JOURNAL OF COMPUTATIONAL PHYSICS 73, 203–230 (1987)

A Comparison of Flux Limited Difference Methods and Characteristic Galerkin Methods for Shock Modelling*,†

K. W. MORTON

Computing Laboratory, Oxford University, 8–11 Keble Road, Oxford OX1 3QD, England

AND

P. K. SWEBY

Department of Mathematics, Reading University, Whiteknights, Reading RG6 2AX, England

Received August 15, 1985; revised May 23, 1986

A comparison is made between flux-limited finite difference methods and characteristic Galerkin methods for approximating hyperbolic conservation laws. At the first-order level, the characteristic Galerkin scheme using piecewise constants is closely related to the difference schemes of Engquist, Osher, and Roe. Adaptive recovery techniques used to improve accuracy then have much in common with the flux limiters used with difference methods. These relationships are explored and comparisons made using the linear advection, inviscid Burgers and Euler equations. A new, simple formulation is given of the characteristic Galerkin method using piecewise constant elements with piecewise linear recovery: it reduces to the Engquist–Osher algorithm but with a modified flux function when the CFL number is no greater than one half.

1. INTRODUCTION

In recent years many authors have devised high resolution, total variation diminishing (TVD) finite difference schemes in order to obtain sharper profiles to represent discontinuities than is possible with first-order schemes, whilst avoiding the spurious oscillations which plague the more classical second-order schemes. One important class of techniques uses flux limiters [4, 22, 23, 26, 27, 29] which, as with FCT (flux-corrected transport) methods [2, 14, 31], utilise a limited amount of anti-diffusive flux to add to a first-order scheme.

* The work reported here forms part of the research programme of the Oxford/Reading Institute for Computational Fluid Dynamics and has been supported by the Science and Engineering Research Council Grant GR/D/39512.

† This paper is dedicated to the memory of Keith Roberts from whom the first author received untold inspiration and learned the value of combining physical principles with mathematical ingenuity.

0021-9991/87 $3.00

More recently and in parallel with this work, finite element methods based on the characteristic Galerkin formulation have started to be developed for shock modelling [15, 16, 17]. As shown in [15], piecewise constant elements lead to a first-order scheme equivalent to the Engquist–Osher scheme [5] but data recovery techniques can be used to obtain higher accuracy.

In this paper we explore this idea using recovery techniques based on piecewise linear functions. We show that the recovery can be applied adaptively to ensure that monotonicity is preserved and thus has a similar role to that of flux limiters. Moreover, we show that one form of the update procedures has much in common with the difference schemes of Roe [22, 23], while a new simpler formulation again reduces to the Engquist–Osher algorithm for CFL numbers up to one half, but now with a locally modified flux function.

The two approaches are described in their scalar forms in Sections 2 and 3. Then in Section 4 we present their extensions to systems of equations by decomposing them into characteristic fields. Comparisons of numerical results for one-dimensional model problems are made in Section 5. The results are seen to be very comparable: sometimes one approach has the edge, sometimes the other. Thus we conclude with a brief discussion of their relative merits, particularly in regard to the prospects for multi-dimensional problems.

2. Flux Limited Difference Schemes

Consider the scalar conservation law in one dimension with convex flux $f(u)$

$$\partial_t u + \partial_x f(u) = 0 \qquad t > 0,\ x \in \mathbb{R}$$

and with initial data

$$u(x, 0) = u^0(x) \tag{2.1}$$

given. Weak solutions $v(x, t)$ of this equation have the property

$$\frac{d}{dt} \int |\partial_x v|\, dx \leqslant 0, \tag{2.2}$$

i.e., their total variation is non-increasing.

We approximate the conservation law (2.1) by an explicit conservative finite difference scheme on a uniform mesh $(\Delta x, \Delta t)$,

$$u_k^{n+1} = u_k^n - \lambda(h_{k+1/2}^n - h_{k-1/2}^n), \tag{2.3}$$

where λ is the mesh ratio

$$\lambda = \Delta t/\Delta x \tag{2.4}$$

and $h^n_{k+1/2} \equiv h(u^n_{k-l}, ..., u^n_{k+m})$ is a consistent numerical flux function, such that

$$h(u, ..., u) = f(u). \tag{2.5}$$

The property of the discrete solution corresponding to the total variation property of the exact solution is

$$\sum_k |u^{n+1}_{k+1} - u^{n+1}_k| \leqslant \sum_k |u^n_{k+1} - u^n_k| \tag{2.6}$$

and has been dubbed "total variation diminishing" (TVD) by Harten [8]. Unlike solutions of the conservation law, solutions of the difference scheme (2.3) do not necessarily possess the TVD property (2.6), and hence produce spurious oscillations, particularly near any discontinuities of the solution.

In order to eliminate such spurious features of the numerical solution, as well as to obtain desirable convergence properties, many authors now seek schemes whose solutions do satisfy this criterion (2.6). Harten [8] and others (e.g., [27, 12, 24, 10]) have shown that a difference scheme written in the form

$$u^{n+1}_k = u^n_k - C^n_{k-1/2}\,\delta u^n_{k-1/2} + D^n_{k+1/2}\,\delta u^n_{k+1/2}, \tag{2.7}$$

where $C^n_{k+1/2}$, $D^n_{k+1/2}$ may depend on the set $\{u^n_k\}$ and $\delta v_{k+1/2} := v_{k+1} - v_k$, produces TVD solutions iff

$$0 \leqslant C^n_{k+1/2}, \qquad 0 \leqslant D^n_{k+1/2}, \qquad C^n_{k+1/2} + D^n_{k+1/2} \leqslant 1. \tag{2.8}$$

Note that the final inequality imposes a CFL-like condition on the scheme.

In general first-order accurate schemes are TVD but give poor resolution whilst, although giving higher resolution, the classical second-order schemes, such as the Lax–Wendroff, are not TVD. It is easily shown that constant coefficient schemes of second or higher order accuracy cannot be TVD and hence there has been much work carried out on adaptive schemes which, by using solution dependent coefficients, are able to combine high resolution with the TVD property (see, e.g., Boris and Book [2], Harten [8], Roe [23], van Leer [29], to name just a few).

One of the pioneer methods is the flux corrected transport (FCT) method of Boris and Book [2], and more recently of Zalesak [31] and McDonald and Ambrosiano [14]. The technique used is to supplement the numerical flux of a low order scheme with the difference in the fluxes of that scheme and a higher order scheme but corrected in such a way as to ensure the TVD property.

We consider here a special subset of FCT, that of flux limiters (Sweby [26]), where a limited anti-diffusive flux is added to a first-order TVD scheme to obtain a higher resolution TVD scheme. (Note that the "limiting" performed here can in fact be "enhancement" of the anti-diffusive flux if this does not violate the TVD constraints.) It is desirable for this first-order TVD scheme to be entropy satisfying as well, thus avoiding non-physical shocks; such a class of schemes is provided by the

E-schemes of Osher [20], which have been generalised to the fully discrete case by Tadmor [28]. This class of schemes is characterised by their numerical flux satisfying

$$(u^n_{k+1}-u^n_k)(h^n_{k+1/2}-f(u))\leqslant 0 \qquad \forall u \text{ between } u^n_k,\ u^n_{k+1}. \tag{2.9}$$

Now we define left- and right-moving flux differences

$$(\Delta f^n_{k+1/2})^- := [h^n_{k+1/2}-f^n_k] \qquad (\Delta f^n_{k+1/2})^+ := -[f^n_{k+1}-h^n_{k+1/2}], \tag{2.10}$$

where $f^n_k := f(u^n_k)$, from which the corresponding CFL numbers

$$v^{\pm}_{k+1/2} := \lambda(\Delta f^n_{k+1/2})^{\pm}/\delta u^n_{k+1/2} \tag{2.11}$$

are obtained: each takes the sign of its superscript. Equating $v^+_{k+1/2}$ with $C^n_{k+1/2}$ and $-v^-_{k+1/2}$ with $D^n_{k+1/2}$ in (2.7), it is easily seen that the scheme (2.3) is TVD if it is an E-scheme, so long as it satisfies the CFL condition

$$v^+_{k+1/2}-v^-_{k+1/2}\leqslant 1. \tag{2.12}$$

In this formulation it should be noted that $C^n_{k+1/2}$ and $D^n_{k+1/2}$ include terms corresponding to right and left numerical viscosities for the cell (x_k, x_{k+1}), and second-order TVD is achieved by effectively modifying them within the bounds of the inequalities (2.8). When using flux limiters to obtain high resolution schemes, $C^n_{k+1/2}$ and $D^n_{k+1/2}$ are modified by adding a limited anti-diffusive flux to the numerical flux function of a first-order scheme of the form (2.3), viz.,

$$\begin{aligned} u^{n+1}_k = u^n_k - \lambda\Delta_-\{&h^n_{k+1/2}+\tfrac{1}{2}\phi(r^+_k)(1-v^+_{k+1/2})(\Delta f^n_{k+1/2})^+ \\ &-\tfrac{1}{2}\phi(r^-_{k+1})(1+v^-_{k+1/2})(\Delta f^n_{k+1/2})^-\}, \end{aligned} \tag{2.13}$$

where $\Delta_- v_k := v_k - v_{k-1} =: \Delta_+ v_{k-1}$ and $\phi(r)$ is the limiter: this is a function of the ratios

$$r^+_k := \frac{\tfrac{1}{2}(1-v^+_{k-1/2})(\Delta f^n_{k-1/2})^+}{\tfrac{1}{2}(1-v^+_{k+1/2})(\Delta f^n_{k+1/2})^+}, \qquad r^-_k := \frac{\tfrac{1}{2}(1+v^-_{k+1/2})(\Delta f^n_{k+1/2})^-}{\tfrac{1}{2}(1+v^-_{k-1/2})(\Delta f^n_{k-1/2})^-}. \tag{2.14}$$

After a little manipulation (2.13) can be put in the form (2.7) by setting

$$\begin{aligned} C^n_{k+1/2} &= v^+_{k+1/2}\{1+\tfrac{1}{2}(1-v^+_{k+1/2})[\phi(r^+_{k+1})/r^+_{k+1}-\phi(r^+_k)]\} \\ D^n_{k+1/2} &= -v^-_{k+1/2}\{(1+\tfrac{1}{2}(1+v^-_{k+1/2})[\phi(r^-_k)/r^-_k-\phi(r^-_{k+1})]\}. \end{aligned} \tag{2.15}$$

Conditions that $\phi(r)$ must satisfy in order that (2.13) be TVD can then be readily established (see Sweby [26]): assuming that we set $\phi(r)\equiv 0$ for $r<0$, we need both $\phi(r)$ and $\phi(r)/r$ to lie in the interval $[0, 2]$. Furthermore, since the Lax–Wendroff method corresponds to $\phi(r)\equiv 1$ and the Warming and Beam [30] second-order upwind scheme to $\phi(r)\equiv r$, a $\phi(r)$ lying between these will lead to a convex average

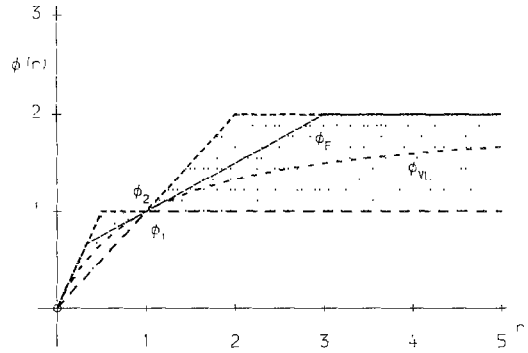

FIG. 1. Feasible regions for flux limiters: examples are $\phi_1 = \text{minmod}$, $\phi_2 = \text{superbee}$, $\phi_{VL} = \text{van Leer}$, and ϕ_F based on Fromm.

of them and hence normally of second-order accuracy (this is a sufficient condition; the necessary condition for second-order accuracy is simply $\phi(1+\varepsilon) = 1 + O(\varepsilon)$). These regions are shown on Fig. 1, which also shows some of the flux limiters in common use—namely the minmod limiter ϕ_1 and the superbee limiter ϕ_2 of Roe [23], van Leer's limiter ϕ_{VL} [29], and a limiter ϕ_F based on Fromm's scheme [6, 1].

