JOURNAL OF COMPUTATIONAL PHYSICS 83, 32–78 (1989)

Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes, II

CHI-WANG SHU*

Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912

AND

STANLEY OSHER†

Department of Mathematics, University of California, Los Angeles, California 90024-1555

Received April 15, 1988; revised August 2, 1988

In this paper we extend our earlier work on the efficient implementation of ENO (essentially non-oscillatory) shock-capturing schemes. We provide a new simplified expression for the ENO construction procedure based again on numerical fluxes rather than cell-averages. We also consider two improvements which we label ENO-LLF (local Lax–Friedrichs) and ENO-Roe, which yield sharper shock transitions, improved overall accuracy, for lower computational cost than previous implementation of the ENO schemes. Two methods of sharpening contact discontinuities—the subcell resolution idea of Harten and the artificial compression idea of Yang, which those authors originally used in the cell average framework—are applied to the current ENO schemes using numerical fluxes and TVD Runge–Kutta time discretizations. The implementation for nonlinear systems and multi-dimensions is given. Finally, many numerical examples, including a compressible shock turbulence interaction flow calculation, are presented. © 1989 Academic Press, Inc.

I. INTRODUCTION

In this paper we extend the construction we began in [12] of efficient implementation of ENO (essentially non-oscillatory) schemes approximating systems of hyperbolic conservation laws of the type:

$$\mathbf{u}_t + \sum_{i=1}^{d} \mathbf{f}_i(\mathbf{u})_{x_i} = 0 \qquad (\text{or} = \mathbf{g}(\mathbf{u}, \mathbf{x}, t), \text{ a forcing term}) \tag{11.1a}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}^0(\mathbf{x}), \tag{1.1b}$$

* Research partically supported by NSF Grant DMS88-10150 and by the Institute for Mathematics and Its Applications.

† Research supported by NSF Grant DMS85-03294, DARPA grant in the ACMP Program, ONR Grant N00014-86-K-0691, NASA Langley Grant NAG-1-270.

0021-9991/89 $3.00

Copyright © 1989 by Academic Press, Inc.
All rights of reproduction in any form reserved.

where $\mathbf{u} = (u_1, \ldots u_m)^T$, $\mathbf{x} = (x^1, x^2, \ldots, x^d)$, and, for real $\xi = (\xi_1, \ldots, \xi_d)$, the combination $\sum_{i=1}^{d} \xi_1 (\partial \mathbf{f}_i / \partial \mathbf{u})$ always has m real eigenvalues and a complete set of eigenvectors. We use the notations $\mathbf{x}_\mathbf{J} = \mathbf{J} \cdot \Delta \mathbf{x}$, $t_n = n\Delta t$, and use $\mathbf{u}_\mathbf{J}^n$ to denote the computed approximation to the exact solution $\mathbf{u}(\mathbf{x}_\mathbf{J}, t)$ of (1.1). We use bold face letters for vectors and plain letters for scalars.

ENO schemes, constructed by Harten, Osher, Engquist, and Chakravarthy [2–5], use a local adaptive stencil to obtain information automatically from regions of smoothness when the solution develops discontinuities. As a result, approximations using these methods can obtain uniformly high-order accuracy right up to discontinuities, while keeping a sharp, essentially non-oscillatory shock transition. The original ENO schemes in [2–5] used a cell-average framework which involved a reconstruction procedure to recover accurate point values from cell averages and a Lax–Wendroff procedure (replacing time derivatives by space derivatives, using the PDE) for the time discretization. This can become a bit complicated for multi-dimensional problems [2]. Harten is currently investigating efficient and more local methods [7]. For ease of implementation we constructed [12] ENO schemes applying the adaptive stencil idea to the numerical fluxes and using TVD Runge–Kutta type time discretizations. The ENO schemes in [12] skip the reconstruction step and the Lax–Wendroff time discretization procedure, hence they are simpler to program, especially for multi-space-dimensional problems. We hope that [12] and this paper will encourage potential users to apply these high-order methods to real physical computations for systems of conservation laws in several space dimensions.

We shall use the same notation as in [12]: $\Delta_\pm$ are the usual difference operators $\Delta_\pm a_j = \pm(a_{j\pm1} - a_j)$; Eq. (1.1a) is sometimes written in an abstract form:

$$\mathbf{u}_t = \mathscr{L}(\mathbf{u}). \tag{1.2}$$

The ENO spatial operator $\mathbf{L}(\mathbf{u})$ is supposed to approximate $\mathscr{L}(\mathbf{u})$ to rth order:

$$\mathbf{L}(\mathbf{u}) = \mathscr{L}(\mathbf{u}) + O(h^r) \tag{1.3}$$

for smooth $\mathbf{u}$, where h is the maximum mesh size, and the Euler forward version

$$\mathbf{w} = \mathbf{T}(\mathbf{u}) = (\mathbf{I} + \Delta t \mathbf{L})(\mathbf{u}) \tag{1.4}$$

is *assumed* to be total variation stable, for scalar, 1D nonlinear problems, under a suitable CFL restriction

$$\lambda = \frac{\Delta t}{\Delta x} \leqslant \lambda_0 \tag{1.5}$$

where λ_0 is usually inversely proportional to $\max |f'(u)|$. At present this stability cannot be proven for unmodified third-order or higher ENO schemes. There is, however, strong theorical and numerical evidence to indicate that the methods are indeed TV stable [3–5, 12].

An rth order TVD Runge–Kutta time discretization is then applied:

$$\mathbf{u}^{(i)} = \sum_{k=0}^{i-1} [\alpha_{ik}\mathbf{u}^{(k)} + \beta_{ik}\Delta t L(\mathbf{u}^{(k)})], \qquad i = 1, ..., \bar{r} \tag{1.6a}$$

$$\mathbf{u}^{(0)} = \mathbf{u}^n, \qquad u^{(\bar{r})} = \mathbf{u}^{n+1} \tag{1.6b}$$

with

$$\mathrm{TV}(\mathbf{u}^{n+1}) \leqslant \mathrm{TV}(T(\mathbf{u}^n)) \tag{1.7a}$$

under the CFL restriction:

$$\lambda = \frac{\Delta t}{\Delta x} \leqslant c_r \cdot \lambda_0. \tag{1.7b}$$

We call c_r the CFL coefficient. We constructed schemes [12] for $r \leqslant 4$ with $\bar{r} = r$ and $c_2 = c_3 = 1$, $c_4 = \frac{2}{3}$. For $r = 5$ we needed $\bar{r} = 6$ and $c_5 = \frac{7}{30}$. For $r \geqslant 4$ we also needed $\tilde{\mathbf{L}}$ which approximates $\mathscr{L}$ in an "adjoint" equation $\mathbf{u}_t = -\mathscr{L}(\mathbf{u})$ because some of the β_{ik} become negative. The details are presented in [12]. Some of the schemes used in Section 5 are listed in Table I.

We shall always use conservative schemes, i.e., for the scalar 1D problem (1.1) we write

$$L(u)_j = -\frac{1}{\Delta x}(\hat{f}_{j+1/2} - \hat{f}_{j-1/2}) \tag{1.8a}$$

for a consistent numerical flux

$$\hat{f}_{j+1/2} = \hat{f}(u_{j-l}, ..., u_{j+k}); \qquad \hat{f}(u, ..., u) = f(u). \tag{1.8b}$$

TABLE I

TVD Runge–Kutta Schemes (1.6)

Order	α_i	β_i	CFL coefficients
2	1 $\frac{1}{2}\ \ \frac{1}{2}$	1 $0\ \ \frac{1}{2}$	1
3	1 $\frac{3}{4}\ \ \frac{1}{4}$ $\frac{1}{3}\ \ 0\ \ \frac{2}{3}$	1 $0\ \ \frac{1}{4}$ $0\ \ 0\ \ \frac{2}{3}$	1
4	1 $\frac{1}{2}\ \ \frac{1}{2}$ $\frac{1}{9}\ \ \frac{2}{9}\ \ \frac{2}{3}$ $0\ \ \frac{1}{3}\ \ \frac{1}{3}\ \ \frac{1}{3}$	$\frac{1}{2}$ $-\frac{1}{4}\ \ \frac{1}{2}$ $-\frac{1}{9}\ \ -\frac{1}{3}\ \ 1$ $0\ \ \frac{1}{6}\ \ 0\ \ \frac{1}{6}$	$\frac{2}{3}$

We proved in [12] the existence of constants $a_2, a_4, \ldots$ such that

$$\hat{f}_{j+1/2} = f_{j+1/2} + \sum_{k=1}^{[(r-1)/2]} a_{2k} \Delta x^{2k} \left(\frac{\partial^{2k}}{\partial x^{2k}} f \right)_{j+1/2} + O(\Delta x^{r+1}) \tag{1.9}$$

guarantees rth order accuracy (1.3). For example, $a_2 = -\frac{1}{24}$, $a_4 = \frac{7}{5760}$, ...

The use of the a_{2k} may seem a bit unnatural. We need to evaluate all the even derivatives of the interpolating polynomials (except the last one if r is even), which increases the computational cost for large r. In Section 2 of this paper we provide a simplified (but equivalent) version of (1.9). We also consider in Section 2 two improvements labelled ENO-LLF (local Lax–Friedrichs) and ENO-Roe, which yield sharper shock transitions, better overall accuracy, and lower computational costs than the methods discribed in [12].

Shocks have a self-sharpening mechanism due to converging characteristics. This is not true for contact discontinuities, which are usually smeared in shock capturing calculations. Recently, Harten [6] introduced the notion of subcell resolution in order to sharpen contact discontinuities. He did this in the context of cell average based ENO schemes. Although in [6] the fact that cell averages contain information leading to the location of the shock is strongly used, this information is actually contained in any *conservative* scheme (1.8). In Section 3 we translate Harten's subcell resolution idea to the ENO schemes using fluxes and Runge–Kutta techniques. The result is simpler than [6] but less accurate in theoretical resolution. However, the numerical results (in Section 5) are quite close to those presented in [6]. The main drawback of our extension of Harten's approach is that it is not clear how to generalize it effectively to several space dimensions. A naive generalization gave unsatisfactory results in our 2D numerical tests. Currently Harten is considering truly 2D subcell resolution methods [7]. Also recently Yang [15] introduced a simple artificial compression technique applied to cell average based ENO schemes. He demonstrated that the technique preserves the TVD (or ENO) properties when applied to a TVD (or ENO) scheme. We translate his result to our present framework in Section 3. The numerical results presented in Section 5 indicate that this adaptation works well both for 1D and 2D problems.

In Section 4 we extend the present implementation to nonlinear systems and to multi-space dimensions. Section 5 contains numerical experiments, including scalar, 1D nonconvex Riemann problems; examples of sharpening contact discontinuities; 2D Riemann problems for Burger's equation; 1D and 2D Euler equations of gas dynamics including a compressible shock-turbulence interaction problem. Concluding remarks are presented in Section 6.

II. Simplifications and Improvements

We begin with the 1D scalar nonlinear problem, i.e., $d = m = 1$ in (1.1). To simplify (1.9) we use the following elementary result:

LEMMA 2.1. *If a function $h(x)$ satisfies*

$$f(u(x)) = \frac{1}{\Delta x} \int_{x-\Delta x/2}^{x+\Delta x/2} h(\xi)\, d\xi, \tag{2.1}$$

then

$$f(u(x))_x = \left(h\left(x + \frac{\Delta x}{2}\right) - h\left(x - \frac{\Delta x}{2}\right) \right) \Big/ \Delta x$$

The proof is trivial.

Lemma 2.1 tells us that the numerical flux $\hat{f}_{j+1/2}$ should approximate $h(x_{j+1/2})$ to a high order. It is not easy to obtain $(h(x)$ directly from (2.1). However, the "reconstruction via primitive function" technique in [4] can be applied to obtain the primitive function of $h(x)$

$$H(x) = \int_{-\infty}^{x} h(\xi)\, d\xi \tag{2.2}$$

at $x_{j+1/2}$ by

$$H(x_{j+1/2}) = \int_{-\infty}^{x_{j+1/2}} h(\xi)\, d\xi = \sum_{k=-\infty}^{j} \int_{x_{k-1/2}}^{x_{k+1/2}} h(\xi)\, d\xi = \Delta x \sum_{k=-\infty}^{j} f(u_k). \tag{2.3}$$

Noticed that the lower limit $-\infty$ is irrelevant. It can be changed to any fixed grid point $x_{j_0+1/2}$.

Given the values of $H(x_{j+1/2})$, we can construct interpolating polynomials in an ENO fashion, i.e., by obtaining a locally "smoothest" stencil starting from one or two points, then adding one point to the stencil at each stage by comparing two divided differences and choosing the one which is smaller in absolute value (see [4]), and then using its derivative at $x_{j+1/2}$ as the numerical flux $\hat{f}_{j+1/2}$. We remark here that by (2.3), we have $(H(x_{j+1/2}) - H(x_{j-1/2}))/\Delta x = f(u_j)$, hence the $(k+1)$th-order divided differences of H can be easily obtained from the kth divided differences of f. Since we shall never need the 0th-order divided differences, we do not perform the summation in (2.3). We simply use the divided difference tables for f.

Lemma 2.1 also explains why the ENO schemes constructed here and in [12] are equivalent (for linear equations in one space dimension only) to the cell-average ENO schemes constructed in [4] via the primitive function for u.

The starting point in the choice of stencil process is very important. "Upwinding" is achieved by this initial choice, and this also is crucial for the evident stability of these methods. We have experimented with using x_{j+1} as the starting point in the choice of stencil process for evaluating $\hat{f}_{j+1/2}$, to solve Burgers' equation (i.e., upwinding in the wrong direction initially) with positive $f'(u)$. We observed instability even if the stencil is chosen in an ENO fashion up to third order. In [12], f is decomposed into $f = f^+ + f^-$ with $(f^+)' \geqslant 0$, $(f^-)' \leqslant 0$, then different

starting points are assigned to $f^{\pm}$ according to the direction of the wind. This simple procedure may smear shocks and affect the overall accuracy (although the smearing is very mild comparing with lower order non-ENO schemes). It was also pointed out in [12], that instead of decomposing f into f^+ and f^- we could also use any E-flux $h_{j+1/2}$ [10], with $df^+_{j+1/2}=f_{j+1}-h_{j+1/2}$, $df^-_{j+1/2}=h_{j+1/2}-f_j$ replacing the first (undivided) differences $f^{\pm}_{j+1}-f^{\pm}_j$; we shall later refer to this procedure as ENO schemes with $h_{j+1/2}$ building blocks. However, that procedure requires smoothness of $h_{j+1/2}$ at sonic points to keep the high accuracy there. It is known that most E-fluxes are *not* smooth at sonic points, they are typically only Lipschitz continuous or perhaps C^1 there. Numerical results do indeed indicate a loss of accuracy at sonic points when such building blocks are used. Although we may modify any E-flux to make it smooth at sonic points, this procedure is too complicated to be computationally pleasing.

We now use a different approach to overcome this difficulty. We choose the first point in the stencil according to the *local* sign of $f'(u)$ at $x_{j+1/2}$. We may use, e.g., the "Roe" speed

$$\tilde{a}_{j+1/2}=\frac{f(u_{j+1})-f(u_j)}{u_{j+1}-u_j} \tag{2.4}$$

to determine the sign of $f'(u_{j+1/2})$. We then have the following algorithm:

ALGORITHM 2.1. (ENO–Roe).

(1) Compute the divided difference table of f, and identify

$$H[x_{l-1/2}, x_{l+1/2}]=f[u(x_l)] \tag{2.5a}$$

$$H[x_{l-1/2}, x_{l+1/2}, \ldots, x_{l+k+1/2}]$$

$$=\frac{1}{k+1}f[u(x_l), \ldots, u(x_{l+k})], \qquad k=1, 2, \ldots, r. \tag{2.5b}$$

(2) If $\tilde{a}_{j+1/2}=(f(u_{j+1})-f(u_j))/u_{j+1}-u_j \geqslant 0$, then

$$k^{(1)}_{\min}=j \tag{2.6a}$$

else

$$k^{(1)}_{\min}=j+1. \tag{2.6b}$$

(3) $Q^{(1)}(x)=H[x_{k^{(1)}_{\min}-1/2}, x_{k^{(1)}_{\min}+1/2}](x-x_{k^{(1)}_{\min}-1/2})$.

