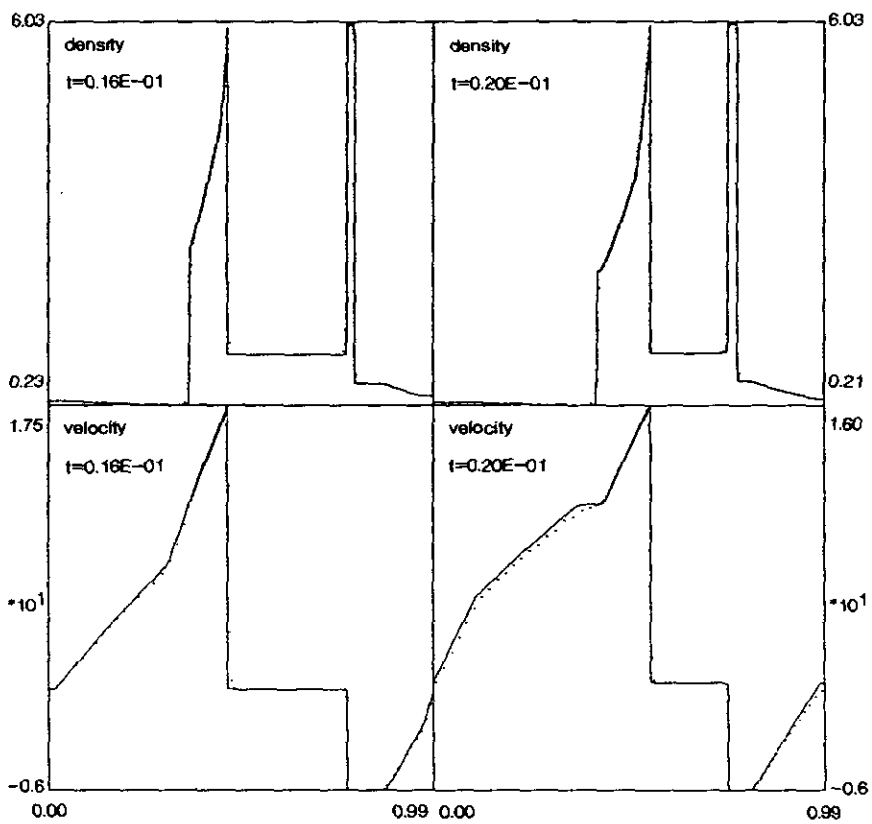

FIG. 2. Density and velocity profiles for 200 (points) and 1200 (line) Lagrangian zones, at times 0.016 and 0.02, for the blast wave test [12].