We now consider an alternative viewpoint for flux limiter schemes, namely the increment and transfer formulation [23], which will be helpful in the comparison with characteristic Galerkin schemes. Dropping the superscript n, we denote by $C^1_{k+1/2}$ and $D^1_{k+1/2}$ the coefficients of the first-order scheme employed. This scheme can be regarded as an algorithm for each "cell" (x_k, x_{k+1}):

$$\textit{increment the value of } u_k \quad \textit{by } D^1_{k+1/2}\,\delta u_{k+1/2}$$

and (2.16)

$$\textit{decrement the value of } u_{k+1} \textit{ by } C^1_{k+1/2}\,\delta u_{k+1/2}.$$

The flux limiter may then be regarded as introducing a transfer step; that is, for each cell (x_k, x_{k+1}):

$$\textit{transfer } \tfrac{1}{2}\phi(r_k^+)(1 - C^1_{k+1/2})\,C^1_{k+1/2}\,\delta u_{k+1/2} \textit{ from } u_{k+1} \textit{ to } u_k$$

and (2.17)

$$\textit{transfer } \tfrac{1}{2}\phi(r_k^-)(1 - D^1_{k+1/2})\,D^1_{k+1/2}\,\delta u_{k+1/2} \textit{ from } u_k \quad \textit{to } u_{k+1}.$$

Since $C^1_{k+1/2}$ and $D^1_{k+1/2}$ are usually heavily dependent on the sign of the wave speed, $\delta f/\delta u$, the first-order stage amounts to a decision as to which direction to

allocate given increments. For example, the Cole–Murman first-order upwind scheme [19] may be formulated simply as

$$\text{increment} \begin{cases} u_k \\ u_{k+1} \end{cases} \quad \text{by} \quad -\lambda \delta f_{k+1/2} \quad \text{if} \quad \frac{\delta f_{k+1/2}}{\delta u_{k+1/2}} \begin{cases} <0 \\ >0. \end{cases} \tag{2.18}$$

This scheme takes as its highest priority the imposition of a Rankine–Hugoniot-type jump condition on any discontinuity, giving the correct shock speed but unfortunately also allowing entropy-violating jumps. On the other hand, the Engquist–Osher scheme [5] takes entropy satisfaction as its main criterion and, by using overturned manifolds, gives physically correct solutions, but at the cost of poorer shock resolution. This scheme may be formulated (for convex $f(u)$) as

$$\text{use } -\lambda(f_{k+1}-\bar{f}) \text{ to increment} \begin{cases} u_k & \text{if } f'(u_{k+1})<0 \\ u_{k+1} & \text{if } f'(u_{k+1})>0 \end{cases}$$

and ((2.19)

$$\text{use } -\lambda(\bar{f}-f_k) \quad \text{to increment} \begin{cases} u_k & \text{if } f'(u_k)<0 \\ u_{k+1} & \text{if } f'(u_k)>0, \end{cases}$$

where $\bar{f}$ is the sonic value of f, i.e., $f(\bar{u})$ such that $f'(\bar{u})=0$.

At this first-order level, of course, the Godunov scheme [7] based on the exact solution of the Riemann problem gives the best choice: it reduces to the Cole–Murman scheme at physical shocks and the Engquist–Osher at expansion waves. However, in general this requires detailed knowledge of the function f and is much more difficult to extend to systems of equations and to higher order schemes. Thus the objective is to model adequately the true evolution of the approximation from time level n, using only the minimum information on f: (2.18) uses only f_k and f_{k+1}, while (2.19) also uses f'_k, f'_{k+1}, and $\bar{f}$. Roe [23] has proposed a modification of the Cole–Murman scheme which, assuming convex f, interpolates for $\bar{f}$ to yield

$$\begin{aligned} -\lambda(f_{k+1}-\bar{f}) &\approx -\lambda \frac{v_{k+1/2}-\hat{v}^L_{k+1/2}}{\hat{v}^R_{k+1/2}-\hat{v}^L_{k+1/2}} \hat{v}^R_{k+1/2}\,\delta u_{k+1/2} \\ -\lambda(\bar{f}-f_k) &\approx -\lambda \frac{\hat{v}^R_{k+1/2}-v_{k+1/2}}{\hat{v}^R_{k+1/2}-\hat{v}^L_{k+1/2}} \hat{v}^L_{k+1/2}\,\delta u_{k+1/2}, \end{aligned} \tag{2.20}$$

where

$$\hat{v}^L_{k+1/2} = \text{Min}(v_k, v_{k+1/2}), \qquad \hat{v}^R_{k+1/2} = \text{Max}(v_{k+1}, v_{k+1/2}), \tag{2.21}$$

$v_{k+1/2}$ is the usual cell CFL number and v_k, v_{k+1} are (approximations to) $\lambda f'(u_k)$, $\lambda f'(u_{k+1})$. Then these increments are distributed as in (2.19). Note that if instead of (2.21) we take $\hat{v}^L_{k+1/2}=v_k$ and $\hat{v}^R_{k+1/2}=v_{k+1}$ then shocks are treated via overturned

manifolds as in the Engquist–Osher scheme. Both of these schemes are E-schemes and therefore entropy satisfying.

In the finite element methods of the next section, not only is the assumed form of the approximation at each time level made explicit as is usual with finite element methods, but also the way in which its evolution is modelled.

3. Euler Characteristic Galerkin Schemes

3.1. *Basic Formulae*

Suppose at time level n we approximate the solution of the conservation law (2.1) by an expansion in basis functions $\{\phi_k(x)\}$,

$$U^n(x) = \sum_k U_k^n \phi_k(x). \tag{3.1}$$

The basic characteristic Galerkin method using Euler timestepping (hence ECG method) can be written (see Morton [15, 16, 17]) in the following form, where $\langle \cdot, \cdot \rangle$ denotes the usual L^2 inner product over the space variable x,

$$\langle U^{n+1} - U^n, \phi_k \rangle + \Delta t \langle \partial_x f(U^n), \Phi_k^n \rangle = 0. \tag{3.2a}$$

Here the special test function in the second inner product is given by

$$\Phi_k^n(x) := \frac{1}{a(U^n)\,\Delta t} \int_x^{x + a(U^n)\,\Delta t} \phi_k(z)\, dz \tag{3.2b}$$

and $a(u) = \partial f / \partial u$ is the characteristic speed; that is, Φ_k^n is the average of the basis function ϕ_k over the distance a characteristic travels in one timestep. Such a scheme is clearly conservative, so long as the $\{\phi_k\}$ spans the unit constant. For linear problems it is unconditionally stable: however, it is generally much simpler in form if used for a limited range of CFL numbers.

The derivation and validity of the formula (3.2) is best approached through the "transport collapse" operator used by Brenier [3]. Suppose at time-level n we have an approximation $v(x)$ which, although it may be discontinuous as a function of x, we assume has a continuous graph $[v, x]$ in the (x, u)-plane. Then we introduce the evolutionary operator $\hat{E} = \hat{E}(\Delta t)$ given by the mapping

$$y = x + a(v(x))\,\Delta t \tag{3.3a}$$

$$(\hat{E}v)(y) = v(x). \tag{3.3b}$$

This yields another continuous graph $[\hat{E}v, y]$ which, however, may correspond to $\hat{E}v$ being a multivalued function of y if the characteristics have overtaken one another. In terms of this operator we can write the basic ECG scheme in the form

$$\langle U^{n+1}, \phi_k \rangle = \langle \hat{E}U^n, \phi_k \rangle$$

$$= \int U^n(x)\, \phi_k(y)\, dy,\ y = x + a(U^n)\, \Delta t \tag{3.4}$$

where the integral is to be calculated along the graph $[\hat{E}U^n, y]$ taking account of the sign changes in dy/dx if the characteristics overtake one another. The two forms (3.2) and (3.4) are complementary and both are useful in obtaining the explicit algorithms given below. The form (3.2) shows the role of the flux function more clearly and facilitates comparisons with the difference schemes of the previous section: its identification with (3.4) is achieved by substitution of (3.2b) into (3.2a) and integrating by parts. In all such integrations and in the interpretation of (3.2), the integrals should be interpreted as being along the graphs in the (x, u)-plane.

For the present comparison we will take only piecewise constant basis functions, that is, on a general mesh $\phi_k(x) \equiv 1$ for $x_{k-1/2} < x < x_{k+1/2}$; we call this the element k, consistent with the usual finite element terminology and distinct from the term cell used in Section 2 to denote the interval (x_k, x_{k+1}), where $x_k = \frac{1}{2}(x_{k-1/2} + x_{k+1/2})$ and $\Delta x_k = x_{k+1/2} - x_{k-1/2}$. Then (3.2a) can be written

$$\Delta x_k (U_k^{n+1} - U_k^n) + \Delta t \langle \partial_x f(U^n), \Phi_k^n \rangle = 0. \tag{3.5}$$

For CFL numbers less than unity this reduces to the first-order Engquist–Osher difference scheme (2.19): each flux difference is broken up to give increments

$$-\Delta t[f_{k+1}^n - f(\bar{u})] \qquad \text{and} \qquad -\Delta t[f(\bar{u}) - f_k^n], \tag{3.6}$$

where $\bar{u}$ corresponds to the sonic point (assumed unique, as in the convex f case) at which $a(\bar{u}) = 0$; and the first increment is added to the $\Delta x_k U_k^n$ or $\Delta x_{k+1} U_{k+1}^n$ according to whether $a_{k+1}^n := a(U_{k+1}^n)$ is negative or positive, with the second distributed according to the sign of a_k^n.