(4) Inductively, if $k^{(l-1)}_{\min}$ and $Q^{(l-1)}(x)$ are both defined, then let

$$a^{(l)}=H[x_{k^{(l-1)}_{\min}-1/2}, \ldots, x_{k^{(l-1)}_{\min}+l-1/2}] \tag{2.7a}$$

$$b^{(l)}=H[x_{k^{(l-1)}_{\min}-1-1/2}, \ldots, x_{k^{(l-1)}_{\min}+l-1-1/2}] \tag{2.7b}$$

and

(i) if $|a^{(l)}| \geqslant |b^{(l)}|$, then

$$c^{(l)} = b^{(l)}, \qquad k_{\min}^{(l)} = k_{\min}^{(l-1)} - 1 \tag{2.8a}$$

otherwise

$$c^{(l)} = a^{(l)}, \qquad k_{\min}^{(l)} = k_{\min}^{(l-1)} \tag{2.8b}$$

(ii) form

$$Q^{(l)}(x) = Q^{(l-1)}(x) + c^{(l)} \prod_{k=k_{\min}^{(l-1)}}^{k_{\min}^{(l-1)}+l-1} (x - x_{k-1/2}). \tag{2.9}$$

(5) $Q_{j+1/2}(x) = Q^{(r+1)}(x)$.

(6) We then take

$$\hat{f}_{j+1/2} = \frac{d}{dx} Q_{j+1/2}(x)|_{x=x_{j+1/2}}. \tag{2.10}$$

Remark 2.1. (a) Notice that unlike the ENO scheme which uses the Roe flux as a building block, Algorithm 2.1 gives a uniformly high-order flux $\hat{f}_{j+1/2}$, even at sonic points. The scheme described above will be $(r+1)$th-order accurate except perhaps at isolated zeros of derivatives of the flux $f(u(x))$, where it may degenerate to rth order.

(b) Since $x = x_{j+1/2}$ is a node of $Q_{j+1/2}$ in (2.10), the evaluation of the derivative in (2.10) costs very little.

Algorithm 2.1 yields a scheme admitting a stationary entropy-violating expansion shock since it is based on the first-order Roe scheme. However, the "entropy fix" in this framework turns out to be very simple. Before stating this remedy, let us present another algorithm:

ALGORITHM 2.2 (ENO-LLF).

(1) Compute the divided difference table of f and u, and identify

$$\begin{aligned} &H^{\pm}[x_{l-1/2}, x_{l+1/2}] \\ &\quad = \tfrac{1}{2}(f[u(x_l)] \pm \alpha_{j+1/2} u[x_l]), \\ &\quad\quad l = j-r, \ldots, j+r \text{ for } H^+, \qquad l = j-r+1, \ldots, j+r+1 \text{ for } H^-; \end{aligned} \tag{2.11a}$$

$$\begin{aligned} &H^{\pm}[x_{l-1/2}, x_{l+1/2}, \ldots, x_{l+k+1/2}] \\ &\quad = \frac{1}{k+1} \cdot \frac{1}{2} (f[u(x_l), \ldots, u(x_{l+k})] \pm \alpha_{j+1/2} u[x_l, \ldots, x_{l+k}]) \\ &\quad\quad l = j-r, \ldots, j+r-k \text{ for } H^+, \\ &\quad\quad l = j-r+1, \ldots, j+r-k+1 \text{ for } H^-, k = 1, 2, \ldots, r, \end{aligned} \tag{2.11b}$$

where

$$\alpha_{j+1/2} = \max_{u_j \leqslant u \leqslant u_{j+1}} |f'(u)|. \tag{2.12}$$

(2) For H^+, $k_{\min}^{(1)} = j$, then repeat steps (3)–(4) in Algorithm 2.1 to get $Q_+^{(r+1)}(x)$. Let $Q_{j+1/2}^+(x) = Q_+^{(r+1)}(x)$;

(3) For H^-, $k_{\min}^{(1)} = j+1$, then repeat steps (3)–(4) in Algorithm 2.1 to get $Q_-^{(r+1)}(x)$. Let $Q_{j+1/2}^- = Q_-^{(r+1)}(x)$;

(4) We then take

$$\hat{f}_{j+1/2} = \frac{d}{dx} Q_{j+1/2}^+(x)|_{x=x_{j+1/2}} + \frac{d}{dx} Q_{j+1/2}^-(x)|_{x=x_{j+1/2}}. \tag{2.13}$$

Remark 2.2. (a) When f is convex, $f'' \geqslant 0$, (2.12) simplifies to

$$\alpha_{j+1/2} = \max(|f'(u_j)|, |f'(u_{j+1})|). \tag{2.14}$$

(b) It can be verified that the local Lax–Friedrichs flux

$$h_{j+1/2}^{\mathrm{LLF}} = \tfrac{1}{2}[f(u_j) + f(u_{j+1}) - \alpha_{j+1/2}(u_{j+1} - u_j)], \tag{2.15}$$

where $\alpha_{j+1/2}$ is defined by (2.12), is a monotone flux (see [1] for definitions). We omit the details. However, Algorithm 2.2 is not equivalent to the ENO scheme obtained by using $h_{j+1/2}^{\mathrm{LLF}}$ as a building block [12]. The advantage of Algorithm 2.2 is again its uniformly high order of accuracy.

(c) In terms of cost, Algorithm 2.2 is equivalent to ENO schemes using $f^{\pm}$ in [12] (we shall denote it by ENO-LF), i.e., evaluating two divided difference tables, while Algorithm 2.1 is half as expensive because it only needs one divided difference table.

(d) Roughly speaking the viscosity of ENO-LF is larger than that of ENO-LLF, which is in turn larger than that of ENO-Roe. Hence in going from ENO-LF to ENO-LLF to ENO-Roe we can expect increasingly less shock smearing and better overall accuracy; this is verified by the numerical experiments of Section 5.

(e) As was true for the schemes described in [12], when $r = 1$, Algorithm 2.1 yields a familiar TVD second-order accurate scheme [9], when the minimum of the absolute value function is changed to the minmod function:

$$m(a_1, \ldots, a_n) = \begin{cases} s \cdot \min\limits_{1 \leqslant i \leqslant n} |a_i|, & \text{if } s = \operatorname{sign}(a_1) = \cdots = \operatorname{sign}(a_n) \\ 0, & \text{otherwise.} \end{cases} \tag{2.16}$$

We omit the details of the derivation here.

We now describe the "entropy fix" version of Algorithm 2.1.

ALGORITHM 2.3 (ENO-RF). Let $\hat{f}_{j+1/2}$ be defined by (2.10) if $f'(u)$ does not change sign between u_j and u_{j+1}; otherwise, let $\hat{f}_{j+1/2}$ be defined by (2.13).

Remark 2.3. If f is convex, $f''(u) \geqslant 0$, we can use (2.10) unless $f'(u_j) < 0 < f'(u_{j+1})$; i.e., we use (2.13) only in the cells which contain "expansion shocks." This is the case for, e.g., the Euler equations of fluid gas dynamics. See Section 5.

Notice that the cost of Algorithm 2.3 is very close to that of Algorithm 2.1 because sonic points are isolated and the divided differences of u need only be computed locally near sonic points.

We may associate Algorithm 2.3 with the following E-flux.

$$h^{RF}_{j+1/2} = \begin{cases} f(u_j), & \text{if } f'(u) \geqslant 0 \text{ between } u_j \text{ and } u_{j+1} \\ f(u_{j+1}), & \text{if } f'(u) \leqslant 0 \text{ between } u_j \text{ and } u_{j+1} \\ h^{\mathrm{LLF}}_{j+1/2}, & \text{otherwise;} \end{cases} \tag{2.17}$$

however, as mentioned above, this algorithm is not equivalent to the ENO scheme obtained by using $h^{RF}_{j+1/2}$ as a building block.

Numerically we observe that Algorithm 2.3 always yields the correct entropy solution even for nonconvex f. See Section 5 for some examples.

III. SHARPENING OF CONTACT DISCONTINUITIES

We first consider the 1D, scalar, linear version of (1.1), i.e., $d = m = 1$, $f(u) = au$. Discontinuous solutions to this problem (called contact discontinuities) are usually smeared more severely than shocks, when computed by shock capturing schemes.

Recently Harten [6] introduced the concept of subcell resolution which he used to sharpen contact discontinuities. This notion uses the observation that the cell averages carry information about shock locations. In fact, this information is also contained in any *conservative* scheme (1.8), whether cell average or numerical flux based. We can thus translate Harten's subcell resolution techniques to our ENO schemes which use fluxes and Runge–Kutta time discretizations. Notice that for a linear, constant coefficient problem $f(u) = au$, ENO-LF, ENO-LLF, and ENO-Roe are identical. We write out the following algorithm for the case $a > 0$.

ALGORITHM 3.1. For $a > 0$, at the beginning of every Runge–Kutta cycle:

(1) Let $s_j = |m(\Delta_+ u^n_j, \Delta_- u^n_j)|$, where m is the minmod function (2.16). Define the "critical intervals" (intervals containing discontinuities) $I_j = (x_{j-1/2}, x_{j+1/2})$ by $s_j \geqslant s_{j+1}$, $s_j > s_{j-1}$;

(2) For any "critical interval" I_j, let $\theta_j = (u_j - u_{j-1})/(u_{j+1} - u_{j-1})$ or more accurately let θ_j be the solution in $[0, 1]$ of the quadratic equation

$$(u_{j+1} + u_{j-1} - u_{j+2} - u_{j-2})\theta_j^2 + (3u_{j+2} + 3u_{j-1} - 5u_{j+1} - u_{j-2})\theta_j + (4u_{j+1} - 2u_{j+2} - 2u_j) = 0.$$

We use $x_{j-1/2}+\theta_j \Delta x$ as an approximation to the discontinuity location inside the cell I_j; we then perform each step of the Runge–Kutta cycle as follows:

(3) Let $\hat{f}_{j+1/2}$ be defined as usual (Algorithm 2.1) *unless* I_j or (for the second, third, etc. steps in the Runge–Kutta cycle) I_{j-1} is a "critical interval." If I_j is a "critical interval," we define

$$\hat{f}_{j+1/2}=(1-\xi_j)\,\hat{f}^{(L)}_{j+1/2}+\xi_j\,\hat{f}^{(R)}_{j+1/2} \qquad \text{with} \quad \xi_j=\min\left(\frac{(1-\theta_j)\Delta x}{a\Delta t},\,1\right), \tag{3.1}$$

where $\hat{f}^{(L)}_{j+1/2}$ and $\hat{f}^{(R)}_{j+1/2}$ are again computed by Algorithm 2.1 with the modification on step (2) only, i.e., the first step in the choice of stencil. We use $k^{(1)}_{min}=j-1$ for $\hat{f}^{(L)}_{j+1/2}$ and $k^{(1)}_{\min}=j+1$ for $\hat{f}^{(R)}_{j+1/2}$. To be safe, for the second, third, etc. steps in the Runge–Kutta cycle, we may choose $k^{(1)}_{\min}=j+2$ for $\hat{f}^{(R)}_{j+1/2}$ if $\xi_j<1$, and, when I_{j-1} is a "critical interval" and $\xi_{j-1}<1$, we should choose $k^{(1)}_{\min}=j+1$ to evaluate $\hat{f}_{j+1/2}$.

Remark 3.1 (a) The case $a<0$ is easily obtained by symmetry.

(b) The discontinuity detector in step (1) is somewhat different than that of [6]. The purpose of our choice is to avoid identifying an interval containing a smooth extremum as "critical interval."

(c) As in [6], the modification in (3.1) does not affect accuracy if u happens to be smooth in I_j.

(d) The whole procedure here is simpler than in [6], especially if the first expression for θ_j is used in step (2). The price we pay is that it is less accurate in theoretical resolution, because it is based on piecewise constant (first choice of θ_j) or piecewise linear (second choice of θ_j) interpolants. Numerically, we observe results very close to those found in [6]. See Section 5 for details.

(e) The naive generalization to 2D, namely writing $u+\Delta t L_x u+\Delta t L_y u$ as $\frac{1}{2}(u+2\Delta t L_x u)+\frac{1}{2}(u+2\Delta t L_y u)$ and applying the algorithm separately to the two parts, does not work satisfactorily, according to our numerical tests (see Section 5). A truly 2D subcell resolution seems to be needed. Currently Harten [7] is investigating this within the cell average framework.

(f) For details of the analysis of Algorithm 3.1, we refer the readers to [6]. Although the two algorithms are different, we have essentially borrowed the philosophy of [6] and applied it to the present case.

Another interesting method we can use to sharpen contact discontinuities is an artificial compression method introduced by Yang [15], again in a cell average context, i.e., for MUSCL type schemes. The idea is to sharpen the slopes in the discontinuous cells while still preserving monotonicity. In the MUSCL framework one simply increases the slopes using the jumps in the interpolant across interfaces, while still keeping the ENO property. Translated to our current framework, it gives the following algorithm (again for $a>0$).

ALGORITHM 3.2. Let $\hat{f}^A_{j+1/2} = \hat{f}_{j+1/2} + c_{j+1/2}$, where

$$c_{j+1/2} = m\left[\frac{\alpha}{2} m(\hat{f}^{(R)}_{j+1/2} - \hat{f}_{j+1/2}, \hat{f}^{(R)}_{j-1/2} - \hat{f}_{j-1/2}), f_{j+1} - \hat{f}_{j+1/2}, \hat{f}^{(R)}_{j-1/2} - f_{j-1}\right]. \quad (3.2)$$

Here, as in (3.1), $\hat{f}^{(R)}_{j+1/2}$ corresponds to $k^{(1)}_{\min} = j+1$ in (2.6), step (2) of Algorithm 2.1. m is again the minmod function (2.16).

Remark 3.2. (a) α in (3.2) is a positive parameter. It can be tuned for each individual problem to optimize the results. The bigger the α, the more the compression. Monotonicity is preserved for any α, due to the last two arguments in the outer minmod function in (3.2). We usually use $2 \leqslant \alpha \leqslant 5$, but have not yet found a general rule for determining α.

(b) The approximation for $a < 0$ is constructed symmetrically.

(c) Other versions of (3.2) are possible. For example, we may use

$$\alpha = \alpha_0 \left(\frac{|u_{j+1} - 2u_j + u_{j-1}|}{|u_{j+1} - u_j| + |u_j - u_{j-1}|}\right)^{\beta} \quad (3.3)$$

in (3.2), where $\alpha_0 > 0$ and $\beta = \text{integer} \geqslant 0$ are two parameters. The heuristic reason to use (3.3) is that α is then very small ($\sim O(h^{\beta})$) except near a discontinuity in either u or its derivative, hence over-compression in smooth regions should be avoided. Computationally this improves the performance, especially with some tuning of β. We have not yet been able to find a general rule in choosing the optimal β.

(d) Again, the modification (3.2) does not affect the accuracy in smooth regions.

Both algorithms in this section are easily generalizable to variable coefficient or nonlinear problems. However special caution is needed when one tries to sharpen a (nonlinear) shock, to avoid obtaining a nonphysical, entropy condition violating solution. In the computation for the Euler equations of compressible gas dynamics (Section 5), we only use the compression in the linearly degenerate fields.

IV. IMPLEMENTATIONS IN MULTI-DIMENSIONS AND SYSTEMS

A special advantage of ENO schemes using fluxes and Runge–Kutta methods is their relative simplicity in multi-dimensions. The algorithms in Section 2 are applied to each of the terms $f_i(u)_{x_i}$ in (1.1a), keeping all other variables fixed. The Runge–Kutta method (1.6) is then applied. A typical CFL restriction $(\Delta t/\Delta x) \max_u |f'(u)| \leqslant \lambda_0$ will be replaced by $\Delta t \max_u \sum_{i=1}^{d} (1/\Delta x_i)|f_i'(u)| \leqslant \lambda_0$.