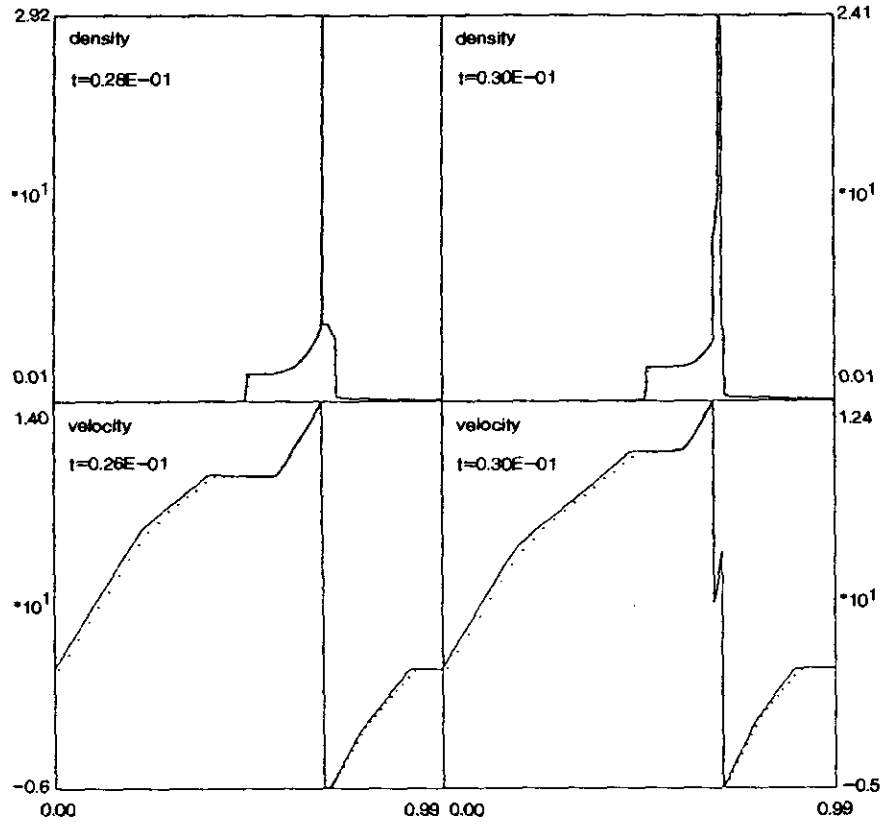

FIG. 3. Density and velocity profiles for 200 (points) and 1200 (line) Lagrangian zones, at times 0.028 and 0.30, for the blast wave test [12].

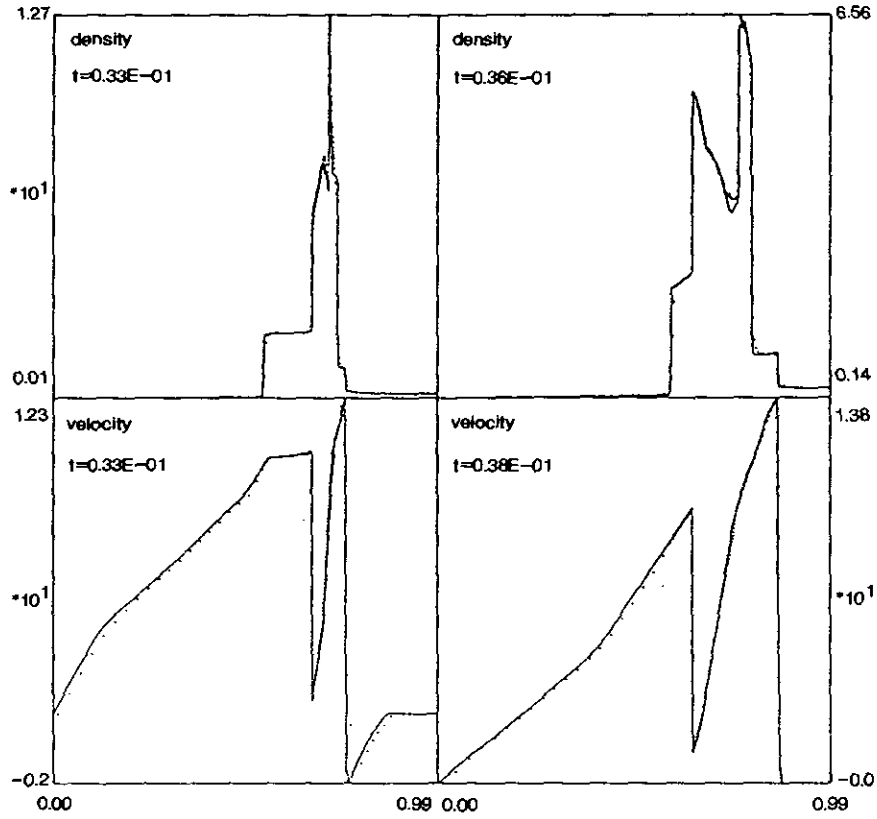

FIG. 4. Density and velocity profiles for 200 (points) and 1200 (line) Lagrangian zones, at times 0.033 and 0.038, for the blast wave test [12].

high number of zones (1800), which is remapped at the final time over the actual mesh.

The blast wave problem [12] is initially two independent Riemann problems (see Table I), with two shock waves traveling toward the center. These two shocks collide at $t \approx 0.028$ and a new contact discontinuity is produced with two emerging shocks. In Lagrangian calculations numerical problems could arise in the region around this contact discontinuity. RMFCT gives accurate results (see Figs. 2–4) in this region and we take as an exact solution the Lagrangian results with 1200 zones. This result is projected (via PPM interpolation) at time 0.038 on the Eulerian mesh. With these values we measure the error (Fig. 5) of the Eulerian + Lagrangian (remapping at each time step) solutions.