The nodal parameters $\{U_k^n\}$ are interpreted as averages over the kth element so that $U^n(x)$ is regarded as representing as closely as possible the projection of the true solution $u^n(x)$ on to the space of piecewise constant functions. Thus (3.2) corresponds to taking this projection $U^n(x)$, tracing its evolution through one timestep by following the characteristics (including the possibility of the solution curve "overturning"), before projecting again to get $U^{n+1}(x)$. Assuming that the evolution stage is carried out sufficiently accurately (it can be done exactly here because the characteristics are straight lines) and in the absence of shocks, the main opportunity for obtaining greater accuracy without going to higher order basis functions lies in deducing more about the true solution $u^n(x)$ from its projection

$U^n(x)$ before the evolution stage. Thus suppose we combine a number of neighbouring values U_k^n to recover a function $\tilde{u}^n(x)$ whose projection is $U^n(x)$:

$$\langle U^n - \tilde{u}^n, \phi_k \rangle = 0 \qquad \forall k. \tag{3.7}$$

Then we can replace (3.3) by

$$\Delta x_k (U_k^{n+1} - U_k^n) + \Delta t \langle \partial_x f(\tilde{u}^n), \tilde{\Phi}_k^n \rangle = 0, \tag{3.8}$$

where $\tilde{\Phi}_k^n$ is given by (3.2b) with U^n replaced by $\tilde{u}^n$. For example, in [16] it was shown that for linear advection the effect of using higher order basis functions can be reproduced in this way: thus if $\tilde{u}^n$ is a quadratic spline (3.8) yields the highly accurate (3*rd*-order) scheme obtained directly from (3.2) using continuous linear basis functions $\phi_k(x)$.

Use of such smooth recovery functions is inappropriate here. On the other hand, in [15, 17] shock recovery algorithms have been given in which each element k is scanned for the presence of a shock, involving a jump from U_{k-1}^n to U_{k+1}^n; this is then moved with the correct shock speed $\Delta_0 f_k^n / \Delta_0 U_k^n$, where $\Delta_0 := \frac{1}{2}(\Delta_+ + \Delta_-)$. The result of this procedure can also be expressed in terms of an integral along a graph, as in (3.4); it can also be combined with some form of smooth recovery between the shocks and we therefore denote this by $[E^S \tilde{u}^n, y]$. From the viewpoint of implementation, however, we have found it most useful to work from (3.2). Suppose we have a single shock and let us denote by $\langle \cdot, \cdot \rangle_L$ and $\langle \cdot, \cdot \rangle_R$ the integrals up to and beyond the shock, which joins recovered states $\tilde{u}_L^n$ and $\tilde{u}_R^n$. Then with the shock speed

$$a_S := \frac{f(\tilde{u}_R^n) - f(\tilde{u}_L^n)}{\tilde{u}_R^n - \tilde{u}_L^n}, \tag{3.9a}$$

we can define

$$\tilde{\Phi}_i^n(x_S) := \frac{1}{a_S \Delta t} \int_{x_S}^{x_S + a_S \Delta t} \phi_i(s)\, ds \tag{3.9b}$$

and hence obtain as an extension of (3.2),

$$\langle U^{n+1} - U^n, \phi_i \rangle + \Delta t \{ \langle \partial_x f(\tilde{u}^n), \tilde{\Phi}_i^n \rangle_L + \langle \partial_x f(\tilde{u}^n), \tilde{\Phi}_i^n \rangle_R + [f(\tilde{u}_R^n) - f(\tilde{u}_L^n)]\, \tilde{\Phi}_i^n(x_S) \} = 0. \tag{3.10}$$

This is the form we shall use in this paper. The shock recovery will be combined with an adaptive recovery procedure using piecewise linear functions. The method and its results will be compared with the flux limiters described in the previous section. For this comparison we shall confine ourselves to a uniform mesh henceforth.

3.2. *Piecewise Linear Recovery*

Leaving aside the shock recovery for the moment, the piecewise linear recovery is based on spreading the discontinuity at $(k+\frac{1}{2})\,\Delta x$ over $\frac{1}{2}\theta_{k+1/2}\,\Delta x$ either side with $0 \leqslant \theta_{k+1/2} \leqslant 1$. Then dropping the superscript n, (3.7) implies that the intermediate recovered levels $\tilde{u}_k$ must satisfy

$$\tfrac{1}{8}\theta_{k-1/2}\tilde{u}_{k-1} + (1 - \tfrac{1}{8}\theta_{k-1/2} - \tfrac{1}{8}\theta_{k+1/2})\,\tilde{u}_k + \tfrac{1}{8}\theta_{k+1/2}\tilde{u}_{k+1} = U_k. \tag{3.11}$$

(On a non-uniform mesh $\theta_{k+1/2} = 1$ corresponds to a linear section from x_k to x_{k+1}.)

The choice of the parameters $\theta_{k+1/2}$ is somewhat analogous to that of the flux limiters $\phi(r)$ in the previous section. At the limit $\theta = 1$, and only then, one obtains a second-order accurate scheme; for linear advection and with $|\nu| \leqslant \frac{1}{2}$ it reduces to

$$U_k^{n+1} = [U - \nu\Delta_0\tilde{u} + \tfrac{1}{2}\nu^2\delta^2\tilde{u}]_k^n \tag{3.12a}$$

$$= [1 - \nu\Delta_0 + \tfrac{1}{2}(\nu^2 - \tfrac{1}{4})\,\delta^2]\,\tilde{u}_k^n. \tag{3.12b}$$

Note that this is an implicit scheme, through the "mass matrix" introduced in solving (3.11); further it always makes use of just three neighbouring values of $\tilde{u}_k^n$, the set of three depending on the interval in which ν lies. There is also a valuable stability margin so that (3.12), for instance, which is used for $|\nu| \leqslant \frac{1}{2}$ would, as a fixed difference scheme, be stable for $\nu^2 \leqslant \frac{1}{2}$. However, the scheme is not TVD, nor is it monotonicity preserving.

Thus in the spirit of the last section we shall consider choosing θ as large as possible while preventing the generating of spurious oscillations. However, note that we do this only at the recovery stage and rely on the evolution algorithm not to introduce any oscillations at that stage. We have found the most useful general criterion to be of the form

$$\Delta_+ U_k \geqslant 0 \Rightarrow \Delta_+ \tilde{u}_k \geqslant 0$$

and (3.13)

$$\Delta_+ U_k \leqslant 0 \Rightarrow \Delta_+ \tilde{u}_k \leqslant 0.$$

This ensures no new extrema are created. Unfortunately, because of (3.11), maximising θ subject to (3.13) is not a local problem. We have used two basic algorithms:

(A) Starting with $\tilde{u}_k^{(0)} = U_k\ \forall k$, use Jacobi iteration on the differenced form of (3.11),

$$(1 - \tfrac{1}{4}\theta_{k+1/2}^{(m)})\,\Delta_+\tilde{u}_k^{(m)} = \Delta_+ U_k - \tfrac{1}{8}\theta_{k-1/2}^{(m)}\Delta_+\tilde{u}_{k-1}^{(m-1)} - \tfrac{1}{8}\theta_{k+1/2}^{(m)}\Delta_+\tilde{u}_{k+1}^{(m-1)} \tag{3.14a}$$

with the θ parameters determined by difference ratios calculated from either two or three elements either side of the element boundary:

$$\theta^{(m)}_{k+1/2} = 1/\max\{1, \sigma^{+}_{k+1/2}, \sigma^{-}_{k+1/2}\}. \tag{3.14b}$$

Using just two elements either side, we can set

$$\sigma^{+}_{k+1/2} = \tfrac{1}{4}\Delta_+ \tilde{u}^{(m-1)}_k/\Delta_+ U_{k+1}, \qquad \sigma^{-}_{k+1/2} = \tfrac{1}{4}\Delta_+ \tilde{u}^{(m-1)}_k/\Delta_+ U_{k-1} \tag{3.14c}$$

and this will ensure (3.13) is satisfied at each stage of the iteration. Similarly, (3.13) is satisfied if three elements are used to replace (3.14c) by

$$\sigma^{+}_{k+1/2} = \tfrac{1}{16}[\Delta_+ \tilde{u}^{(m-1)}_k/\Delta_+ U_{k+1}](3 + \operatorname{sgn} r_{k+2}), \tag{3.14d}$$

where $r_k = \Delta_+ U_{k-1}/\Delta_+ U_k$, and the corresponding expression for $\sigma^{-}_{k+1/2}$ involving r_{k-1}; numerical experiments show that (3.14d) gives significantly better results. Alternatively, the Jacobi iteration may be used only a few times and then (3.11) solved with the last set of θ's to maintain the projection property (3.7). We cannot then guarantee the property (3.13) but good results have been obtained in practice in this way. In particular, with no Jacobi iterations but using just (3.14b) to go immediately to (3.11) the $\sigma^{+}_{k+1/2}$ of (3.14c) reduces to $\frac{1}{4}r_{k+1}$ and the $\sigma^{-}_{k+1/2}$ to $1/(4r_k)$ and gives a very simple algorithm; moreover, if the choice (3.14b) is overridden to give $\theta^{(m)}_{k+1/2} = 1$ if either $r_{k-1} < 0$ or $r_{k+2} < 0$, excellent results are obtained.

(B) The alternative type of algorithm that we have used starts with large values of θ and then reduces them locally to satisfy (3.13). Poor convergence is experienced if one starts with $\theta^{(0)}_{k+1/2} = 1\ \forall k$. We have therefore used values obtained from considering the local discrete Green's function for (3.11) with constant θ and smooth U_k. This gives

$$\theta^{(0)}_{k+1/2} = \max\{0, \min[1, \phi_{k-1/2}, \phi_{k+1/2}, \phi_{k+3/2}]\}, \tag{3.15a}$$

where

$$\phi_{k+1/2} := \frac{8\Delta_+ U_k(\Delta_+ U_{k+1} + \Delta_+ U_{k-1})}{(\Delta_+ U_{k+1} + \Delta_+ U_k + \Delta_+ U_{k-1})^2}$$

$$= \frac{8(2 + \kappa_{k+1/2})}{(3 + \kappa_{k+1/2})^2}, \qquad \text{where} \quad \kappa_{k+1/2} = \frac{\delta^2 \Delta_+ U_k}{\Delta_+ U_k}. \tag{3.15b}$$

The parameter $\kappa_{k+1/2}$ is one whose significance has been noted by others engaged in shock modelling. Failure of the criterion (3.13) occurs where the parameter

$$\tau_{k+1/2} := (\Delta_+ U_k - \Delta_+ \tilde{u}_k)/\Delta_+ U_k \tag{3.15c}$$

is greater than unity for any k: a set of three neighbouring θ values are then reduced by setting

$$\theta^{(m)}_{k+1/2} = \theta^{(m-1)}_{k+1/2}/\max\{1, \tau_{k-1/2}, \tau_{k+1/2}, \tau_{k+3/2}\} \tag{3.15d}$$

before recalculating $\tilde{u}$ from (3.11). This algorithm converges very quickly and gives excellent results.