For nonlinear systems, we simply apply the algorithms in Section 2 in each (local) characteristic field. We take an 1D system to exemplify this process. Let $A_{j+1/2}$ be some "average" Jacobian at $x_{j+1/2}$. Examples include $A_{j+1/2} = (\partial \mathbf{f}/\partial \mathbf{u})|_{\mathbf{u}=(1/2)(\mathbf{u}_j+\mathbf{u}_{j+1})}$ or, in the case of Euler equations of gas dynamics $A_{j+1/2} = (\partial \mathbf{f}/\partial \mathbf{u})|_{\mathbf{u}=\mathbf{u}_{j+1/2}^{(\mathrm{Roe})}}$, where $\mathbf{u}_{j+1/2}^{(\mathrm{Roe})} = \mathbf{R}(\mathbf{u}_j, \mathbf{u}_{j+1})$ is the Roe average of $\mathbf{u}_j$ and $\mathbf{u}_{j+1}$ [10]. We denote the eigenvalues and left and right eigenvectors of $A_{j+1/2}$ by $\lambda_{j+1/2}^{(p)}, \mathbf{l}_{j+1/2}^{(p)}, \mathbf{r}_{j+1/2}^{(p)}, p = 1, ..., m$, normalized so that

$$\mathbf{l}_{j+1/2}^{(p)} \cdot \mathbf{r}_{j+1/2}^{(q)} = \delta_{pq} = \begin{cases} 1, & \text{if } p = q \\ 0, & \text{if } p \neq q. \end{cases}$$

For any vector $\mathbf{a}$,

$$a^{(p)} = \mathbf{l}_{j+1/2}^{(p)} \cdot \mathbf{a} \tag{4.1}$$

is the component of $\mathbf{a}$ in the pth (local) characteristic field, because

$$\mathbf{a} = \sum_{p=1}^{m} a^{(p)} \mathbf{r}_{j+1/2}^{(p)}. \tag{4.2}$$

Algorithm 2.1 now becomes:

ALGORITHM 4.1.

(1) Same as step (1) in Algorithm 2.1, changing vectors to bold face letters.

(2) For each j, compute $H^{(p)}[x_{l-1/2}, ..., x_{l+k+1/2}]$ for $l = j-\gamma, ..., j+1$ and $k = 1. 2. ..., r$, by using the divided difference tables of H (derived from those of f) and (4.1). Here $1 \leqslant p \leqslant m$.

(3) Apply steps (2)–(6) of Algorithm 2.1 to $H^{(p)}$, using $\bar{a}_{j+1/2} = \lambda_{j+1/2}^{(p)}$, to get $\hat{f}_{j+1/2}^{(p)}$.

(4) Use (4.2) to get $\hat{f}_{j+1/2}$.

We may similarly generalize Algorithms 2.2 and 2.3. Note that (2.12) becomes, for the pth field,

$$\alpha_{j+1/2}^{(p)} = \max_{\mathbf{u} \in L(\mathbf{u}_j, \mathbf{u}_{j+1})} |\lambda^{(p)}(\mathbf{u})|, \tag{4.3}$$

where $L(\mathbf{u}_j, \mathbf{u}_{j+1})$ is some curve (e.g., a straight line) in phase space connecting $\mathbf{u}_j$ and $\mathbf{u}_{j+1}$. For Euler equations of gas dynamics, the fields are either genuinely nonlinear or linearly degenerate, hence we may use

$$\alpha_{j+1/2}^{(p)} = \max(|\lambda^{(p)}(\mathbf{u}_j)|, |\lambda^{(p)}(\mathbf{u}_{j+1})|), \tag{4.4}$$

which is similar to (2.14).

In terms of computational cost ENO-Roe or ENO-RF is still about half as expensive as ENO-LF, because of the number of divided difference tables needed. The characteristic decompositions (4.1)–(4.2) do involve more vector multiplications for higher orders, e.g., third-order schemes need roughly $\frac{16}{9}$ times more vector multiplications than second-order ones. One way to reduce this cost is to apply the characteristic decomposition only in rapid transition regions. See [15] for an interesting approach to this. We shall not pursue this idea in this paper.

V. Numerical Results

We use the notation ENO $-X-(Y)-r$, where $X=$ Roe, LLF, RF, or LF refers to Algorithms 2.1, 2.2, 2.3 or the $f^{\pm}$ version in [11]; $Y=S$ or A refers to Algorithms 3.1 or 3.2; r is the order of the scheme. For the TVD Runge–Kutta methods used please see Table I. We have run most examples using second-, third-, and fourth-order schemes, but here we usually include only third-order results as representatives.

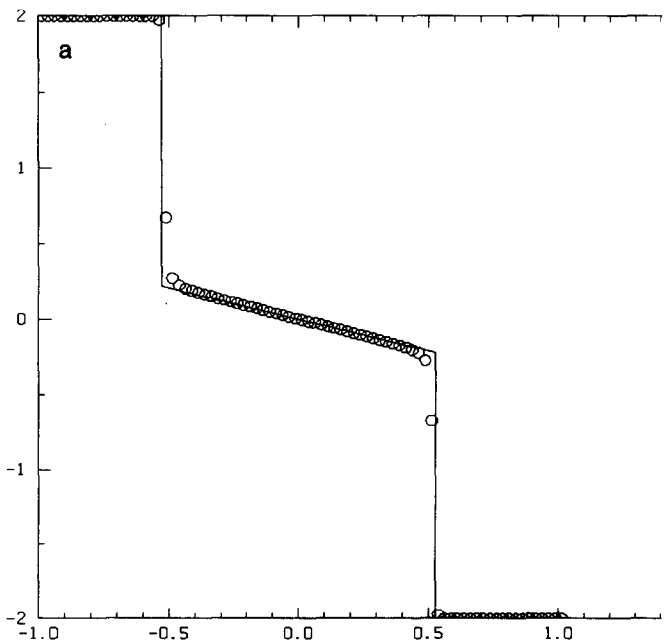


Fig. 1. ENO-RF-3, non-convex fluxes (solid lines are exact solutions; circles are numerical solutions): (a) (5.1) with $u_{\text{left}}=2, u_{\text{right}}=-2$; (b) (5.1) with $u_{\text{left}}=-3, u_{\text{right}}=3$; (c) (5.2) with $u^0(x)=-1$ in $[-\frac{1}{2}, 0]$; $u^0(x)=0$ elsewhere.

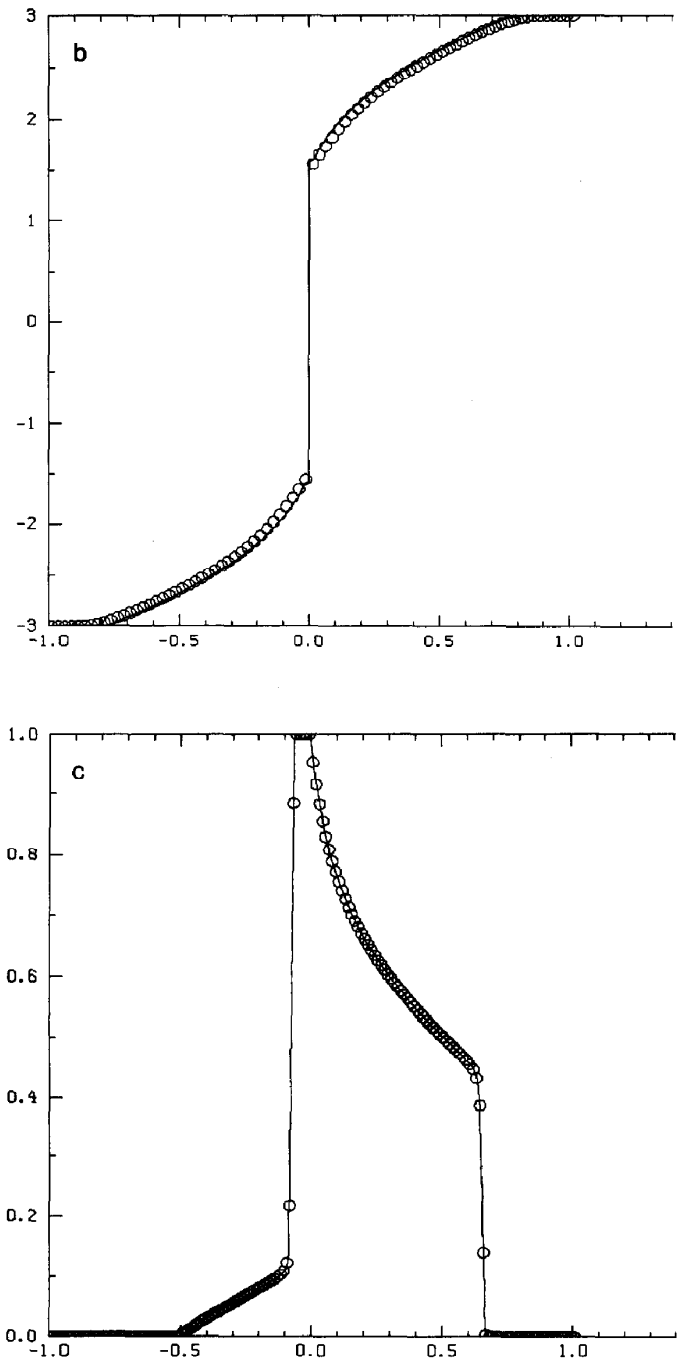


Fig. 1. (*Continued*)

EXAMPLE 1. This is the Example 4 in [12, Section IV] revisited. We solve the scalar 1D Riemann problems of (1.1) with two nonconvex functions f,

$$f(u) = \tfrac{1}{4}(u^2 - 1)(u^2 - 4) \tag{5.1}$$

$$f(u) = \frac{4u^2}{4u^2 + (1-u)^2}. \tag{5.2}$$

ENO-RF-3 is used. The results are displayed in Fig. 1. We observe considerable improvements in shock transition and overall resolution compared to the results shown in Fig. 14, 18, and 22 in [12]. ENO-RF-3 and ENO-RF-4 were also tested on several other nonconvex problems. We always observed convergence to the correct entropy solution and a well-resolved solution. When applied to Burgers' equation (Example 1 in [12, Section IV]) we observed improvements in accuracy and shock transition of ENO-RF over ENO-LF. We omit the details here.

EXAMPLE 2. This example uses the two contact-discontinuity-sharpening algorithms in Section 3 applied to scalar 1D linear problems. We solve the model equation

$$u_t + u_x = 0, \qquad -1 \leqslant x < 1 \tag{5.3a}$$

$$u(x, 0) = u^0(x), \qquad u^0(x) \text{ periodic with period 2.} \tag{5.3b}$$

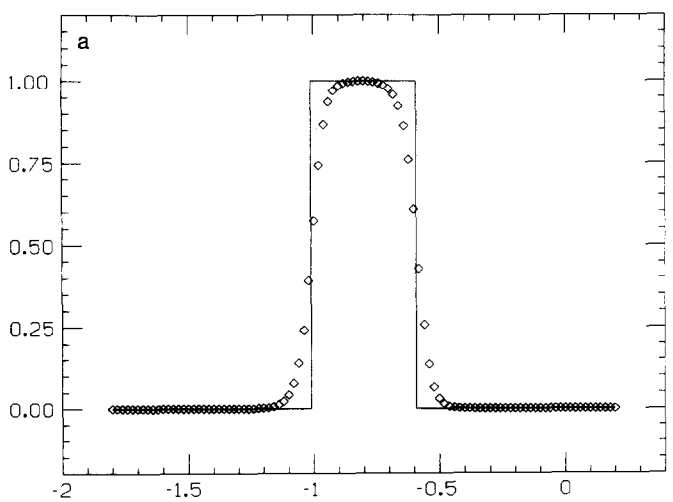


FIG. 2. (5.3)–(5.4a) (solid lines are exact solutions; diamonds are numerical solutions): (a) ENO-Roe-3; (b) ENO-Roe-S-3; (c) ENO-Roe-A-3.

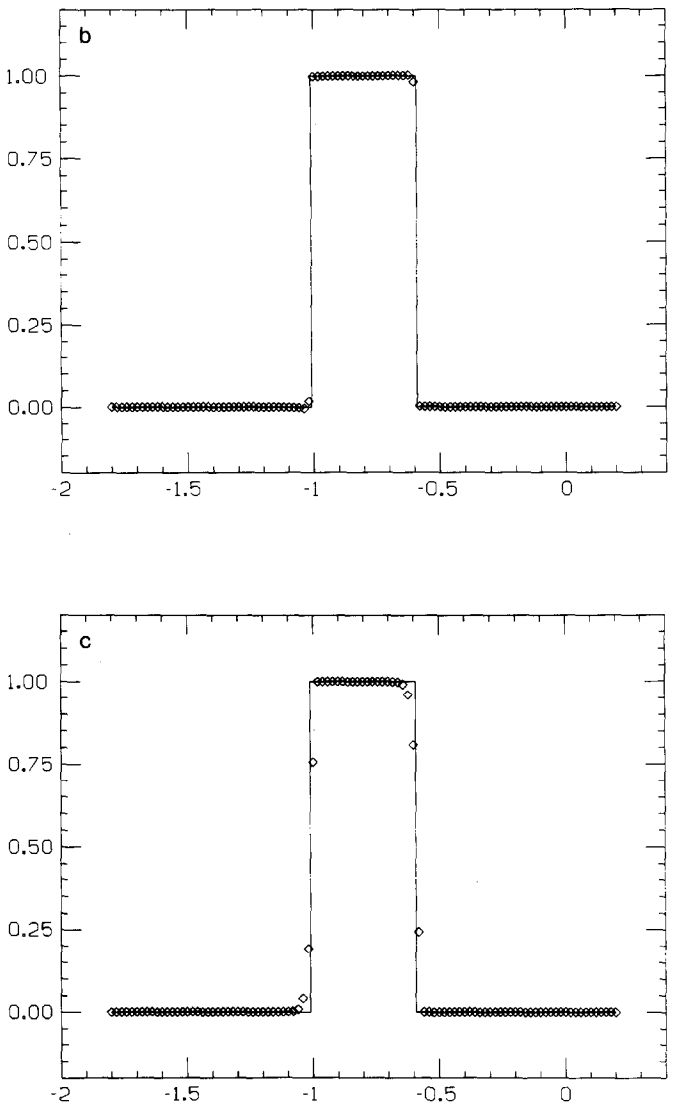


Fig. 2. (*Continued*)

Notice that in this linear case all the three algorithms ENO-Roe, ENO-LLF, and ENO-RF in Section 2 are identical.

Four initial conditions $u^0(x)$ are used. The first three are used by Zalesak [16], using 100 equally spaced grid points in $[-1, 1)$ with

$$u^0(x) = \begin{cases} 1, & -\frac{1}{5} \leqslant x \leqslant \frac{1}{5}, \\ 0, & \text{otherwise}, \end{cases} \quad 800 \text{ time steps, CFL} = 0.2 \tag{5.4a}$$

$$u^0(x) = e^{-300x^2}, \quad 600 \text{ time steps, CFL} = 0.1 \tag{5.4b}$$

$$u^0(x) = \begin{cases} (1 - (\frac{10}{3}x)^2)^{1/2}, & |x| < \frac{3}{10}, \\ 0, & \text{otherwise}, \end{cases} \quad 600 \text{ time steps, CFL} = 0.1 \tag{5.4c}$$

(since $u^0(x)$ is periodic we only give the definitions in $[-1, 1)$). The last is used by Harten *et al.* [4, 6]:

$$u^0(x+0.5) = \begin{cases} -x \sin(\frac{3}{2}\pi x^2), & -1 \leqslant x < -\frac{1}{3}, \\ |\sin(2\pi x)|, & |x| < \frac{1}{3}, \quad t = 8 \\ 2x = 1 - \sin(3\pi x)/6, & \frac{1}{3} < x < 1, \end{cases} \tag{5.4d}$$

The results are displayed in Fig. 2–5. We observe apparent improvements of both algorithms in all cases. For comparisons we refer the readers to [6, 16].