The error depends strongly on the contact discontinuity detector built in the PPM projection, but the convergence rate is hardly affected. The highest loss in accuracy is observed in the two leftmost contact discontinuities (with the higher jump in density) and in the region between the first contact discontinuity and the shock wave. If the internal energy is directly remapped with the same coefficients for the contact discontinuity detector as in the PPM method, the error on the density decreases dramatically for RMFCT. However, in that case, the velocity and pressure around the contact discontinuity on the left side oscillates on three zones during the whole simulation.

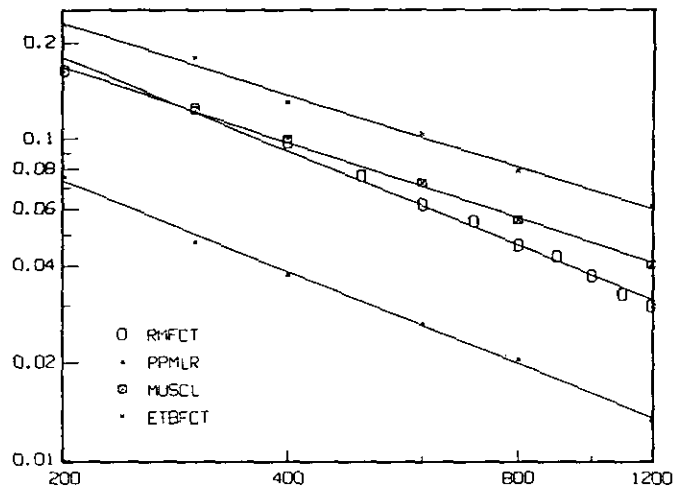

FIG. 5. Errors of several algorithms on the (Eulerian) blast wave problem. Errors of ETBFCT, MUSCL, and PPMLR are taken from [12]. A linear adjustment is chosen for each algorithm. The error ε is measured as $\varepsilon = (1/N) \sum |\rho_i - \rho_i^*|$ with ρ^* the Lagrangian solution with 1200 zones.

For the present remapping step the solution of this problem for the velocity profile is erroneous for the expansion fan at the left. This error (4%) is produced at early times on the left side of the first contact discontinuity and is not effectively damped. This can be observed at the leftmost zones (Fig. 8) at $t = 0.038$ and does not appear in the pure

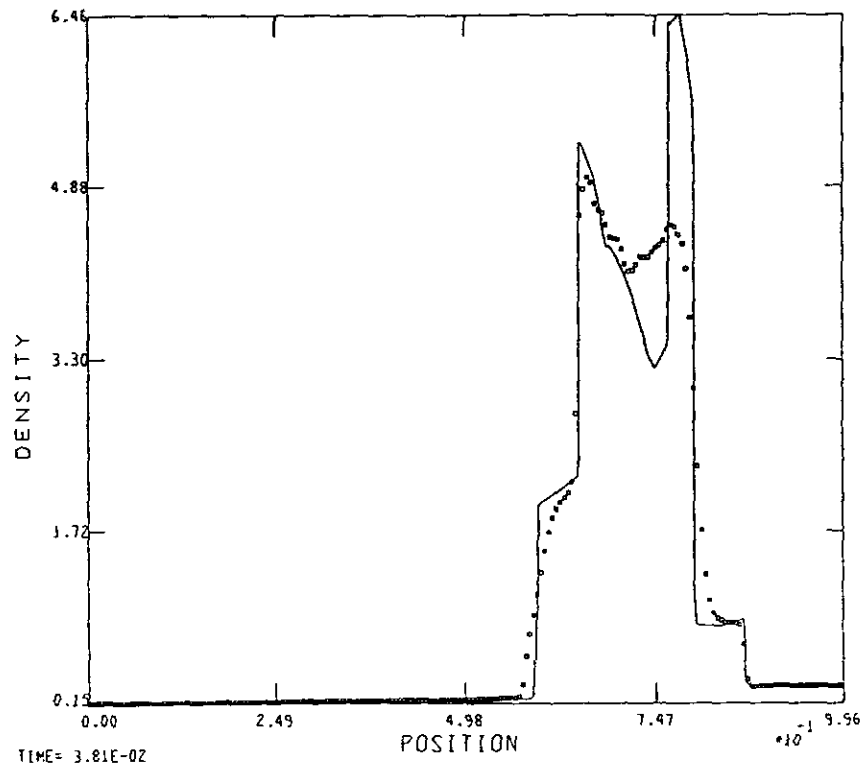

FIG. 6. Density distribution of the blast wave test with 200 Eulerian zones. The 1200 Lagrangian solution, taken as exact, is plotted for comparison.