3.3. *Update Algorithm with Linear Recovery*

We have already noted that without recovery the update algorithm for the ECG scheme is identical to the Engquist–Osher difference scheme, the increment form with the increments given by (3.6) being the most natural. After recovery with piecewise linears it is again natural to break up the contributions to the update, this time into those from each interval over which $\tilde{u}$ varies linearly. First we combine (3.7) and (3.8) in the form

$$\langle U^{n+1}, \phi_i \rangle = \langle \tilde{u}^n, \phi_i \rangle - \Delta t \langle \partial_x f(\tilde{u}^n), \tilde{\Phi}_i^n \rangle. \tag{3.16}$$

Then the first integral on the right can be broken up into

$$\begin{aligned} \langle \tilde{u}^n, \phi_i \rangle &= \tilde{u}_i^n \, \Delta x + \Sigma_{(k)} \int_{x_k}^{x_{k+1}} [\tilde{u}^n(x) - \tilde{u}_i^n] \, \phi_i(x) \, dx \\ &= \tilde{u}_i^n \, \Delta x + \Sigma_{(k)} \int_{x_k}^{x_{k+1}} [\tilde{u}^n(x) - \tilde{u}_i^n] \, d \int_{x_{k+1/2}}^{x} \phi_i(s) \, ds \\ &= \tilde{u}_i^n \, \Delta x - \Sigma_{(k)} \int_{\tilde{u}_k^n}^{\tilde{u}_{k+1}^n} \left[\int_{x_{k+1/2}}^{x} \phi_i(s) \, ds \right] du, \end{aligned} \tag{3.17a}$$

where we have exploited the linearity to change the integral over x to one over u. We treat the last part of (3.16) in a similar way to obtain, from the definition of $\tilde{\Phi}_i^n$ and using $y = x + a(\tilde{u}^n(x)) \, \Delta t$,

$$\begin{aligned} -\Delta t \langle \partial_x f(\tilde{u}^n), \tilde{\Phi}_i^n \rangle &= -\Sigma_{(k)} \int_{x_k}^{x_{k+1}} \partial_x \tilde{u}^n \left[\int_x^y \phi_i(s) \, ds \right] dx \\ &= -\Sigma_{(k)} \int_{\tilde{u}_k^n}^{\tilde{u}_{k+1}^n} \left[\int_x^y \phi_i(s) \, ds \right] du. \end{aligned} \tag{3.17b}$$

We can combine these formulae to obtain

$$\Delta x (U_i^{n+1} - \tilde{u}_i^n) = -\Sigma_{(k)} \int_{\tilde{u}_k^n}^{\tilde{u}_{k+1}^n} \left[\int_{x_{k+1/2}}^{y} \phi_i(s) \, ds \right] du. \tag{3.18}$$

Here $y(x)$ is defined implicitly through $\tilde{u}^n(x)$ over the linear section centred at $x_{k+1/2}$. Therefore we introduce

$$A_{k+1/2}(\tilde{u}) := a(\tilde{u}) + (\tilde{u} - \tilde{u}_{k+1/2})/(\tilde{m}_{k+1/2} \, \Delta t) \tag{3.19a}$$

and

$$F_{k+1/2}(\tilde{u}) := f(\tilde{u}) + \tfrac{1}{2}(\tilde{u} - \tilde{u}_{k+1/2})^2/(\tilde{m}_{k+1/2} \, \Delta t), \tag{3.19b}$$

whereby $A_{k+1/2} = \partial F_{k+1/2}/\partial \tilde{u}$ and from which we can substitute into (3.18)

$$y = x_{k+1/2} + A_{k+1/2}(\tilde{u})\, \Delta t. \tag{3.20}$$

Various explicit algorithms can then be derived from (3.18) and (3.20)

The simplest and most important algorithm for our present purposes comes from assuming

$$|a(\tilde{u})|\ (\Delta t/\Delta x) + \tfrac{1}{2}\theta_{k+1/2} \leqq 1 \tag{3.21}$$

for $\tilde{u}$ between $\tilde{u}_k^n$ and $\tilde{u}_{k+1}^n$; for then the contribution from this section of the graph can only go to update either U_k^n or U_{k+1}^n. We denote by $\tilde{u}_{k+1/2}^{(l)}$ for $l = 1, 2, ..., m$ any sonic points of $F_{k+1/2}(\tilde{u})$ between $\tilde{u}_k^n$ and $\tilde{u}_{k+1}^n$, with $\tilde{u}_{k+1/2}^{(0)} := \tilde{u}_k^n$ and $\tilde{u}_{k+1/2}^{(m+1)} := \tilde{u}_{k+1}^n$ and $F_{k+1/2}^{(l)} := F_{k+1/2}(\tilde{u}_{k+1/2}^{(l)})$. Then using λ as before to denote the mesh ratio $\Delta t/\Delta x$, we obtain the simple update algorithm to be executed for each k to obtain $\{U_k^{n+1}\}$ from $\{\tilde{u}_k^n\}$: for $l = 0, 1, 2, ..., m$,

$$add\ -\lambda[F_{k+1/2}^{(l+1)} - F_{k+1/2}^{(l)}] \qquad to \quad \begin{cases} \tilde{u}_{k+1}^n \\ \tilde{u}_k^n \end{cases} \qquad if \quad \begin{cases} A_{k+1/2} > 0 \\ A_{k+1/2} < 0 \end{cases} \tag{3.22}$$

in the subinterval $u_{k+1/2}^{(l)}$ to $u_{k+1/2}^{(l+1)}$. That is, we have an Engquist–Osher algorithm of the form (2.19) or (3.6) for the modified flux function. Because a quadratic has been added to f to obtain $F_{k+1/2}$, this may contain more than one sonic point even when f is convex; but in the particular cases occurring in gas dynamics this complication never arises.

In earlier work (see [17]) the same scheme was presented differently, without introducing the functions of (3.19) and instead using "crossing points" $\{x_{k+1/2}^{(l)}, l = 1, 2, ..., m\}$, where $y(x)$ crossed $y = x_{k+1/2}$; that is, solutions of

$$z + a(\tilde{u}^n(z))\, \Delta t = x_{k+1/2}. \tag{3.23}$$

Using these and keeping the integrals in (3.17) over x rather than over u, one obtains a form close to the increment and transfer formulation given in (2.16) and (2.17). Omitting details which are given in [17] and using the notation $\tilde{f}_{k+1/2}^{(l)} := f(\tilde{u}^n(x_{k+1/2}^{(l)}))$, $\tilde{v}_{k+1/2}^{(l)} := \lambda a(\tilde{u}^n(x_{k+1/2}^{(l)}))$, we have the following algorithm for updating $\{U_k^n\}$ to $\{U_k^{n+1}\}$ when (3.21) holds:

(i) for $l = 0, 1, ..., m$,

$$add\ -\lambda[\tilde{f}_{k+1/2}^{(l+1)} - \tilde{f}_{k+1/2}^{(l)}] \qquad to \quad \begin{cases} U_{k+1}^n \\ U_k^n \end{cases} \qquad if \quad \begin{cases} y > x_{k+1/2} \\ y < x_{k+1/2} \end{cases} \tag{3.24a}$$

in the sub-interval $(x_{k+1/2}^{(l)}, x_{k+1/2}^{(l+1)})$;

(ii) for $l = 1, 2, ..., m$,

$$\text{transfer } \tfrac{1}{2}\tilde{m}_{k+1/2}(\tilde{v}^{(l)}_{k+1/2})^2\,\Delta x \quad \text{from} \quad \begin{cases} U^n_k \\ U^n_{k+1} \end{cases} \quad \text{to} \quad \begin{cases} U^n_{k+1} \\ U^n_k \end{cases} \tag{3.24b}$$

according to whether $y(x)$ is increasing or decreasing at $x^{(l)}_{k+1/2}$;

(iii) if $\tilde{v}_k > \tfrac{1}{2}\theta_{k+1/2}$ and again (independently) if $\tilde{v}_{k+1/2} < -\tfrac{1}{2}\theta_{k+1/2}$,

$$\text{transfer } \tfrac{1}{8}\theta_{k+1/2}\Delta_+\tilde{u}^n_k \quad \text{from} \quad U^n_k \quad \text{to} \quad U^n_{k+1}. \tag{3.24c}$$

Suppose that $f(u)$ is convex so that $a'(u) > 0$; then since $y(x)$ is increasing if $1 + a'(\tilde{u})\,\tilde{m}_{k+1/2}\,\Delta t > 0$, there cannot be more than one crossing point unless the gradient of $\tilde{u}$ is negative and quite large. Thus if shocks are recovered separately [illegible] none with $m = 0$, and for large θ, normally just one and $m = 1$.

3.4. *Shock Recovery and Update Algorithm*

It remains to describe the shock, or more generally jump, recovery algorithm. We have found the most reliable criteria for determining whether a jump should be recognised as occurring in element k, and retained there in the recovered function $\tilde{u}^n$, are as follows:

$$a(U^n_{k-1}) - a(U^n_{k+1}) \geqslant 0 \tag{3.25a}$$

and

$$r_k > 0, \qquad |r_{k-1}| \ll 1, \qquad |r_{k+1}| \gg 1, \tag{3.25b}$$

where r_k is the ratio of successive differences

$$r_k := (U^n_k - U^n_{k-1})/(U^n_{k+1} - U^n_k). \tag{3.26}$$

The tolerances in (3.25b) have so far been selected by numerical experience.

Combination of jump recovery with the piecewise linear recovery has been carried out as described in [15] and as formally defined in (3.10). For completeness we merely summarise the update process corresponding to a jump in element k. After solving the tridiagonal systems of the form (3.11) either side of element k, one has a jump from $\tilde{u}^n_{k+1}$ to $\tilde{u}^n_{k+1}$ at a fractional position η (see Fig. 2) given by

$$(1-\eta)\,\tilde{u}^n_{k+1} + \eta\tilde{u}^n_{k-1} = U^n_k. \tag{3.27}$$

We introduce the CFL number corresponding to the jump

$$\tilde{v}_k := \lambda\Delta_0 f(\tilde{u}^n_k)/\Delta_0\tilde{u}^n_k \tag{3.28}$$

and allocate the contribution

$$-\Delta t[f(\tilde{u}^n_{k+1}) - f(\tilde{u}^n_{k-1})] \tag{3.29}$$

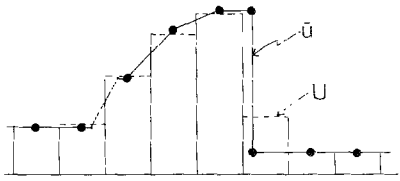

FIG. 2. Sketch of piecewise constant U and corresponding $\tilde{u}$ recovered with a combination of shocks and piecewise linears.

to the three elements $k-1$, k and $k+1$ in the following proportions:

	$k-1$	k	$k+1$
$-1<\tilde{\nu}+\eta<0$	$1+\eta/\tilde{\nu}$	$-\eta/\tilde{\nu}$	0
$0\leqslant\tilde{\nu}+\eta\leqslant 1$	0	1	0
$1<\tilde{\nu}+\eta<2$	0	$(1-\eta)/\tilde{\nu}$	$1-(1-\eta)/\nu$

(3.30)

In summary, the main components of the ECG algorithm used in the numerical comparisons of the next section are as follows:

(i) identify any shocks by means of (3.25), (3.26);

(ii) carry out the linear recovery between the shocks using (3.14) or (3.15);

(iii) update the solution using (3.19), (3.22) and the shock update (3.27)–(3.30).