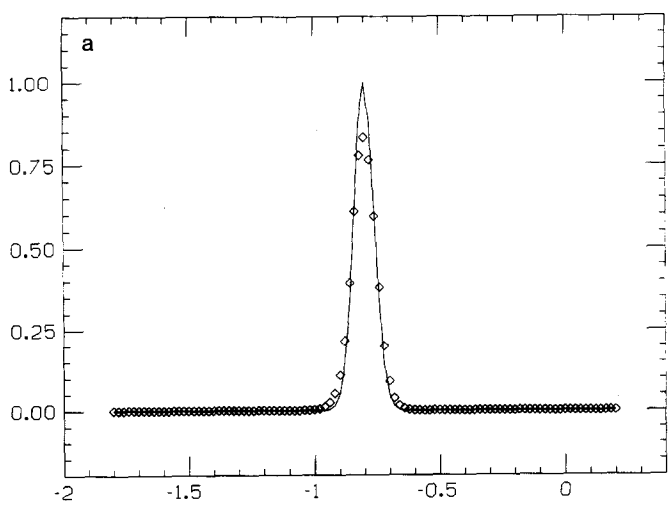
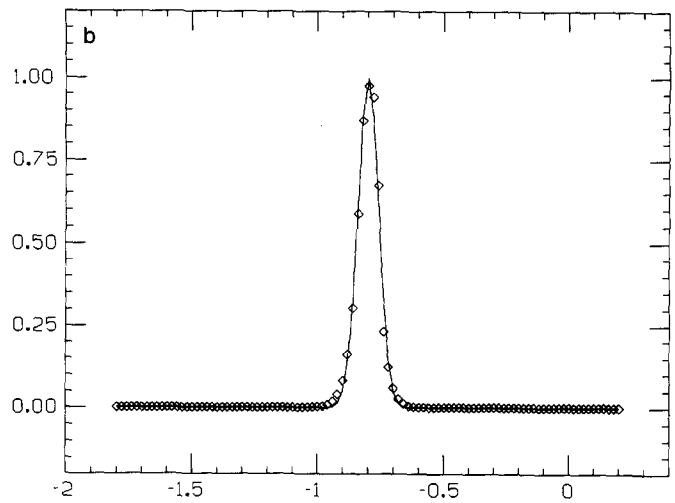


FIG. 3. (5.3)–(5.4b) (solid lines are exact solutions; diamonds are numerical solutions): (a) ENO-Roe-3; (b) ENO-Roe-S-3; (c) ENO-Roe-A-3.



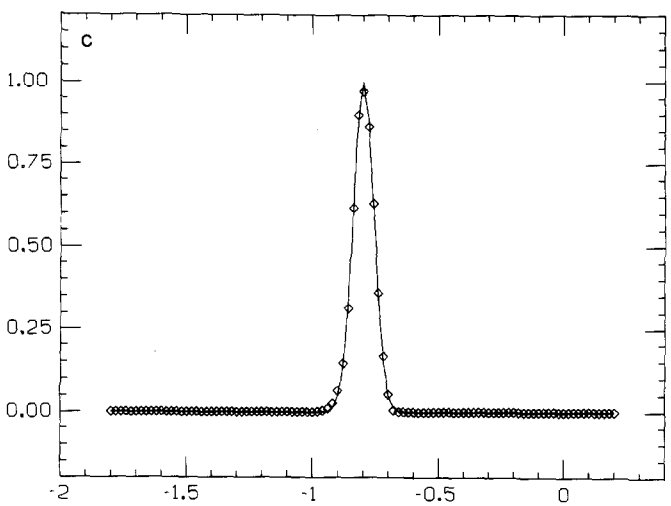


Fig. 3. (*Continued*)

EXAMPLE 3. We solve a Riemann problem for the 2D Burgers' equation

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0 \tag{5.5a}$$

of the type:

$$u(x, y, 0) = \begin{cases} u_1, & x > 0, y > 0 \\ u_2, & x < 0, y > 0 \\ u_3, & x < 0, y < 0 \\ u_4, & x > 0, y < 0. \end{cases} \tag{5.5b}$$

Depending on the orders of the u_i's, there are eight essentially different solution types. See [13] for details. We used ENO-RF and observed convergence to the correct entropy solution with good resolution for all cases. The results are displayed in Fig. 6–7. For one case (Fig. 6) we present the results of ENO-RF-3 for 20×20 and 80×80 grid points, and the result using first-order Engquist–Osher scheme for 80×80 grid points. We observe that ENO-RF-3 using 20×20 points has roughly the same resolution as does the first-order montone scheme with 80×80 points. For the remaining seven cases (Fig. 7) we only show the results of ENO-RF-3 with 80×80 points.

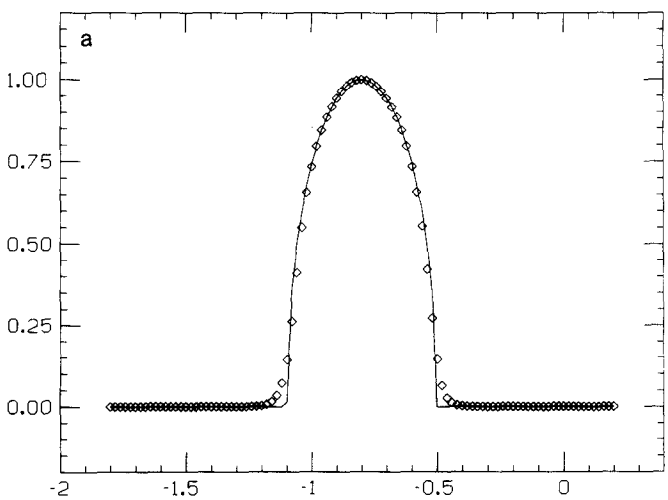


FIG. 4. (5.3)–(5.4c) (solid lines are exact solutions; diamonds are numerical solutions): (a) ENO-Roe-3; (b) ENO-Roe-S-3; (c) ENO-Roe-A-3.

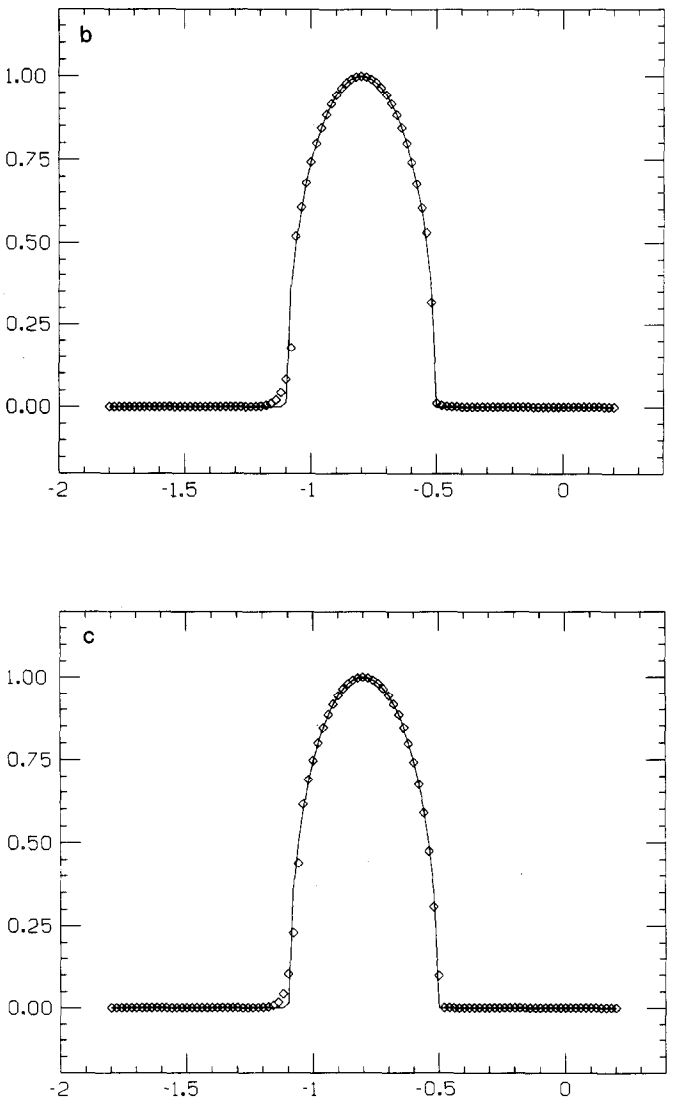


Fig. 4. (*Continued*)

We also remark here that Yang's artificial compression Algorithm 3.2 improves the resolution in this example according to our numerical experiments. Since we are mainly interested in applying Algorithm 3.1 and 3.2 to linear problems, we omit the details here.

EXAMPLE 4. This example is suggested by Professor D. Gottlieb (private communication). It simulates a 2D boundary layer problem. We solve the boundary value problem

$$u_t + \left(\frac{u^2}{2}\right)_x + u_y = 0, \qquad 0 \leqslant x < 2\pi, \quad 0 \leqslant y \leqslant 1 \tag{5.6a}$$

$$u(x, 0, t) = \alpha + \beta \sin x, \qquad u \text{ is periodic in } x \text{ with period } 2\pi \tag{5.6b}$$

to steady state. In steady state it resembles the 1D Burger's equation with $\alpha + \beta \sin x$ as "initial" condition, if y is identified as time t. Hence the exact solution to the steady state can be obtained. This experiment simulates a 2D boundary layer problem because the shock "dissolves" near the boundary $y = 0$. We can adjust the thickness of the boundary layer (i.e., the smooth region near $y = 0$) by adjusting β. We used ENO-RF-3 to compute (5.6), running it to steady state, with initial condition $u(x, y, 0) = \alpha + \beta \sin x$. We imposed periodic boundary conditions

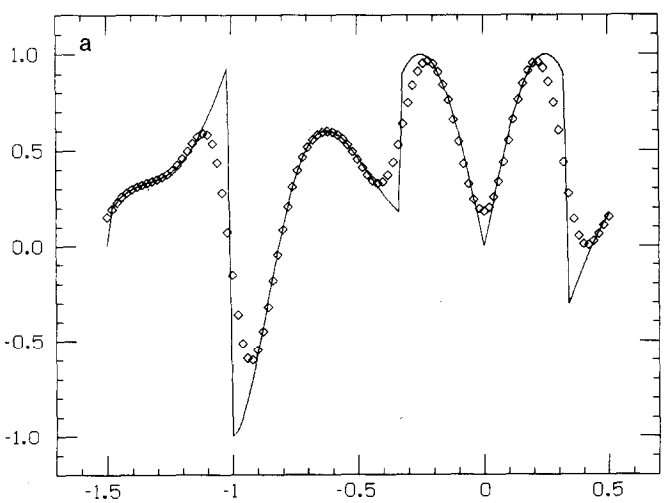


FIG. 5. (5.3)–(5.4d) (solid lines are exact solutions; diamonds are numerical solutions): (a) ENO-Roe-3; (b) ENO-Roe-S-3; (c) ENO-Roe-A-3.

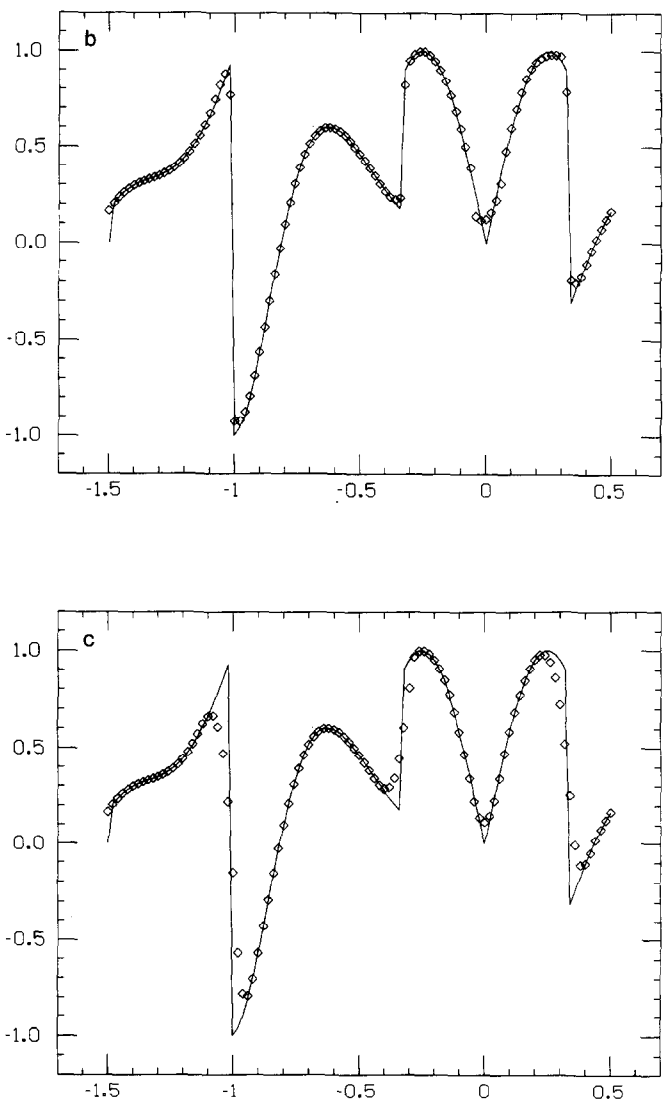


Fig. 5. (*Continued*)

in x, enforced (5.6b) at $y=0$, and imposed no boundary condition at the outflow boundary $y=1$. (For a detailed discussion of the implementation of boundary conditions for ENO schemes we refer the readers to [4].) Figure 8 contains the level curves for $\alpha=0$, $\beta=5$ and $\alpha=2$, $\beta=5$. 30×30 spatial points are used.

EXAMPLE 5. This example was suggested by Professor R. Sanders (private communication), to study the smearing of contact discontinuities in different directions when $\Delta x \neq \Delta y$. We solve

$$u_t + u_x + u_y = 0, \qquad u \text{ is periodic in } x \text{ and } y \text{ with periods } 1 \tag{5.7a}$$

$$u(x, y, 0) = \begin{cases} 1, & \text{if } (x-\frac{1}{2})^2 + (y-\frac{1}{2})^2 \leqslant \frac{1}{8} \\ 0, & \text{otherwise in } 0 \leqslant x, y < 1 \end{cases} \tag{5.7b}$$

with $\Delta y = 2\Delta x = \frac{1}{20}$, and ran it for $t=2$ (two periods). In Fig. 9 the "+" signs are for the cross section $y=\frac{1}{2}$, and the "*" signs are for the cross section $x=\frac{1}{2}$. We observe that ENO-Roe-3 does smear differently in the x and y directions with $\Delta x \neq \Delta y$, but ENO-Roe-A-3 does a much better job. On the other hand, ENO-

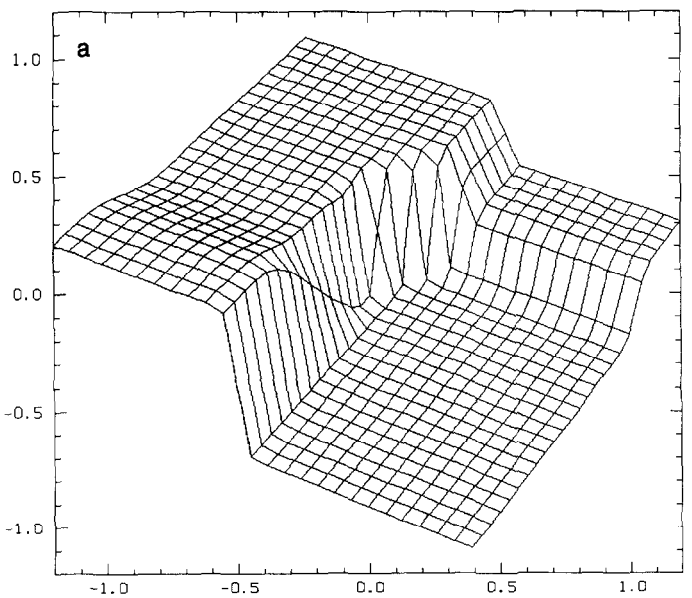


FIG. 6. (5.5 with $(u_1, u_2, u_3, u_4) = (-1, -0.2, 0.5, 0.8)$, $t=1$ (surfaces of numerical solutions): (a) ENO-RF-3, 20×20 points; (b) ENO-RF-3, 80×80 points; (c) First-order Engquist–Osher scheme, 80×80 points.