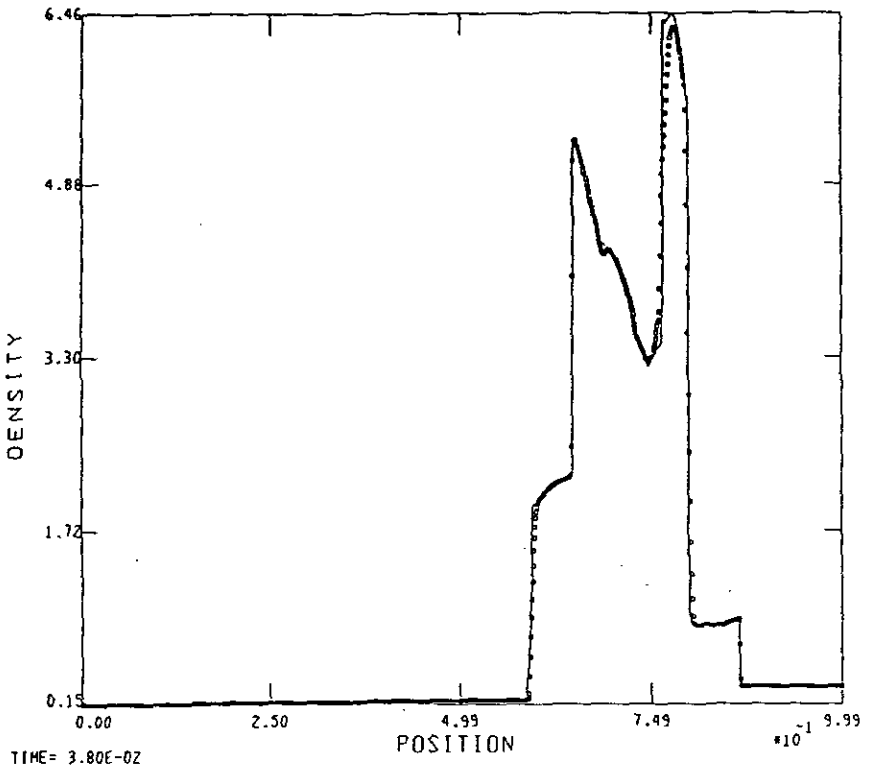

FIG. 7. Density distribution of the blast wave test with 1200 Eulerian zones. The 1200 Lagrangian solution is plotted for comparison.