Coded in a straightforward manner the algorithm takes roughly twice as long as the corresponding flux-limiter programs on all the scalar problems. However, with little extra complication the restriction (3.21) can be set aside and the more general algorithm runs most effectively with CFL numbers larger than unity.

4. SYSTEMS OF EQUATIONS

We now consider extension of the scalar algorithms of Sections 2 and 3 to systems of equations, in particular, to the Euler equations of compressible gas dynamics.

For the Godunov and Engquist Osher difference schemes of Section 2 there exist direct counterparts for systems of equations. However, the former is based on the solution of a one-dimensional Riemann problem at each element boundary and the second as extended by Osher and Solomon [21] involves the solution of similar problems, using overturned manifolds or folded characteristic fields rather than shocks in the case of compressive waves. Either approach makes for a rather complicated algorithm even for a first-order scheme.

A simpler approach is that due to Roe [22]. This uses a Jacobian $A = \partial \mathbf{f}/\partial \mathbf{w}$ averaged over each cell $(k, k+1)$ to linearise the differential system, so decomposing the vector of flux differences $\delta \mathbf{f}_{k+1/2}$ into characteristic fields proportional to the right eigenvectors of A. Then the Cole–Murman scheme (2.18), or flux-limited second-order variants of the form (2.17) can be applied to each field independently.

For any first-order scheme, when a limited anti-diffusive flux is added to improve the resolution there is one main difference from the scalar case described in Section 2: for each characteristic field the left- and right-moving flux differences $(\Delta f^n_{k+1/2})^{\pm}$ of (2.10) are now vectors. Thus the ratios $r_k^{\pm}$ of (2.14) used to determine the flux limiters have to be redefined by selecting an appropriate vector $\mathbf{z}$ and replacing $(\Delta f^n_{k+1/2})^{\pm}$ in (2.14) by

$$\mathbf{z}^{\mathrm{T}}(\Delta \mathbf{f}^n_{k+1/2})^{\pm}. \tag{4.1}$$

In all cases we have chosen the density as the most sensitive indicator for the flux limiters and therefore for the usual definitions of the characteristic fields (see below) taken $\mathbf{z}^{\mathrm{T}} = (1, 0, 0)$.

As pointed out at the end of Section 2, a disadvantage of Roe's original decomposition of flux differences is that not enough information in given to treat transonic expansion waves correctly. An approximate interpolated sonic point as given in (2.20) is sufficient to overcome this when the Roe difference scheme is applied to the Euler equations because one has only to use the appropriate characteristic speed for each CFL number in (2.20) along with the $\delta \mathbf{w}_{k+1/2}$ for the corresponding characteristic field to replace $\delta u_{k+1/2}$. However, in studying various ways of extending the ECG schemes of Section 3 to systems of equations, we have not so far seen an effective way of using the Roe decomposition except for treating jump recovery.

Thus consider the piecewise constant ECG scheme with piecewise linear recovery which led to the algorithm given by (3.22). Instead of decomposing the flux or state vector differences (i.e., flux difference splitting), we decompose into characteristic fields in each element (i.e., flux vector splitting). For the Euler equations

$$\partial_t \begin{pmatrix} \rho \\ \rho u \\ e \end{pmatrix} + \partial_x \begin{pmatrix} \rho u \\ p + \rho u^2 \\ u(e+p) \end{pmatrix} = 0 \tag{4.2}$$

we decompose $\mathbf{w} = (\rho, \rho u, e)^{\mathrm{T}}$ into the three characteristic vectors

$$\frac{1}{2\gamma} \begin{pmatrix} \rho \\ \rho(u-a) \\ \rho(H-ua) \end{pmatrix}, \qquad \frac{\gamma - 1}{\gamma} \begin{pmatrix} \rho \\ \rho u \\ \tfrac{1}{2}\rho u^2 \end{pmatrix}, \qquad \frac{1}{2\gamma} \begin{pmatrix} \rho \\ \rho(u+a) \\ \rho(H+ua) \end{pmatrix} \tag{4.3}$$

corresponding to the characteristic speeds $u-a$, u, $u+a$, respectively; here $a^2 = \gamma p/\rho$ and $H = (e+p)/\rho$, the enthalpy. We denote the vectors (4.3) by $\mathbf{w}^{(1)}$, $\mathbf{w}^{(2)}$, $\mathbf{w}^{(3)}$ and the speeds by $\lambda^{(1)}$, $\lambda^{(2)}$, $\lambda^{(3)}$. Note that $\lambda^{(m)}$ is given by the ratio of the

second to the first component of $\mathbf{w}^{(m)}$ in each case and that the corresponding fluxes are $\mathbf{f}^{(m)} = \lambda^{(m)}\mathbf{w}^{(m)}$, because the homogeneity of the equations implies that $\mathbf{f} = Aw$. (Note, this splitting of the flux vector is one of those analysed by Steger and Warming [25].)

We have used the density, that is, the first component of each $\mathbf{w}^{(m)}$, to select the $\{\theta_{k+1/2}\}$ for the linear recovery using one of the algorithms of (3.14) or (3.15). Then from (3.11) we obtain the three characteristic fields $\tilde{\mathbf{w}}_k^{(m)}$ at each nodal point x_k. To implement the update process of (3.22) we construct the modified fluxes $F^{(m)}$ and characteristic speeds $A^{(m)}$ by generalising (3.19) for each characteristic field; to simplify the notation we suppress the superscript (m) and obtain

$$A_{k+1/2}(\tilde{\mathbf{w}}) := \lambda(\tilde{\mathbf{w}}) + \theta_{k+1/2}\frac{\Delta x}{\Delta t}\frac{\tilde{\rho} - \tilde{\rho}_{k+1/2}}{\tilde{\rho}_{k+1} - \tilde{\rho}_k} \tag{4.4a}$$

$$F_{k+1}(\tilde{\mathbf{w}}) := \lambda(\tilde{\mathbf{w}})\,\tilde{\mathbf{w}} + \tfrac{1}{2}\theta_{k+1/2}\frac{\Delta x}{\Delta t}\frac{\tilde{\rho} - \tilde{\rho}_{k+1/2}}{\tilde{\rho}_{k+1} - \tilde{\rho}_k}(\tilde{\mathbf{w}} - \tilde{\mathbf{w}}_{k+1/2}). \tag{4.4b}$$

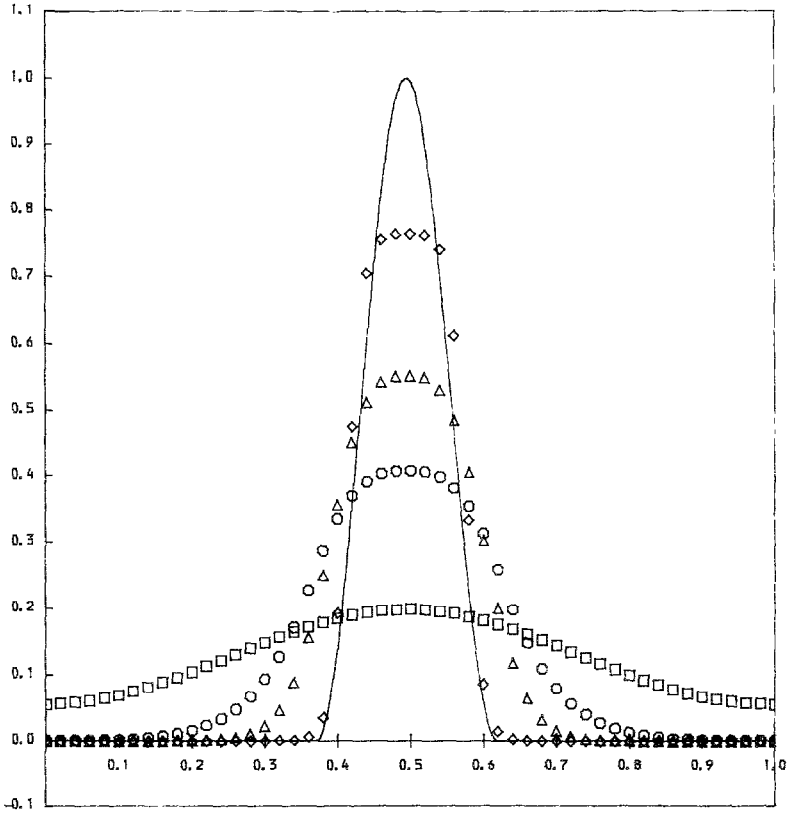

FIG. 3. Linear advection by difference schemes: □ = 1st order, no limiter; ○ = minmod limiter; ◇ = superbee; △ = van Leer.

Here all the components of $\tilde{\mathbf{w}}$ vary linearly between $\tilde{\mathbf{w}}_k$ and $\tilde{\mathbf{w}}_{k+1}$ with $\tilde{\mathbf{w}}_{k+1/2}$ the mid-point; and we exploit the points noted above regarding determination of $\lambda(\tilde{\mathbf{w}})$ and $f(\tilde{\mathbf{w}})$. As a result, the calculation of the sonic points $\tilde{\mathbf{w}}$ at which $\Lambda_{k+1/2}=0$ reduces to the solution of a quadratic, the correct root being obvious.

Jumps (that is, shocks and contact discontinuities) are recognised by using the criteria of (3.25b) applied to the density, but with the criterion (3.25a) applied to all the characteristic fields. Having detected a jump in the solution, jump recovery is implemented using Roe's decomposition [22] locally and then performing the scalar algorithm (3.30) on each field of the decomposition. That is, if a jump is detected in element k, we set the surrounding θ's, $\theta_{k-1/2}$ and $\theta_{k+1/2}$, to zero and decompose the jump $\tilde{\mathbf{u}}_{k-1}$, $\tilde{\mathbf{u}}_{k+1}$ using Roe's decomposition. The position of the jump for each field is found using this and another decomposition on $\tilde{\mathbf{u}}_k$, $\tilde{\mathbf{u}}_{k+1}$ via (3.27). The distribution (3.30) is then applied to the flux differences given by the decomposition, field by field.