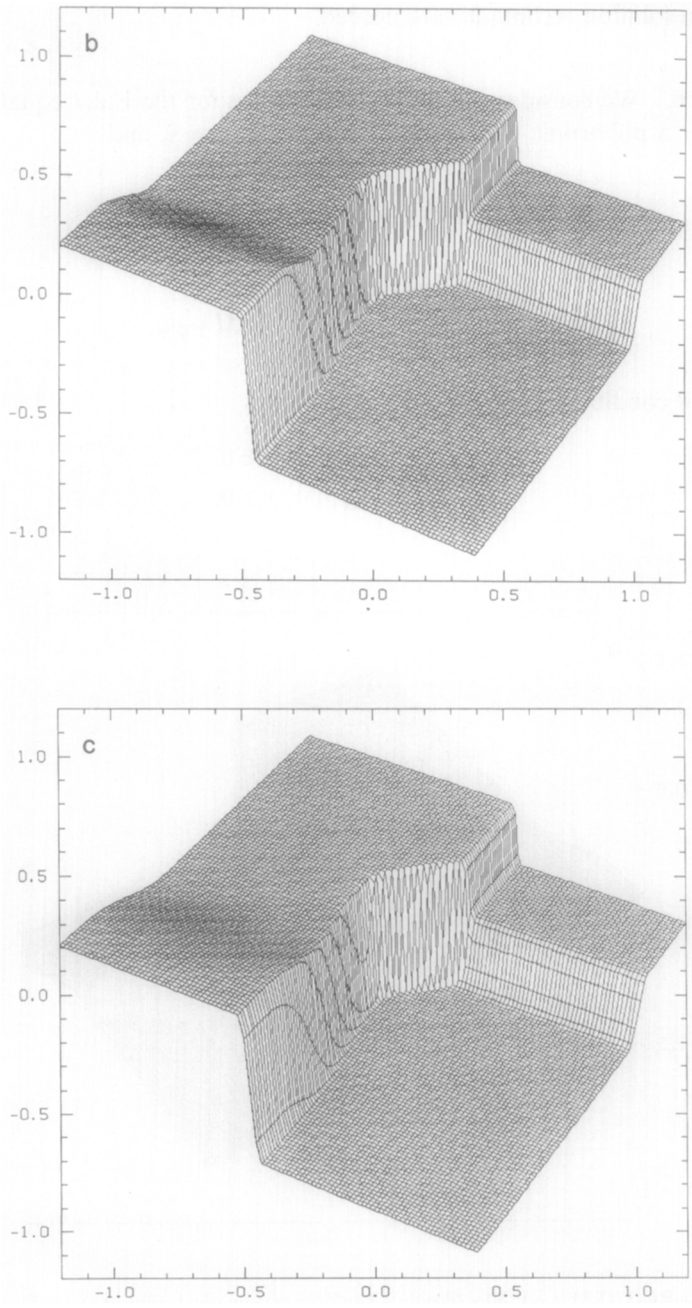


Fig. 6. (*Continued*)

Roe-S-2 (naive generalization to 2D) does not work well. It seems that some truly 2D subcell resolution techniques are needed.

EXAMPLE 6. We consider the Riemann problems for the Euler equations of gas dynamics for a polytropic gas, i.e., (1.1) with $d=1$, $m=3$, and

$$\mathbf{u}=(p, M, E)^{\mathrm{T}}, \qquad \mathbf{f}(\mathbf{u})=q\mathbf{u}+(0, P, qP)^{\mathrm{T}}, \tag{5.8a}$$

where

$$P=(\gamma-1)(E-\tfrac{1}{2}\rho q^2), \qquad M=\rho q \tag{5.8b}$$

for the initial condition

$$\mathbf{u}(x, 0)=\begin{cases}\mathbf{u}_{\mathrm{L}}, & x<0\\ \mathbf{u}_{\mathrm{R}}, & x\geqslant 0.\end{cases} \tag{5.8c}$$

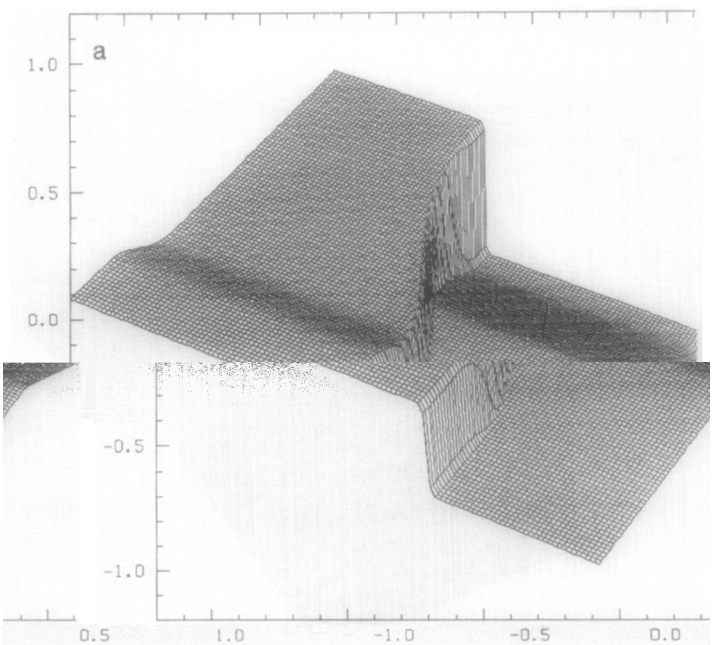


FIG. 7. ENO-RF-3, (5.5), $t=1$, 80×80 points (surfaces of numerical solutions): (a) $(u_1, u_2, u_3, u_4)=(-0.2, -1, 0.5, 0.8)$; (b) $(u_1, u_2, u_3, u_4)=(-1, 0.5, -0.2, 0.8)$; (c) $(u_1, u_2, u_3, u_4)=(-1, 0.5, -0.2, 0.8)$; (d) $(u_1, u_2, u_3, u_4)=(-1, -0.2, 0.8, 0.5)$; (e) $(u_1, u_2, u_3, u_4)=(0.8, -1, 0.2, -0.5)$; (f) $(u_1, u_2, u_3, u_4)=(0.8, -1, 0.5, -0.2)$; (g) $(u_1, u_2, u_3, u_4)=(0.8, -0.2, -1, 0.5)$.

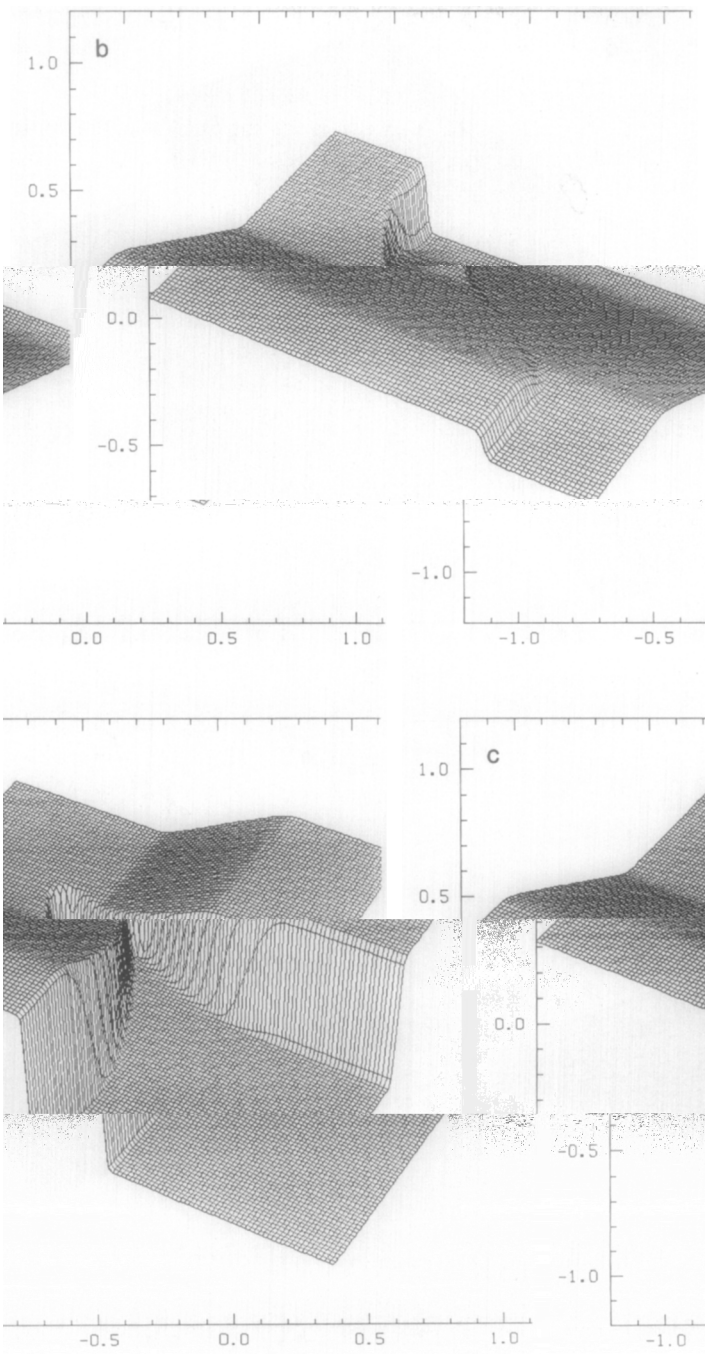


Fig. 7. (*Continued*)

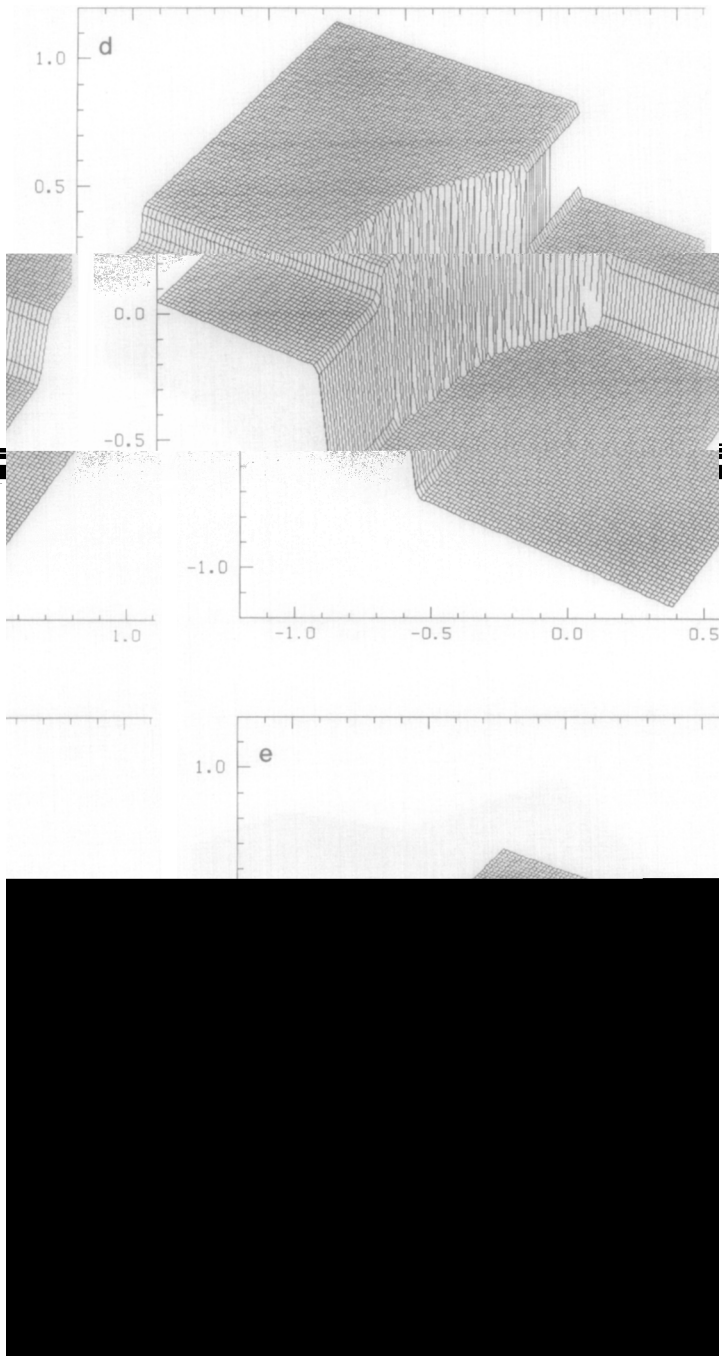


Fig. 7. (*Continued*)

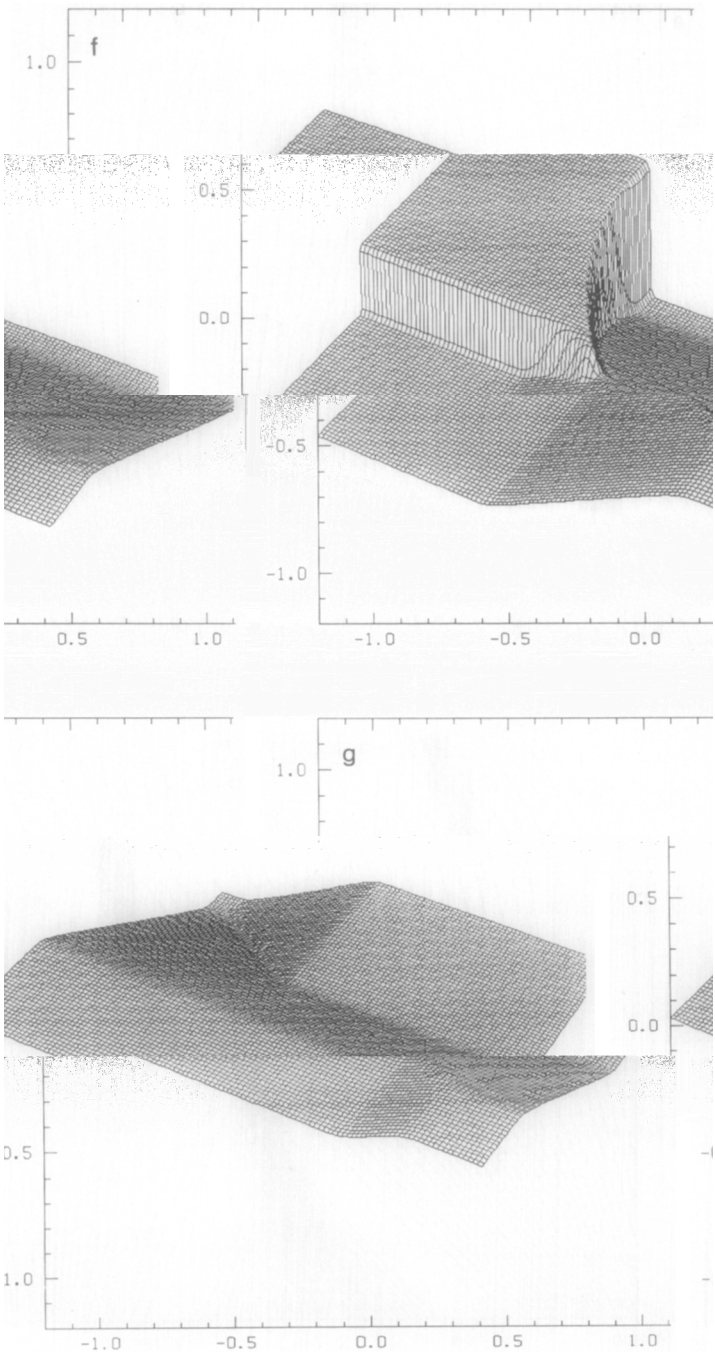


Fig. 7. (*Continued*)