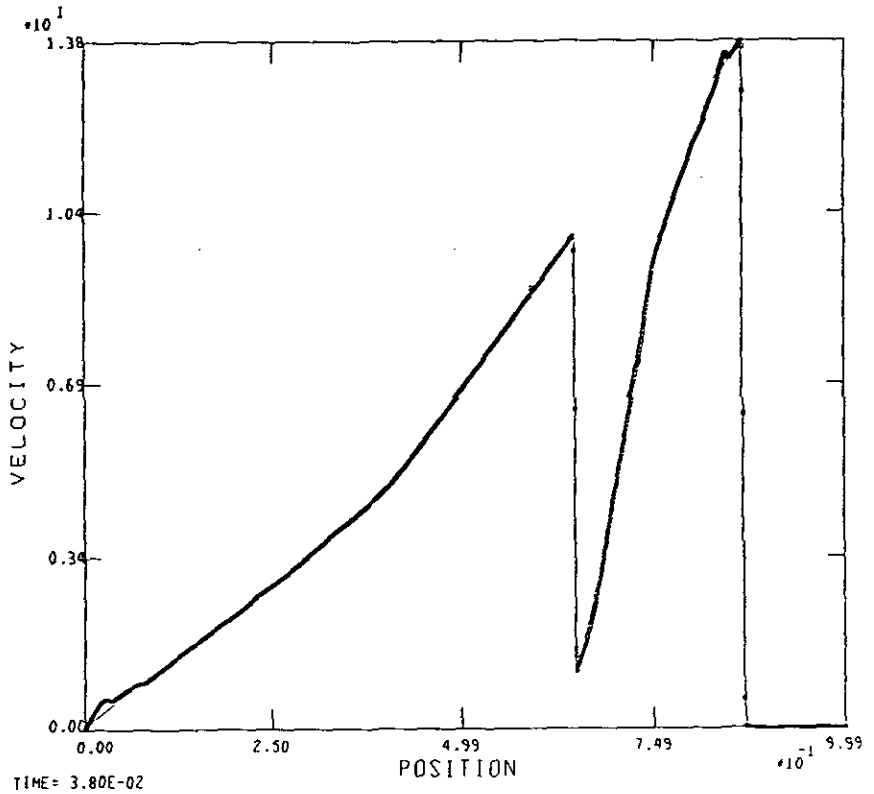

FIG. 8. Velocity distribution of the blast wave test with 1200 Eulerian zones. A large error can be observed near to $x = 0$, which is reminiscent of an overshoot in the initial times.

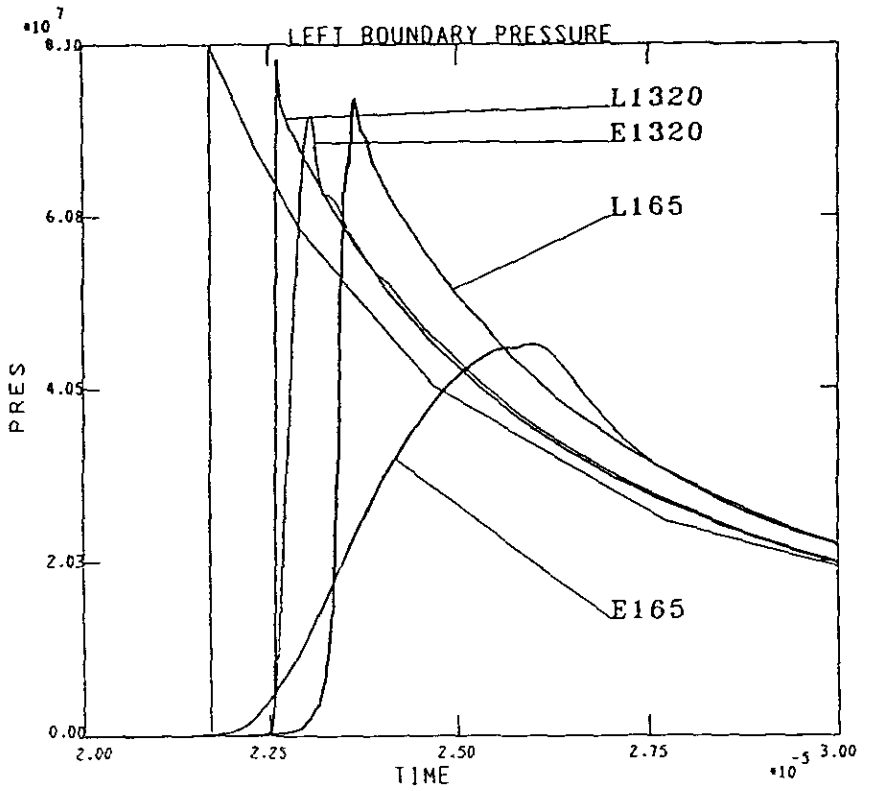

FIG. 9. Evolution of left boundary pressure for the explosion test [13]. Lagrangian (L) and Eulerian (E) results are plotted for 1320 and 165 zones. The leftmost curve is the exact solution. Shock wave velocity is a 4% error even in the more accurate (Lagrangian) solution.