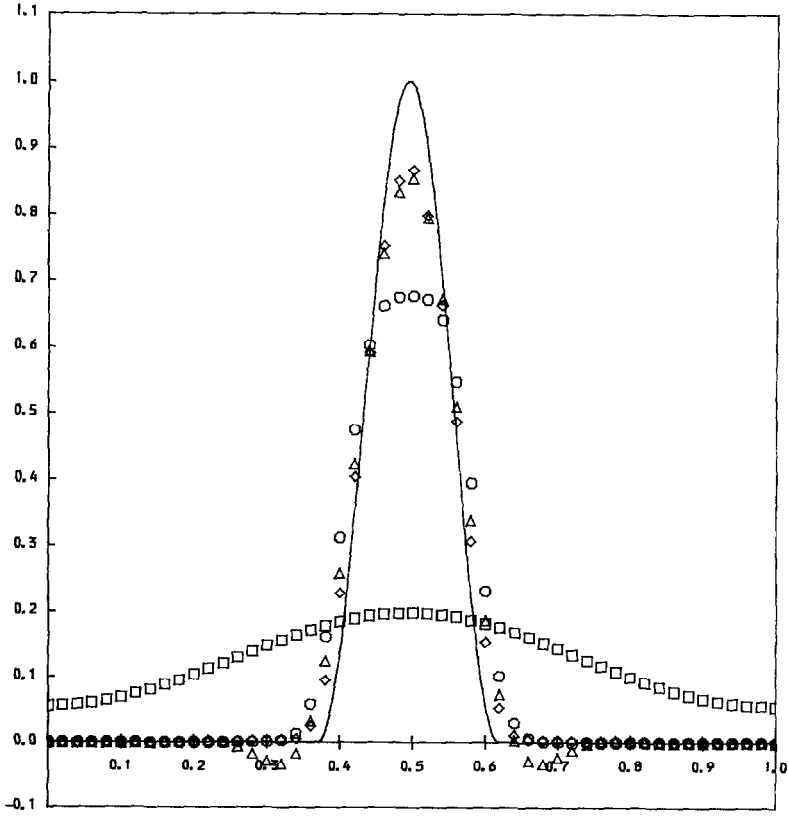

FIG. 4. Linear advection by ECG schemes: □ = no recovery; ○ = simple recovery (3.14a, b, c)); △ = fixed $\theta = 1$ recovery; ◇ = iterated recovery (3.15).

5. Numerical Results

We present results here of computations with the above methods for the linear advection equation, the inviscid Burgers equation and the Euler equations. The capabilities of the flux-limited difference schemes are well known so we use these mainly as yardsticks by which to judge the ECG methods.

Linear advection is one of the most severe tests of accuracy for a general method since, as with contact discontinuities, the solution operator is completely neutral—giving neither steepening nor smoothing. As is now common practice we use a large number of timesteps, 612, with compact data on a periodic domain [0, 1]. The first tests are with the initial pulse $\sin^2 \pi(4x-1)$ on $(\frac{1}{4}, \frac{1}{2})$ advected with unit speed; results are given in Figs. 3 and 4, with $\Delta x = 0.02$ and $\Delta t = 0.01$. In Fig. 3 are shown the results for the various limiters given in Section 2 and represented on Fig. 1; clearly the superbee limiter gives the greatest improvement over the first order upwind scheme and none of course give any undershoot. In Fig. 4 are shown the

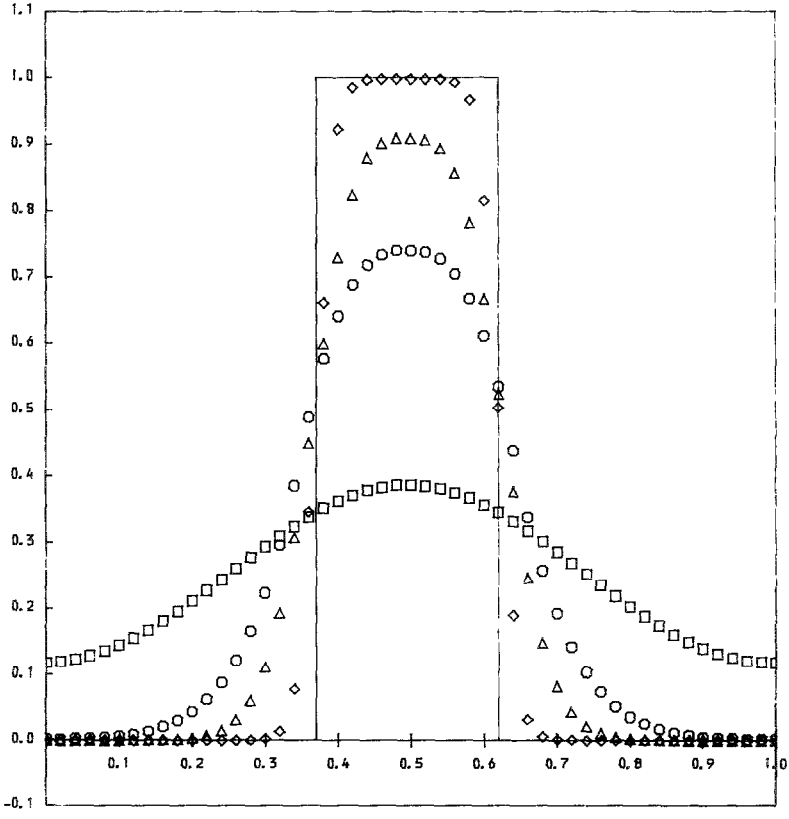

FIG. 5. As Fig. 3, with square pulse data.

corresponding results for the piecewise constant ECG scheme with various linear recovery schemes; the non-adaptive $\theta = 1$ recovery gives a better peak value but, as expected, gives undershoots at the leading and trailing edges. Simple recovery based on (3.14a, b, and c) without iteration gives a result very similar in form to the flux-limited scheme but not as good as superbee; but the more acccurate recovery with (3.15) gives a peak value appreciably higher than superbee and with negligible flattening. Very comparable results can be obtained if (3.14d) is used instead of (3.14c), showing that careful treatment of the situation when the differences change sign is all important.

Similar tests on the advection of a square pulse which is initially on $(\frac{1}{4}, \frac{1}{2})$ give the results of Fig. 5 for three of the flux-limiters; and in Fig. 6 are shown the results for the ECG schemes. Here we see that the superbee flux-limiter performs considerably better than the ECG schemes using linear recovery. However, we would not normally expect to use recovery with piecewise linears on such data: and, as is shown,

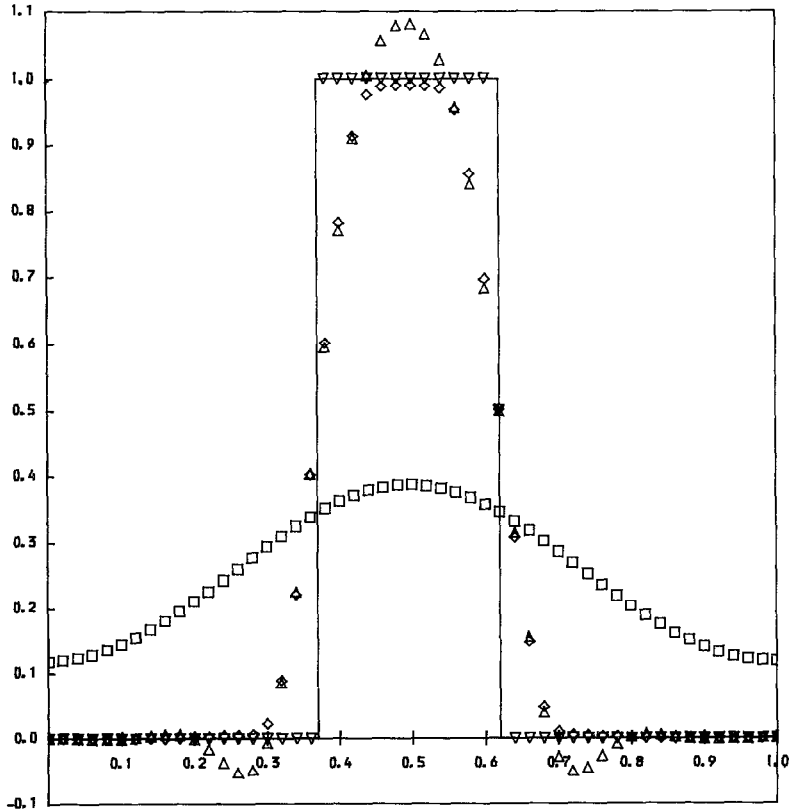

FIG. 6. As Fig. 4, with square pulse data: ∇ = jump recovery.

application of the jump recovery algorithm is successful in recovering both jumps at each timestep so the exact solution is obtained.

The next tests are with the inviscid Burgers equation, starting with the same square pulse except that it is shifted on to a base at $-\frac{1}{4}$ to check the effect of the sonic points. Again results are given for $\Delta x = 0.02$ and $\Delta t = 0.01$, this time after 20 and 140 timesteps. In Fig. 7 we show those obtained with the first-order Engquist–Osher scheme and with the superbee limiter; the latter considerably sharpens up the shock and removes the dog-leg obtained with the E.O. scheme at the rarefaction sonic point; other limiters give similar but not quite such good results. The ECG schemes give the results shown in Fig. 8; the non-adaptive linear recovery is not shown as it gives oscillations going up to 0.6 at the later time. The iterated linear recovery of (3.11) gives results very similar to but not quite as good as superbee; the jump recovery improves on this, capturing the shock with at most one intermediate point.

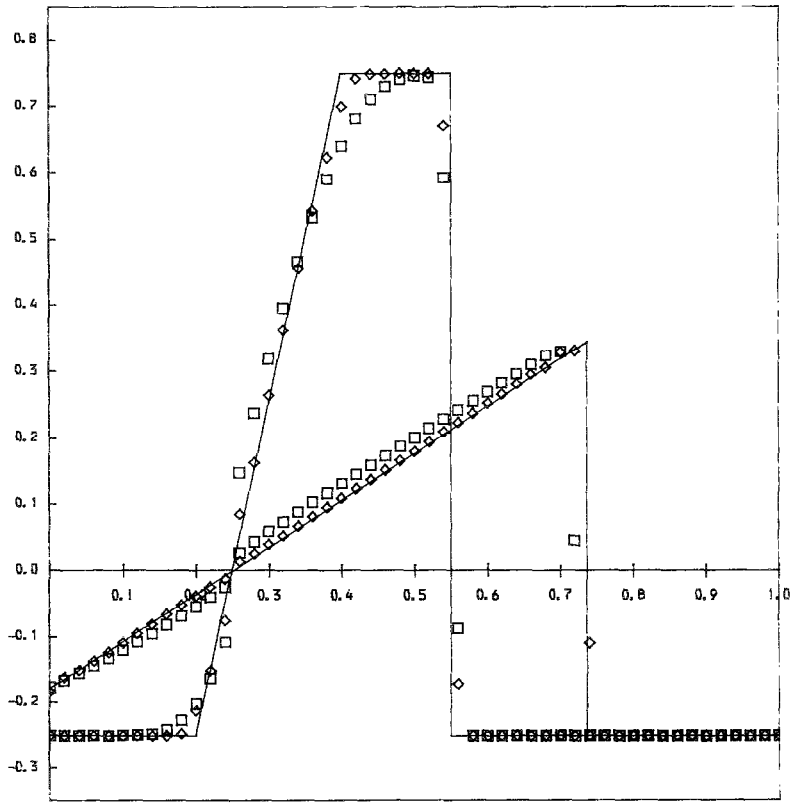

FIG. 7. Inviscid Burgers' equation by difference schemes: □ = Engquist–Osher, no limiter; ◇ = with superbee limiter.