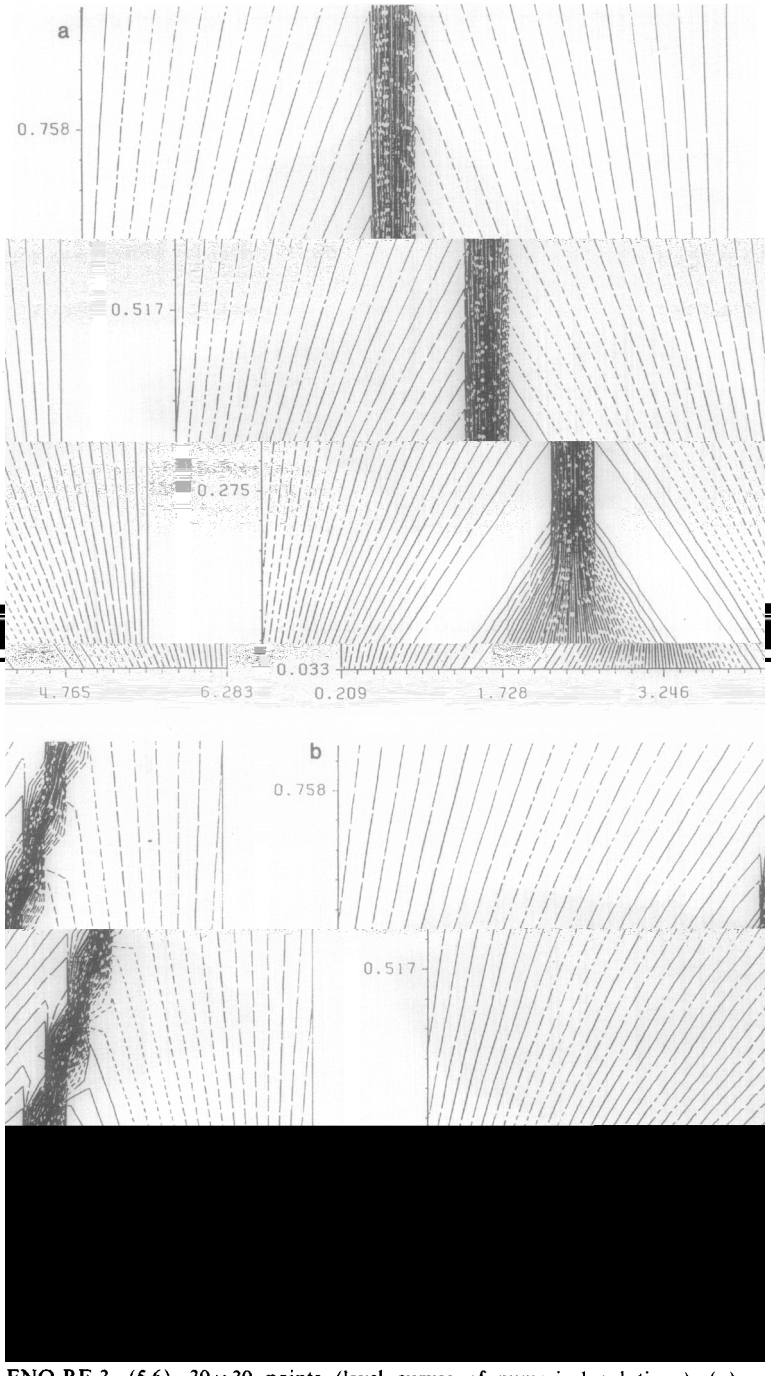


FIG. 8. ENO-RF-3, (5.6), 30×30 points (level curves of numerical solutions): (a) $\alpha = 0, \beta = 5$; (b) $\alpha = 2, \beta = 5$.

$\gamma = 1.4$ is used. For details of the Jacobian, its eigenvalues, and eigenvectors, see [4, 11] or Example 9 below.

We use the same two sets of initial conditions as in [4]:

$$(\rho_L, q_L, P_L) = (1, 0, 1); \qquad (\rho_R, q_R, P_R) = 0.125, 0, 0.10) \tag{5.9a}$$

$$(\rho_L, q_L, P_L) = (0.445, 0.698, 3.528); \qquad (\rho_R, q_R, P_R) = (0.5, 0, 0.571). \tag{5.9b}$$

The results are in Fig. 10–11. Notice the improved treatment of contact discontinuities after using the two contact sharpening agorithms of Section 3. Also notice that the corners of rarefaction waves (discontinuities in derivatives) are not resolved as well. Currently this phenomenon is still under investigation.

EXAMPLE 7. This is the same equation as in Example 6 with the initial condition

$$\mathbf{u}(x, 0) = \begin{cases} \mathbf{u}_L, & 0 \leqslant x < 0.1 \\ \mathbf{u}_M, & 0.1 \leqslant x < 0.9 \\ \mathbf{u}_R, & 0.9 \leqslant x < 1, \end{cases} \tag{5.10a}$$

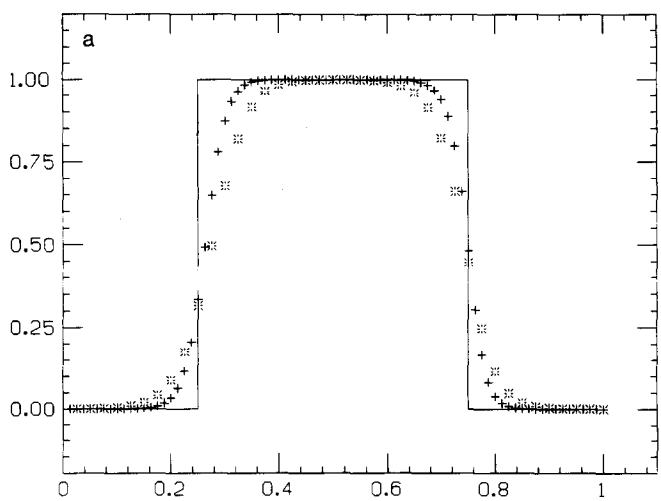
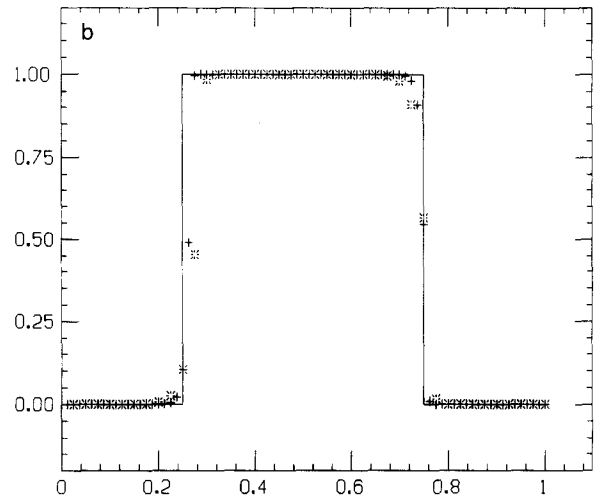


FIG. 9. (5.7), "+" for $y = \frac{1}{2}$, "*" for $x = \frac{1}{2}$; $t = 2$ (solid lines are exact solutions; crosses are numerical solutions): (a) ENO-Roe-3; (b) ENO-Roe-A-3; (c) ENO-Roe-S-2.



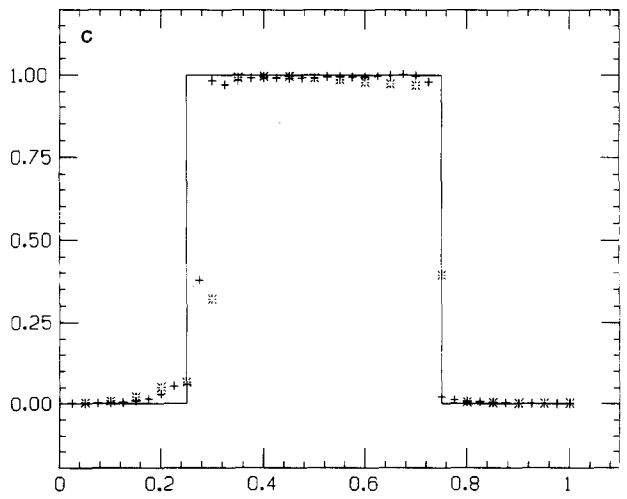


Fig. 9. (*Continued*)

where

$$\rho_L = \rho_M = \rho_R = 1, \qquad q_L = q_M = q_R = 0, \qquad P_L = 10^3, \qquad P_M = 10^{-2}, \qquad P_R = 10^2. \tag{5.10b}$$

A solid wall boundary condition is applied to both ends. See [4, 14] for details.

The results at the final time $t = 0.038$ are in Fig. 12–13. The solid lines are the numerical solutions using ENO-RF-S-3 with 800 points. It can be regarded as an exact solution. Notice that ENO-RF-3 has essentially converged for the velocity and the pressure, but not for the density, due to the smearing of contact discontinuities (Fig. 12). Also notice that ENO-RF-S-3 and ENO-RF-A-3 with 200 points have better resolution for the density than does ENO-RF-3 with 400 points (Fig. 13).

EXAMPLE 8. In the above problems we only show pictures for third-order schemes. If we compare with second- and fourth-order schemes we see the expected improvements in resolution with higher orders. These improvements are usually not very significant for problems lacking structure in smooth regions, as in the case for most of the above examples. This is particularly true when the two contact-discontinuity-sharpening algorithms in Section 3 are used. However, the advantage of higher order schemes becomes significant, when the problem involved has some structure. To exemplify this, we solve the Euler's equation in

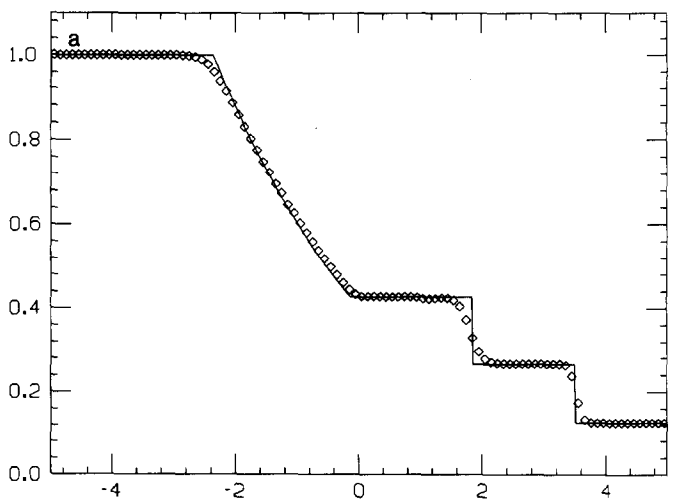


FIG. 10. ENO-RF-3, (5.8)–(5.9a), 100 points, $t = 2$ (solid lines are exact solutions; diamonds are numerical solutions): (a) density; (b) velocity; (c) pressure.

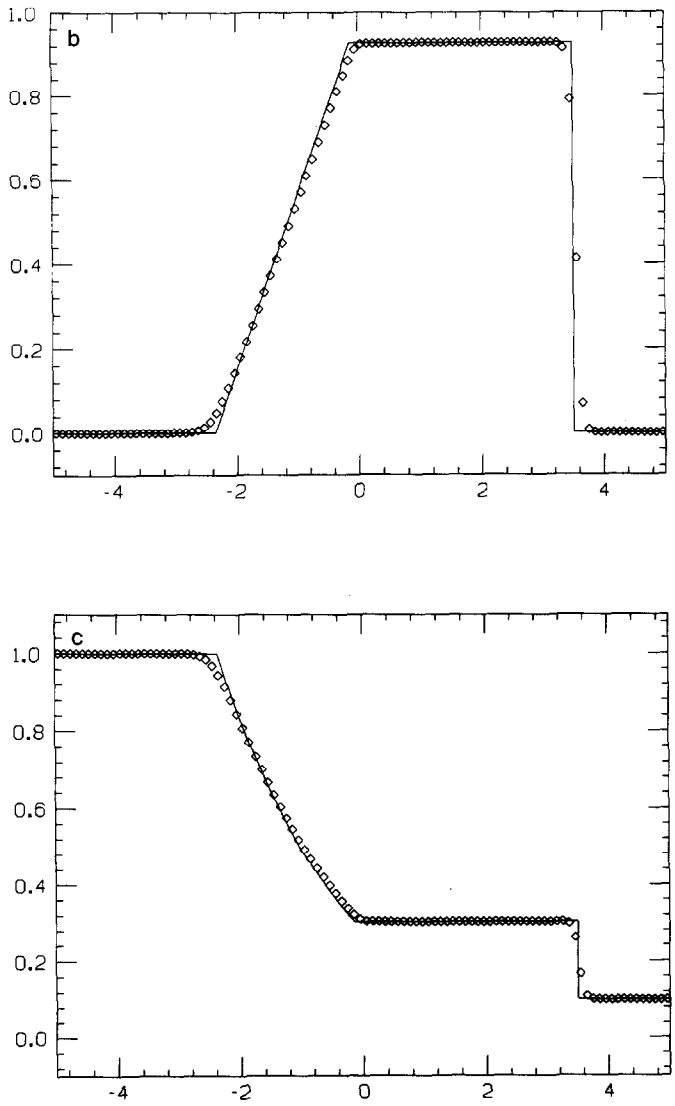


Fig. 10. (*Continued*)

Example 6 with a moving Mach = 3 shock interacting with sine waves in density, i.e., initially

$$\begin{aligned} &\rho = 3.857143; \quad q = 2.629369; \quad P = 10.33333 \quad \text{when } x < -4 \\ &\rho = 1 + \varepsilon \sin 5x; \quad q = 0; \quad P = 1 \quad \text{when } x \geqslant -4. \end{aligned} \tag{5.11}$$

If $\varepsilon = 0$, this is a pure Mach = 3 shock moving to the right.

We take $\varepsilon = 0.2$. For a linearized analysis of this problem see [8]. The results are in Fig. 14–15. The solid lines are numerical solutions of ENO-RF-3 with 1600 grid points. It can be regarded as the exact solution. We observe that the fine structure in the density profile makes the higher order schemes perform much better than the lower order methods. ENO-RF-3 with 200 points has roughly the same resolution as the second-order MUSCL type TVD scheme [9] with 800 points (Fig. 14). On the other hand, the improvement of ENO-RF-3 over the second-order TVD scheme is not so significant for the velocity and pressure profiles, because they both lack any detailed structure (Fig. 15).

EXAMPLE 9. We apply ENO schemes to 2D Euler's equation of gas dynamics, i.e., (1.1) with $d = 2$, $m = 4$, and (we use $\mathbf{f}$, $\mathbf{g}$, x, y instead of $\mathbf{f}_1$, $\mathbf{f}_2$, x_1, x_2):

$$\mathbf{u} = (\rho, M_x, M_y, \mathrm{E})^{\mathrm{T}}, \qquad \mathbf{f}(\mathbf{u}) = q_x \mathbf{u} + (0, P, 0, q_x P), \qquad \mathbf{g}(\mathbf{u}) = q_y \mathbf{u} + (0, 0, P, q_y P), \tag{5.12a}$$

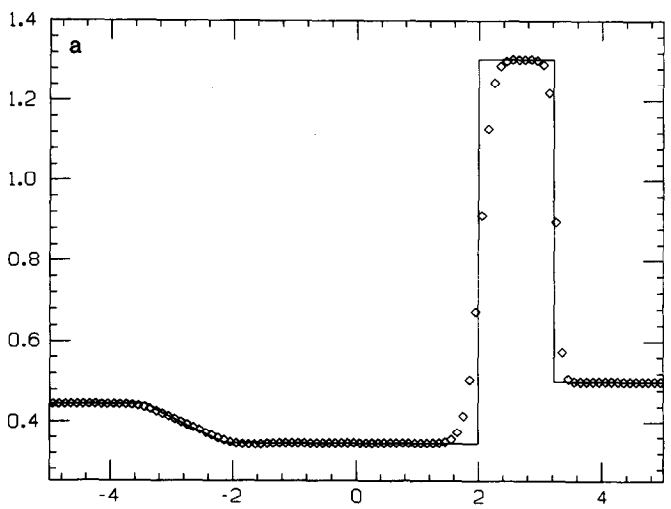


FIG. 11. (5.8)–(5.9b), 100 points, $t = 1.3$, density (solid lines are exact solutions; diamonds are numerical solutions): (a) ENO-RF-3; (b) ENO-RF-S-3; (c) ENO-RF-A-3.