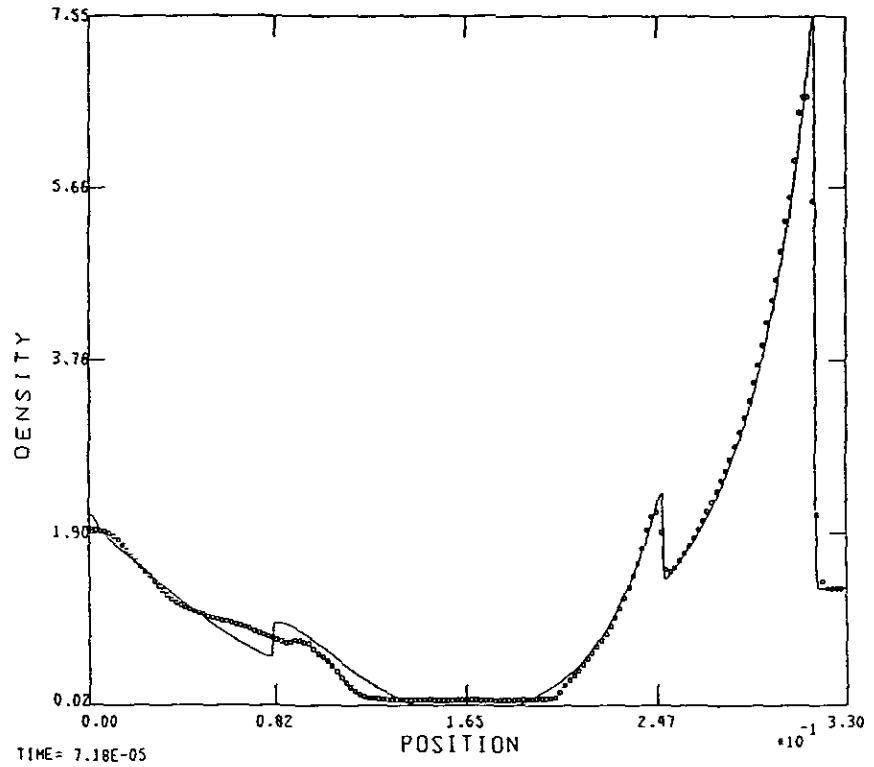

FIG. 10. Comparison between two Eulerian solutions of the density profile with 1320 and 165 zones at time 7.18×10^{-5} for the explosion [13] test.

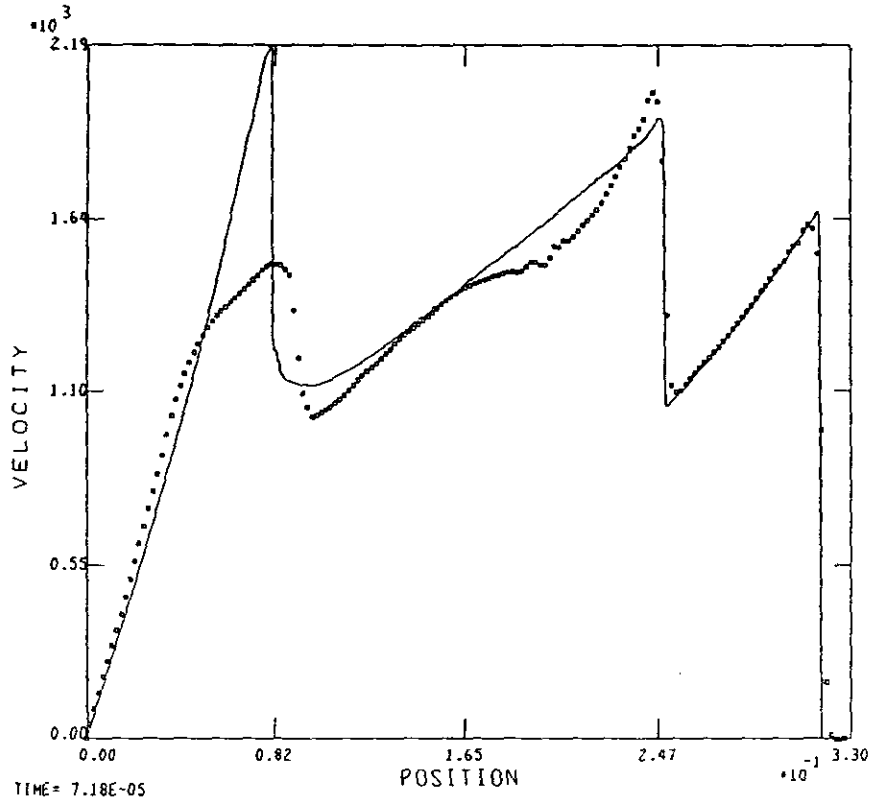

FIG. 11. Comparison between two Eulerian solutions of the velocity profile with 1320 and 165 zones at time 7.18×10^{-5} for the explosion [13] test.