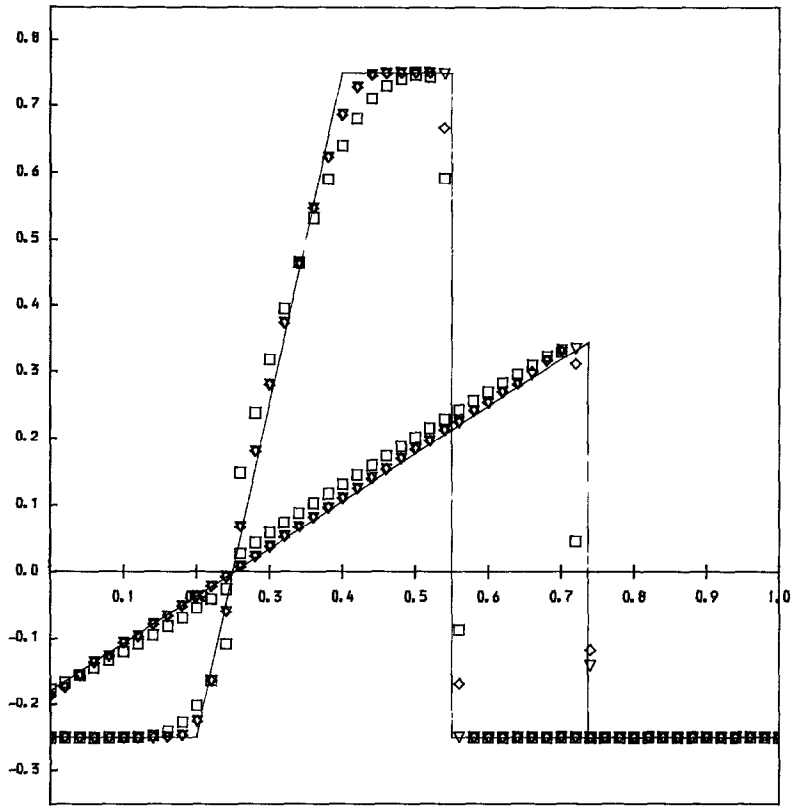

FIG. 8. Inviscid Burgers' equation by ECG schemes: □ = no recovery; ◇ = iterated recovery (3.15); ▽ = iterated plus jump recovery.

Finally we have used Sod's standard shock-tube problem to test the algorithm for the Euler equations. The standard parameters are: $\rho_L = 1.0$, $p_L = 1.0$, $u_L = 0.0$, $\rho_R = 0.1$, $p_R = 0.125$, and $u_R = 0.0$.

We have used $\Delta x = 0.02$ and $\Delta t = 0.003$ and display the results after 48 steps at $t = 0.144$ in Figs. 9–13. As can be seen, from Figs. 9 and 11, the first-order methods both give poor resolution of the discontinuities; but the flux vector splitting, on which the ECG schemes have been based, has greater difficulty in picking up the foot of the expansion fan (see Fig. 11) than the Roe decomposition used in Fig. 9. This difference continues into the higher order schemes: the results of Fig. 10 produced with Roe's decomposition and the superbee limiter are the best of those shown; while the use of iterative linear recovery (3.15) in the ECG scheme to give the results of Fig. 12 shows a marked improvement over those of Fig. 11, they are still not as good as Fig. 10. We note here that simple linear recovery produces very similar results, the full advantage of the more complex recovery not being so noticeable after only a small number of timesteps as taken here.

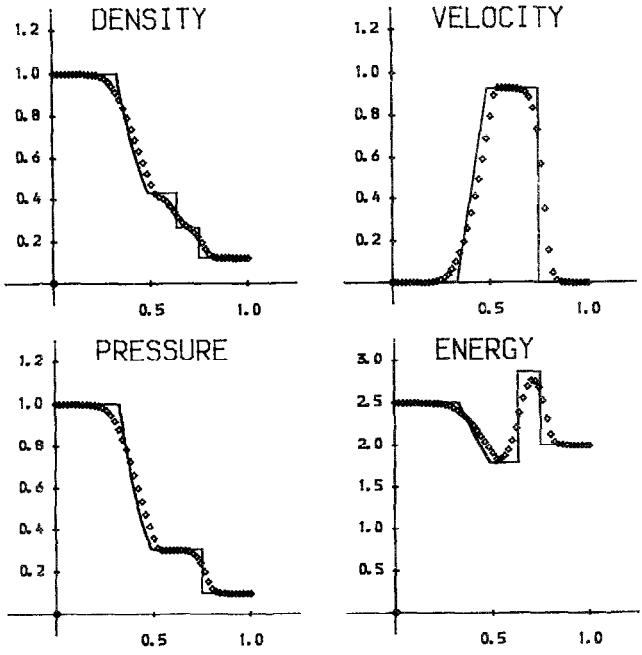

FIG. 9. Shock tube problem using Roe decomposition with 1st-order scheme and no limiter.

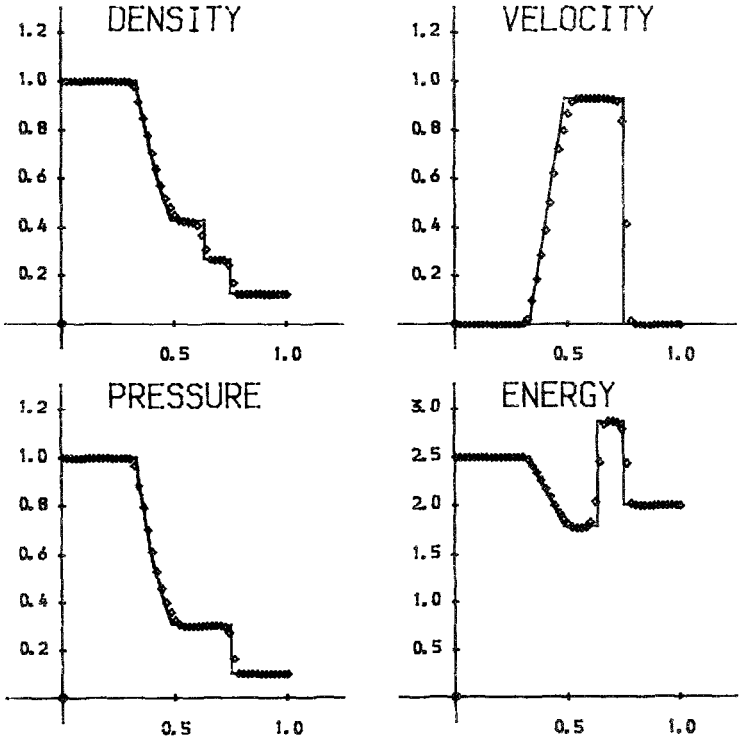

FIG. 10. As Fig. 9 with superbee limiter.

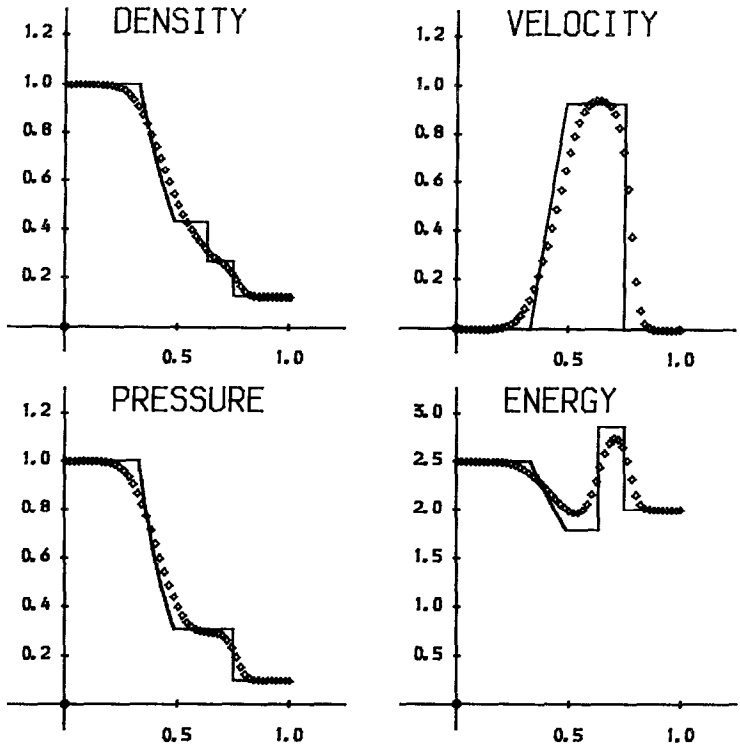

FIG. 11. Shock tube problem with ECG scheme and no recovery.

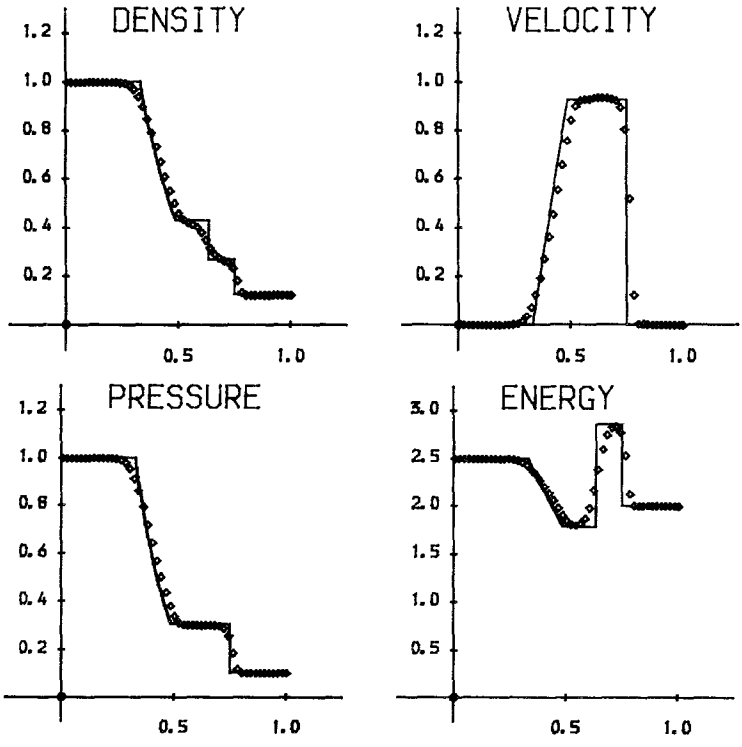

FIG. 12. As Fig. 11 with linear recovery.