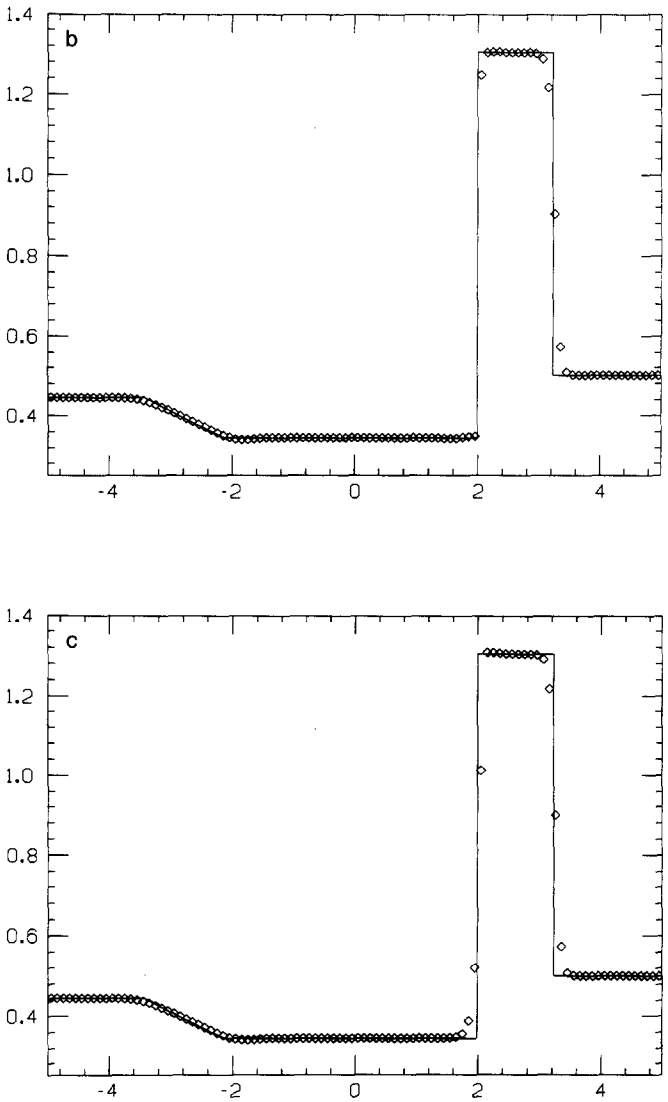


Fig. 11. (*Continued*)

where

$$P=(\gamma-1)(E-\tfrac{1}{2}\rho q^2), \qquad q^2=q_x^2+q_y^2,\ M_x=\rho q_x,\ M_y=\rho q_y. \tag{5.12b}$$

Also

$$c^2=(\gamma-1)(H-\tfrac{1}{2}q^2), \qquad H=\frac{E+P}{\rho}. \tag{5.12c}$$

For $\partial\mathbf{f}/\partial\mathbf{u}$, the eigenvalues are

$$\gamma_1=q_x-c, \qquad \lambda_2=\lambda_3=q_x, \qquad \lambda_4=q_x+c, \tag{5.13a}$$

the right eigenvectors are

$$\mathbf{r}_1=\begin{pmatrix}1\\ q_x-c\\ q_y\\ H-q_xc\end{pmatrix}, \quad \mathbf{r}_2=\begin{pmatrix}0\\ 0\\ 1\\ q_y\end{pmatrix}, \quad \mathbf{r}_3=\begin{pmatrix}1\\ q_x\\ q_y\\ \frac{1}{2}q^2\end{pmatrix}, \quad \mathbf{r}_4=\begin{pmatrix}1\\ q_x+c\\ q_y\\ H+q_xc\end{pmatrix} \tag{5.13b}$$

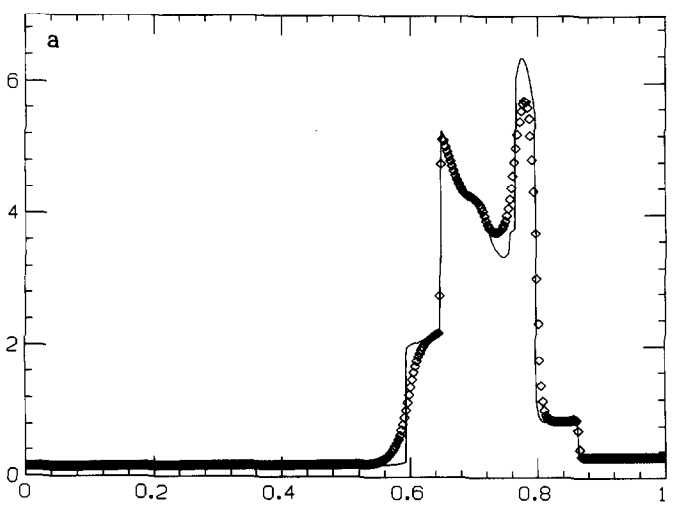


FIG. 12. ENO-RF-3, (5.8a)–(5.10), 400 points, $t=0.038$ (solid lines are the numerical solution of ENO-RF-3, 800 points): (a) density (diamonds); (b) velocity (diamonds); (c) pressure (diamonds).

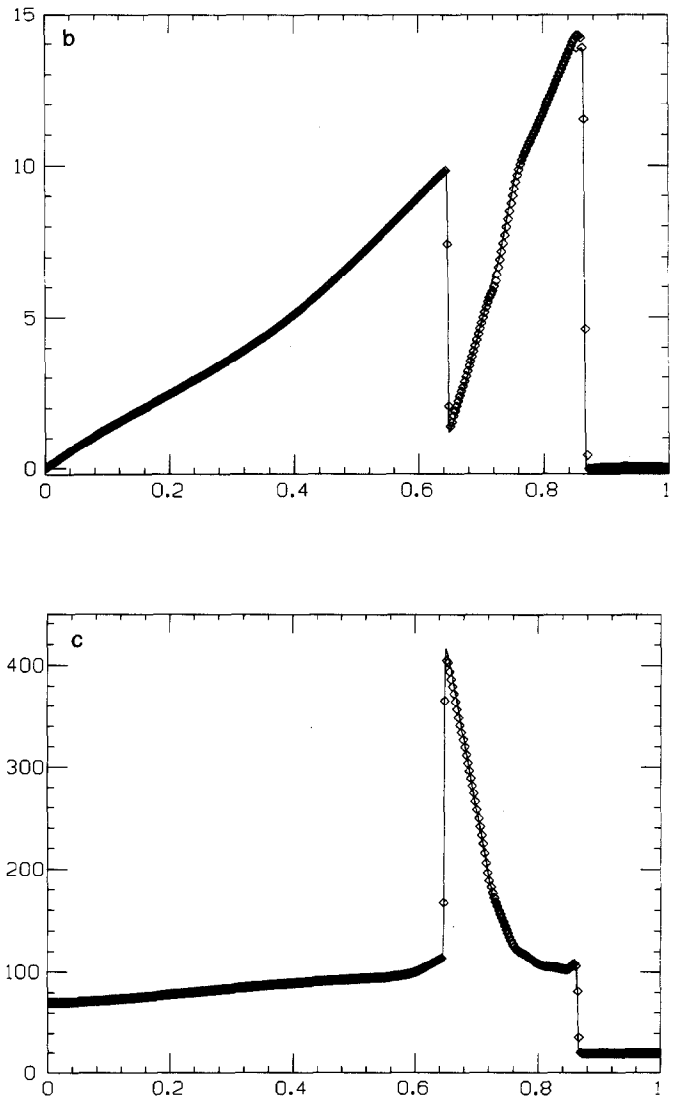


Fig. 12. (*Continued*)

and the left eigenvectors are

$$l_1 = \frac{1}{2}\left(b_2 + \frac{q_x}{2}, -\frac{1}{c} - b_1 q_x, -b_1 q_y, b_1\right),$$
$$l_2 = (-q_y, 0, 1, 0), \qquad l_3 = (1 - b_2, b_1 q_x, b_1 q_y, -b_1), \tag{5.13c}$$
$$l_4 = \frac{1}{2}\left(b_2 - \frac{q_x}{c}, \frac{1}{c} - b_1 q_x, -b_1 q_y, b_1\right),$$

where

$$b_1 = (\gamma - 1)/c^2 \tag{5.13d}$$
$$b_2 = \tfrac{1}{2} q^2 b_1. \tag{5.13e}$$

We can get the results for $\partial \mathbf{g}/\partial \mathbf{u}$ by symmetry.

The test problem we choose is a moving shock interacting with compressible turbulence [17, 18]. At $t = 0$, a Mach 8 shock at $x = -1.0$ is moving to the right into a state with $P_R = 1, \rho_R = 1$, and $q_x = -(c_R/P_R)\sin\theta_R \cos(xk_R \cos\theta_R + yk_R \sin\theta_R)$, $q_y = (c_R/P_R)\cos\theta_R \cos(xk_R \cos\theta_R + yk_R \sin\theta_R)$, where $k_R = 2\pi$, $\theta_R = \pi/6$. We display the results at $t = 0.20$ in Fig. 156. Notice that in [17, 18] similar results were obtained using a shock-fitting rather than a shock capturing method.

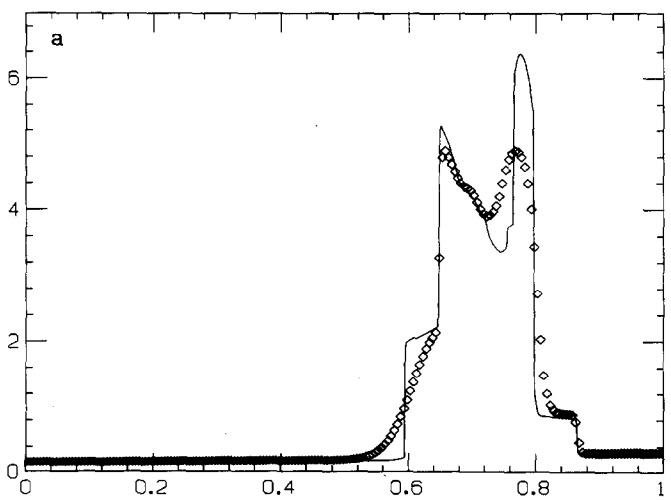


FIG. 13. (5.8a)–(5.10), 200 points, $t = 0.038$, density (solid lines are the numerical solution of ENO-RF-S-3, 800 points): (a) ENO-RF-3 (diamonds); (b) ENO-RF-S-3 (diamonds); (c) ENO-RF-A-3 (diamonds).

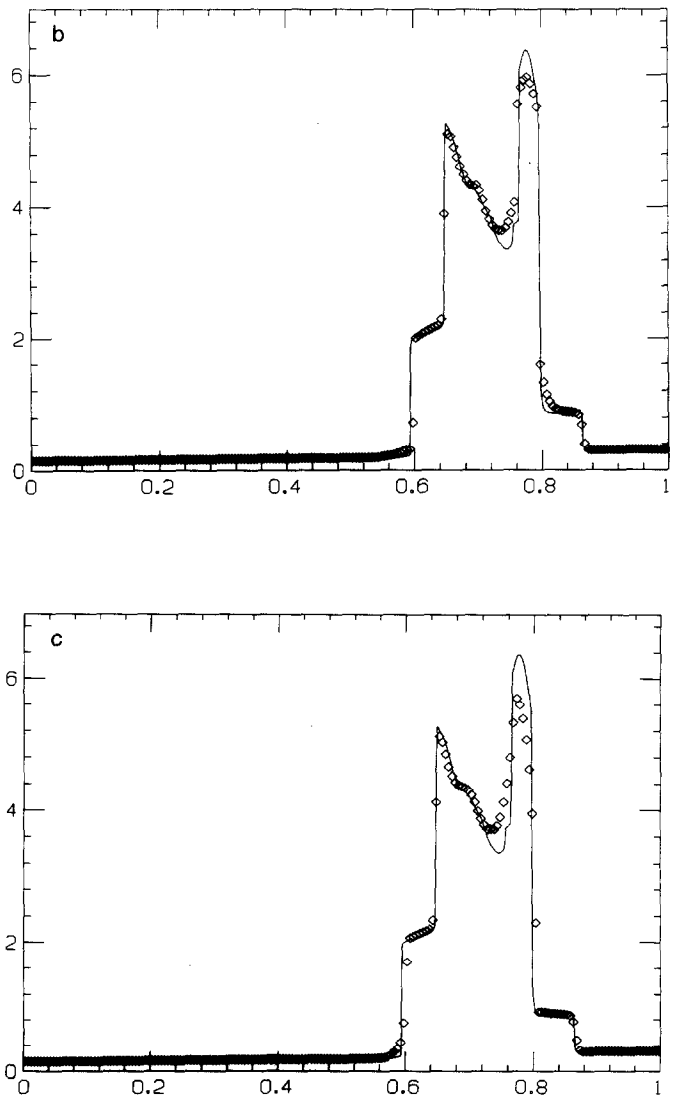


Fig. 13. (*Continued*)

This problem is difficult for shock capturing methods because both properties: high-order accuracy (to resolve the fine structure) and an oscillation-free shock

schemes have excellent possibilities for shock-turbulence computations.

VI. Concluding Remarks

ENO schemes based on fluxes and TVD Runge–Kutta type time discretizations seem to work very well in our numerical tests which include 1D and 2D scalar and systems problems. ENO-RF (Algorithm 2.3) seems preferred, since this method is half as expensive as ENO-LLF or ENO-LF, gives better resolution, and seems always to converge to the correct entropy solution. To sharpen contact discontinuities, Algorithm 3.1 (for 1D) or Algorithm 3.2 (for 1D or 2D) can be used. High-order ENO schemes show their special advantages for problems which have both discontinuities and detailed fine structure in smooth regions, e.g., Example 8 and Example 9 in Section V. Among the practical things needing further investigation are the resolution of rarefaction corners (discontinuities in derivatives). Finally, theoretical justification for the evident stability of these methods would be quite welcome.

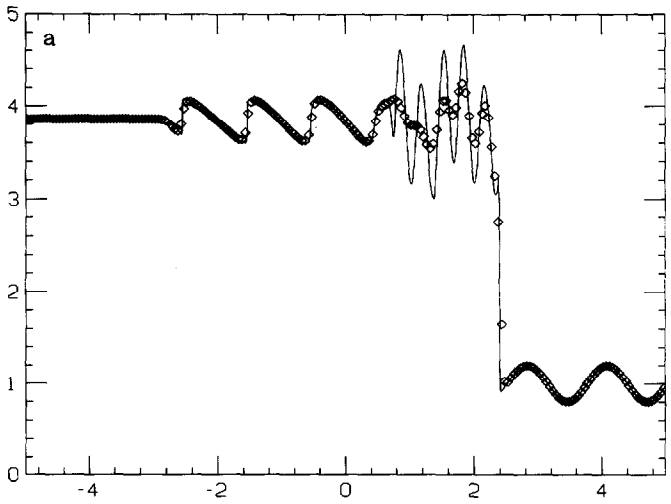


FIG. 14. (5.8a)–(5.11), $t = 1.8$, density (solid lines are the numerical solution of ENO-RF-3, 1600 points): (a) ENO-RF-3, 200 points (diamonds); (b) ENO-RF-3, 400 points (diamonds); (c) Second-order MUSCL-type TVD scheme, 800 points (diamonds).