Lagrangian case. With Eulerian mesh, the density profile is slightly affected (see Figs. 6 to 8) by this error, so the convergence rate remains high.

In the explosion test [13] a finite amount of energy is released in the point $x = 0.1$. The initial internal energy of the gas is set to 0.01, instead of zero as in the original problem. At early times, two shocks of infinite strength travel outward the explosion point. At $t \approx 2.17 \times 10^{-5}$ the leftward traveling shock collides against the wall located at $x = 0$. The reflected shock reaches the explosion point at $t \approx 4.17$, and two new shocks emerge. We compare the results of boundary pressure (Fig. 9), shock velocity, and the density and velocity profiles (Figs. 10, 11) at time 7.1718×10^{-5} with the exact and the fine-mesh solutions.

The high temperature region (infinite in the exact solution) is spread over a large distance even in the case of 1330 Eulerian zones. The density and velocity profiles at the final time do not seem to be affected by this spread, but the arrival time for the shock wave striking the left boundary is in 4% of error in the best case. This error is 8% for 165 zones indicating either poor convergence rate for this magnitude or not convergence at all. There is another error associated with the density near the left wall at $x = 0$. This error increases with the spreading of the shock wave [8], and it is common to almost any numerical method. Finally, the minimum density in the explosion point (zero in the exact solution) is, in the best case, around 0.01. RMCFT does not allow zero density points.

Computer time for the remapping step is similar to that of the PPM scheme, but the Lagrangian step is much faster, similar to the original FCT scheme, where most of the calculation time comes from the flux correction step. According to L^1 errors and cpu time, RMCFT is not competitive with the PPM in terms of the time consumed for a fixed error because of the remapping phase. However, for the Lagrangian phase, no Riemann solver is required in this step so that the schem is appropriate for more complicated conservation laws. Because there is no monotonicity constraint in the interpolation, full 2D extensions are easier, and a higher accuracy is obtained at virtually no time overhead. Finally, the implementation of the method in classical Lagrangian codes does not change the centering of the variables, the program flow, or the equation of state normally used.

6. CONCLUSION

We have described an implementation of FCT on a Lagrangian code with staggered mesh. Diffusion coefficients are obtained in order to damp oscillations even in the entropy distribution. An additional diffusion is recognised as necessary near the shocks. As an example, a third-order

algorithm is used in two stringent tests both in Eulerian and Lagrangian coordinates. The overall method verifies conservation and free stream invariance, giving good convergence rate in the blast wave test and good behaviour around shocks. Better projection methods are under study to prevent the errors pointed out in the tests.

APPENDIX A: FULL LAGRANGIAN PHASE

Let be $L(F, G, f)$ the FCT hybridization given by [2]

$$\begin{aligned} L_{k+1/2}(F, G, f) = F_{k+1/2} + G_{k+1/2} - S_{k+1/2} \\ \cdot \max(0, \min(|G_{k+1/2}|, \\ S_{k+1/2}\delta_{k+1}(\hat{f}_{k+2} - \hat{f}_{k+1}), \\ S_{k+1/2}\delta_{k-1}(\hat{f}_k - \hat{f}_{k-1})) \end{aligned}$$

with

$$S_{k+1/2} = \operatorname{sign}(G_{k+1/2})$$
$$\hat{f}_k = f_k + (F_{k+1/2} + G_{k+1/2} - F_{k-1/2} - G_{k-1/2})/\delta_k.$$