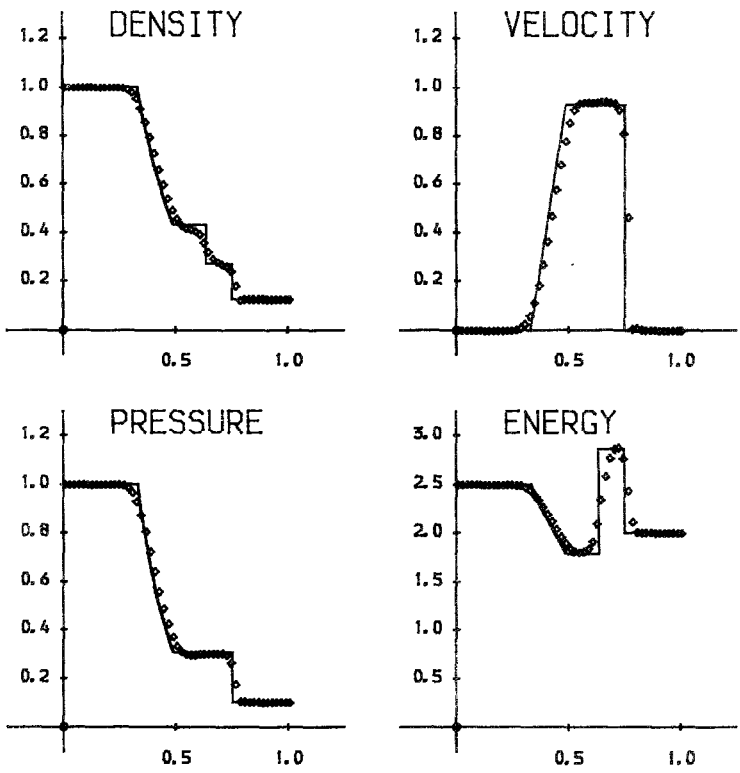

FIG. 13. As Fig. 10 with linear and jump recovery.

The addition of jump recovery to the linear recovery (Fig. 13) sharpens the shock slightly, giving comparable definition to that produced by superbee, but has no effect on the contact discontinuity. Numerical tests starting with data at a time greater than zero indicate that this due to the difficulty in detecting the contact when it resides in the same or adjacent element as the shock, as is the case initially. The jump recovery algorithm that we have described above can be used only when jumps are at least two elements apart. As the shock-tube problem starts with the shock and contact discontinuity coincident, much of the error is generated in the early stages. We have considered using jump recovery algorithms which will deal with closer jumps. For example, in the scalar case one can recognise and recover jumps in elements $(k-1)$ and $(k+2)$ (near neighbour recovery) if $|r_{k-2}| \ll 1$, $|r_{k+2}| \gg 1$ with r_{k-1}, r_k and $r_{k+1} > 0$. The resolution of jumps in neighbouring elements can also be achieved in the case of systems of equations. However, we have not pursued these sophistications further so as not to complicate the task of extending the algorithms to two dimensions.

6. Conclusions

Our objective in this paper has been to take a highly developed class of difference schemes for an important and difficult set of problems and to see whether recent developments of finite element methods can produce similar high quality results. In doing so we have maintained a purist approach to finite element methods; that is, we have avoided the borrowing of finite difference techniques which would invalidate the comparison. Of course, we fully appreciate that ultimately the most useful methods will probably use ideas from both viewpoints.

A finite difference approach to approximating a differential problem uses Taylor series in some form (or several) to approximate the differential or integral form of the operators involved in the problem. There are few general rules and a great variety of schemes result which are applicable to a wide range of problems. A finite element approach is both more disciplined and more limited. The emphasis is on a given functional form for the approximate solution which has to be used consistently throughout the formulation; then some sort of global principle is needed to obtain a defining set of algebraic equations, typically a variational principle, which may limit the applicability of the method.

Modelling hyperbolic conservation laws provides an interesting area for comparison of the two approaches; but one has to compare like with like. One could compare conventional difference schemes like Lax–Wendroff or similar higher order schemes with straightforward weak formulations of finite element methods such as Galerkin, Petrov–Galerkin, or Taylor–Galerkin. This has been done to some extent in [16, 17], and references quoted therein, where these generalised Galerkin methods were related both to each other and to the characteristic Galerkin methods considered here.

The schemes compared in the present paper have a common ancestry in the first-order method of Godunov [7] which can be regarded as either a finite difference or a finite element method. Those of Section 2 have been developed using higher order difference schemes in an adaptive manner and approximate Riemann solvers to retain the best features of the Godunov method while achieving higher accuracy economically. The finite element schemes of Section 3 have used the same piecewise constant approximation as [7] but exploited its projection property to obtain higher accuracy when justified by smoothness, again adaptively; the formulation is based on an approximate evolution operator so as to give a clearly defined method which can also be simply implemented. The result of our comparison has been to show that the different approaches working at comparable orders of approximation not only yield similar levels of accuracy but also algorithms which have similar structures. This has been achieved despite staying strictly within the more formal finite element framework, in contrast to developments of the Petrov–Galerkin methods given in [17] which have been carried out, e.g., in [9], by applying the flux-limiter difference technique in an ad hoc fashion.

For the one-dimensional problems considered here, this can be regarded as a merely academic point. However, some advantages that the finite element approach

has as it moves from this common base of accuracy are already apparent. We see from (3.8) that it takes a non-uniform mesh in its stride. Also arbitrary timesteps can be taken on such a mesh with no loss of stability; one merely has to solve (3.20) for $y = x_{k+p+1/2}$ with $p \neq 0$ rather than finding the sonic points which correspond to $p = 0$. As pointed out by Leveque [13], if the evolution operator one is using does not misrepresent too badly the wave interactions that occur in a longer timestep, one can gain accuracy by increasing Δt because of the fewer projections that are involved.

However, the final payoff has to be sought in multi-dimensional problems and, indeed, the interest in non-uniform meshes and arbitrary CFL numbers also stems from this. Extending the TVD concept and the flux-limited difference schemes without making use of fractional step methods is notoriously difficult. On the other hand, the transport collapse operator of [3] is dimensionally independent and one can show that on a square mesh with piecewise contant elements the Engquist–Osher scheme is augmented by cross-differenced terms arising from interactions at the corners. Also the closely related Lagrange–Galerkin methods are widely applied in two- and three-dimensional problems very successfull using piecewise linear or multi-linear elements. The recovery methods described in the present paper are still under development for higher dimensions but preliminary results are encouraging.

ACKNOWLEDGMENTS

Our thanks are due to the referees whose perceptive comments and careful reading of the first version of this paper have greatly assisted us in the revision.

REFERENCES

1. M. J. BAINES AND P. K. SWEBY, Reading University Numerical Analysis Report 9/84, 1984 (unpublished).
2. J. P. BORIS AND D. L. BOOK, *J. Comput. Phys.* **11**, 38 (1973).
3. Y. BRENIER, *SIAM J. Numer. Anal.* **21**, 1013 (1984).
4. S. CHAKRAWARTHY AND S. OSHER, AIAA Paper 83/943, Proceedings, AIAA Sixth Computational Fluid Dynamics Conference, 1983, p. 363.
5. B. ENGQUIST AND S. OSHER, *Math. Comput.* **34**, 45 (1980).
6. J. E. FROMM, *J. Comput. Phys.* **3**, 176 (1968).
7. S. K. GUDUNOV, *Mat. Sb.* **47**, 271 (1959).
8. A. HARTEN, *J. Comput. Phys.* **49**, 357 (1983).
9. T. J. R. HUGHES AND M. MALLET, in *Finite Elements in Fluids Vol. 6*, edited by R. H. Gallagher *et al.* (Wiley, New York, 1985).
10. A. JAMESON AND P. D. LAX, Princeton University Report MAE 1650, April 1984 (unpublished).
11. P. D. LAX AND B. WENDROFF, *Commun. Pure Appl. Math.* **13**, 217 (1960).
12. A. Y. LE ROUX, *RAIRO* **15**, 151 (1981).
13. R. J. LEVEQUE, *SIAM J. Numer. Anal.* **22**, 1051 (1985).
14. B. E. MCDONALD AND J. AMBROSIANO, *J. Comput. Phys.* **56**, 448 (1984).

15. K. W. MORTON, in *Proceedings of the Eighth International Conference on Numerical Methods in Fluid Dynamics, Aachen, West Germany, 1982*, edited by E. Krause (Lecture Notes in Physics Vol. 170, Springer-Verlag, Berlin/New York, 1982), pp. 77.
16. K. W. MORTON, in *Proceedings of the Fifth GAMM Conference on Numerical Methods in Fluid Mechanics, Rome, Italy, 1983*, edited by M. Pandolfi and R. Piva (Vieweg, Munich, 1983), p. 243.
17. K. W. MORTON, *Comput. Methods Appl. Mech. Eng.* **52**, 847 (1985).
18. K. W. MORTON AND P. K. SWEBY, in *Proceedings of the Ninth International Conference on Numerical Methods in Fluid Dynamics, Saclay, France, 1984*, edited by Soubbaramayer and J. P. Boujot (Lecture Notes in Physics Vol. 218, Springer-Verlag, New York/Berlin, 1985), p. 412.
19. E. M. MURMAN, *AIAA J.* **12**, 626 (1974).
20. S. OSHER, *SIAM J. Numer. Anal.* **21**, 217 (1984).
21. S. OSHER AND S. SOLOMAN, *Math. Comput.* **38**, 339 (1982).
22. P. L. ROE, *J. Comput. Phys.* **43**, 357 (1981).
23. P. L. ROE, in *Proceedings of AMS/SIAM Summer Seminar, La Jolla, USA, 1983*, edited by B. Engquist *et al.* (Lectures in Applied Mathematics Vol. 22, Amer. Math. Soc., Providence, RI, 1985), p. 163.
24. R. SANDERS, *Math. Comput.* **40**, 91 (1983).
25. J. L. STEGER AND R. F. WARMING, *J. Comput. Phys.* **40**, 263 (1981).
26. P. K. SWEBY, *SIAM J. Numer. Anal.* **21**, 995 (1984).
27. P. K. SWEBY AND M. J. BAINES, *J. Comput. Phys.* **5**, 135 (1984).
28. E. TADMOR, *Math. Comput.* **43**, 369 (1984).
29. B. VAN LEER, *J. Comput. Phys.* **14**, 361 (1974).
30. R. F. WARMING AND R. W. BEAM, *AIAA J.* **14**, 1241 (1976).
31. S. T. ZALESAK, *J. Comput. Phys.* **31**, 335 (1979).