With $\bar{u}_{i+1/2}$ as defined in (11), $\bar{P}_i$ as defined in (12), and $F_i^{(u)}$ as defined in (3), the first step is

$$F_i^h = \frac{\delta t}{2}(-\bar{P}_i + g_i^{(u)}(u_{i+1/2} - u_{i-1/2})) \tag{16}$$

$$F_i^c = L_i(F^h, F^{(u)}, u) \tag{17}$$

$$u_{i+1/2}^{1/2} = u_{i+1/2} + (F_{i+1}^c - F_i^c)/\delta_{i+1/2} \tag{18}$$

$$P_i^* = -\frac{2}{\delta t} F_i^c. \tag{19}$$

With $\bar{u}_{i+1/2}$ as defined in (11) and $F_i^{(V)}$ as defined in (2) the next step is

$$F_{i+1/2}^h = \frac{\delta t}{2}(\bar{u}_{i+1/2}^{1/2} + g_i^{(V)}(V_{i+1} - V_i)) \tag{20}$$

$$F_{i+1/2}^c = L_{i+1/2}(F^h, F^{(V)}, V) \tag{21}$$

$$V_i^{1/2} = V_i + (F_{i+1/2}^c - F_{i-1/2}^c)/\delta_i \tag{22}$$

$$u_{i+1/2}^* = \frac{2}{\delta t} F_{i+1/2}^c. \tag{23}$$

With $P_{i+1/2}$ as defined (13), K_i in (14) and $F_{i+1/2}^{(S)}$ in (7), the internal energy is obtained from

$$F_{i+1/2}^h = \frac{\delta t}{2}(-u_{i+1/2}^* P_{i+1/2}^* + g_i^{(V)}(e_{i+1} - e_i)) \tag{24}$$

$$e_i^* = e_i - (K_i^{1/2} - K_i) + (F_{i+1/2}^h - F_{i-1/2}^h)/\delta_i \tag{25}$$

$$F_{i+1/2}^c = L_{i+1/2}(0, F^{(S)}, e^*) \tag{26}$$

$$e_i^{1/2} = e_i^* + (F_{i+1/2}^c - F_{i-1/2}^c)/\delta_i \tag{27}$$

$$P_i^{1/2} = P(V_i^{1/2}, e_i^{1/2}). \tag{28}$$

In the last step we calculate $u_{i+1/2}^1$, changing $\delta t \mapsto 2\delta t$ in Eqs. (16)–(19), and recalculate $u_{i+1/2}^{1/2} = \frac{1}{2}(u_{i+1/2}^1 + u_{i+1/2})$ and then V_i^1 and e_i^1 from (19)–(28).

ACKNOWLEDGMENTS

This work was partially supported by UNESA-OCIDE. The author thanks J. M. Aragonés, J. J. Honrubia, and J. M. Perlado for helpful discussions, and N. Soriano for reading the manuscript

REFERENCES

1. J. von Neuman and R. D. Richtmyer, *J. Appl. Phys.* **21**, 232 (1949).
2. S. T. Zalesak, *J. Comput. Phys.* **31**, 335 (1979).
3. J. P. Boris, Report NRL-3237, NRL, March 1976 (unpublished).
4. Ph. Colella and P. R. Woodward, *J. Comput. Phys.* **54**, 174 (1984).
5. P. L. Roe, in *Numerical Methods for the Euler Equations of Fluid Dynamics*, pp. 14–32, (SIAM, Philadelphia, 1983).
6. P. L. Roe, *J. Comput. Phys.* **43**, 357 (1981).
7. A. Jameson, "Transonic Flow Calculations for Aircraft," in *Numerical Methods in Fluid Dynamics*, edited by Franco Brezzi, (Springer-Verlag, New York/Berlin, 1983), p. 156.
8. W. F. Noh, *J. Comput. Phys.* **72**, 78 (1978).
9. P. M. Velarde, Preprint DENIM 213, Instituto de Fusión Nuclear, Madrid, 1989.
10. A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, *J. Comput. Phys.* **71** (2), 231 (1987).
11. S. L. Thompson, Technical Report SAND77-1339, Sandia National Laboratories, Albuquerque, NM, 1977.
12. P. Woodward and P. Colella, *J. Comput. Phys.* **54**, 115 (1984).
13. D. Huang, "A Test Problem for Unsteady Shock Wave Calculation," in *Tenth International Conference on Numerical Methods in Fluid Dynamics*, edited by F. G. Zhuang and Y. L. Zhu, editors, (Springer-Verlag, New York/Berlin, 1986), p. 324.