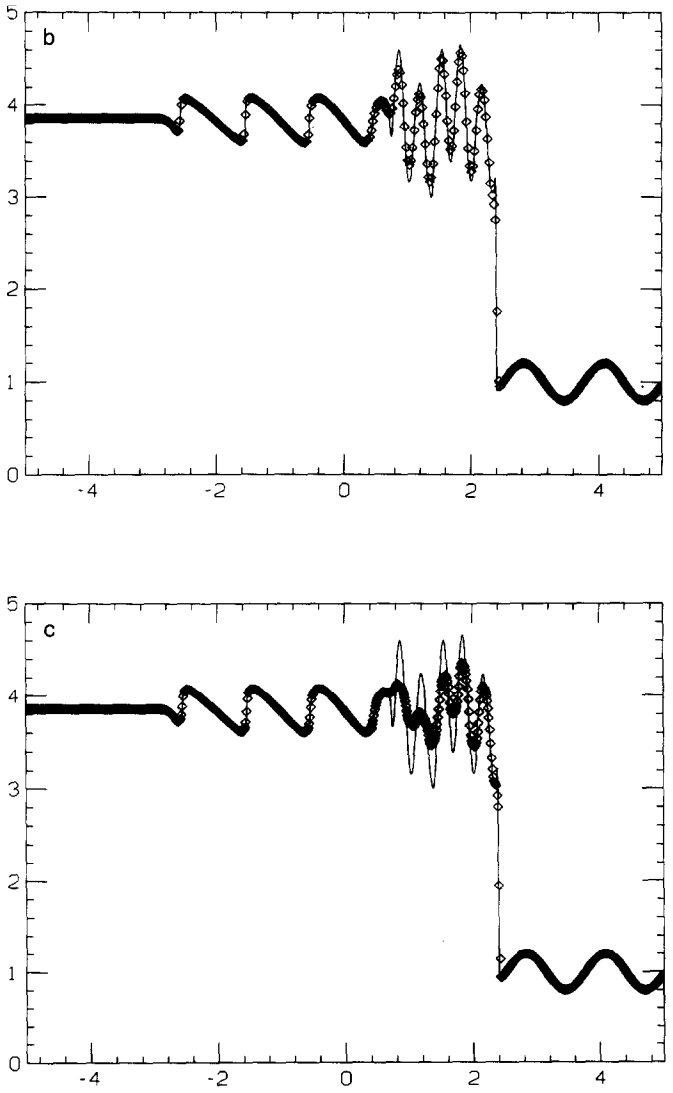


Fig. 14. (*Continued*)

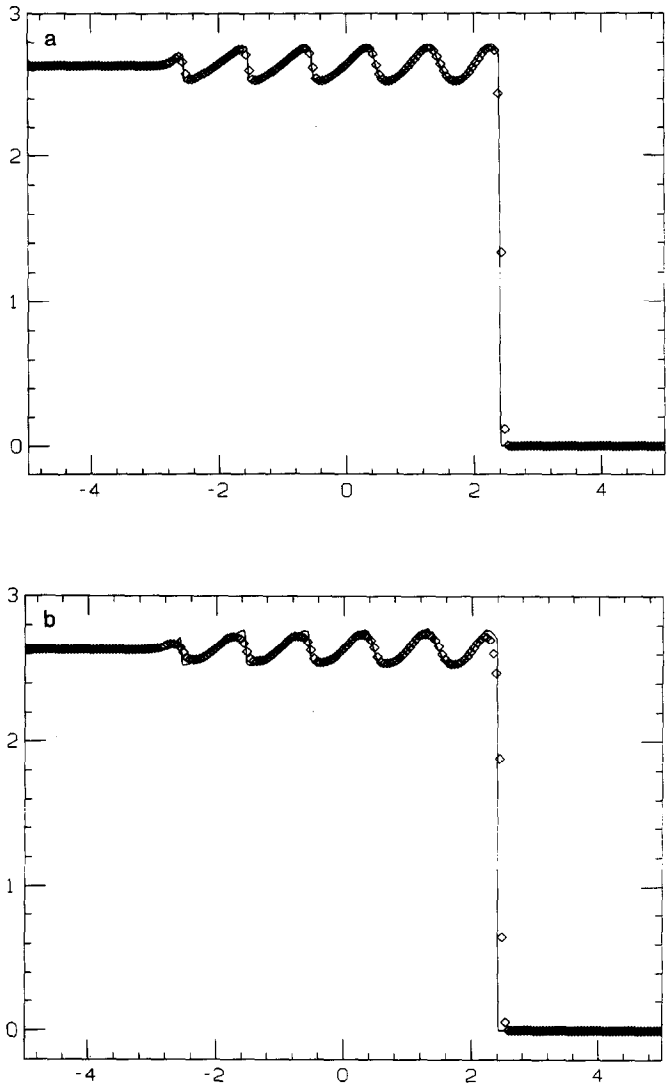


FIG. 15. (5.8a)–(5.11), 200 points, $t = 1.8$ (solid lines are the numerical solution of ENO-RF-3, 1600 points): (a) ENO-RF-3, velocity (diamonds); (b) Second-order MUSCL-type TVD scheme, velocity (diamonds); (c) ENO-RF-3, pressure (diamonds); (d) Second-order MUSCL-type TVD schemes, pressure (diamonds).

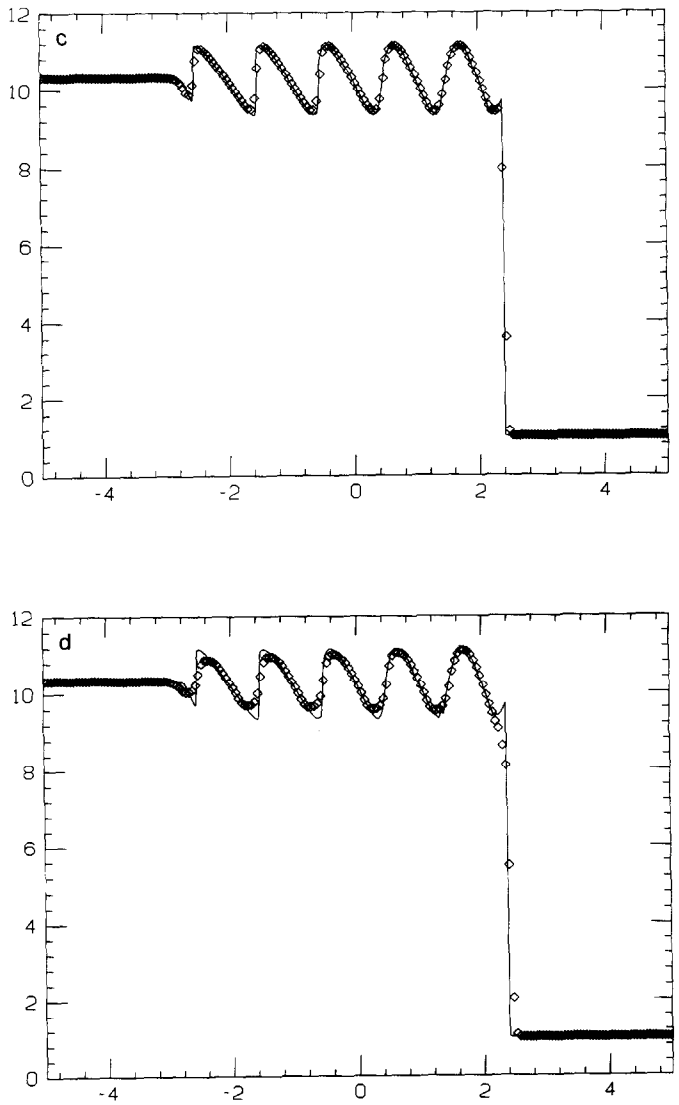


Fig. 15. (*Continued*)

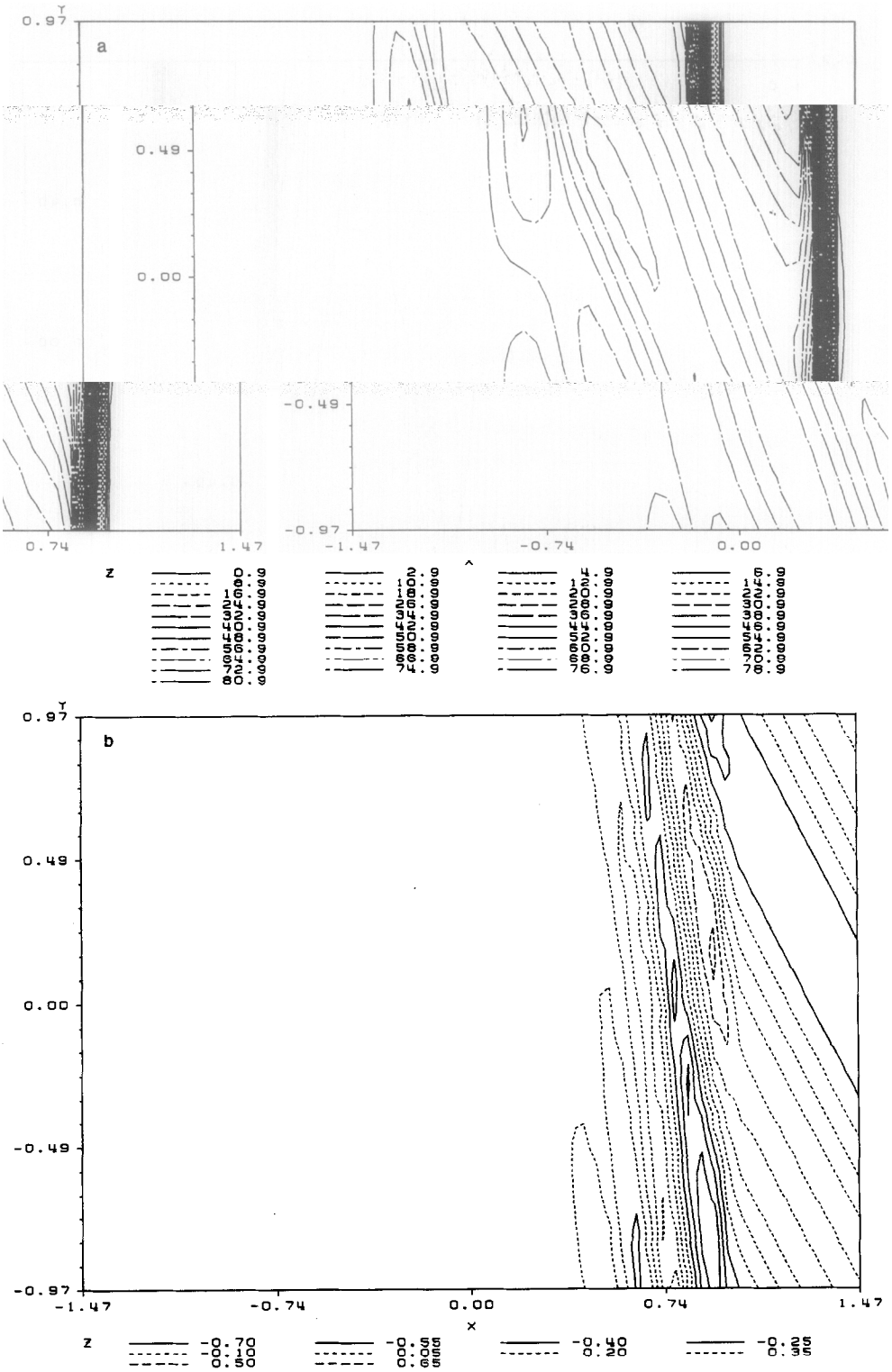


FIG. 16. ENO-RF-3, Example 9, $t = 0.2$ (level curves of numerical solutions): (a) 60×40 points, pressure; (b) 60×40 points, vorticity; (c) 60×40 points, entropy; (d) 120×80 points, pressure; (e) 120×80 points, vorticity; (f) 120×80 points, entropy.

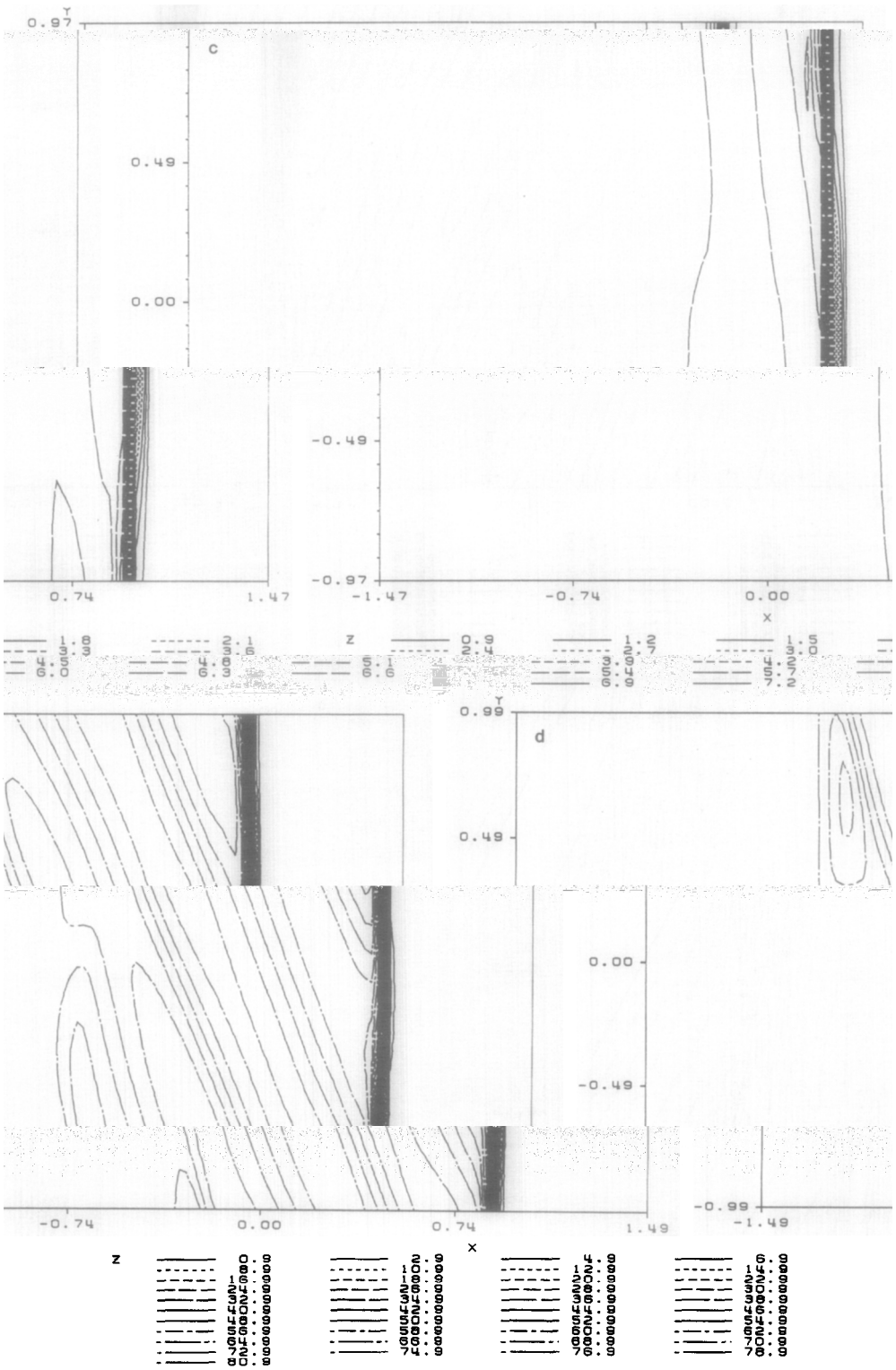


Fig. 16. (*Continued*)

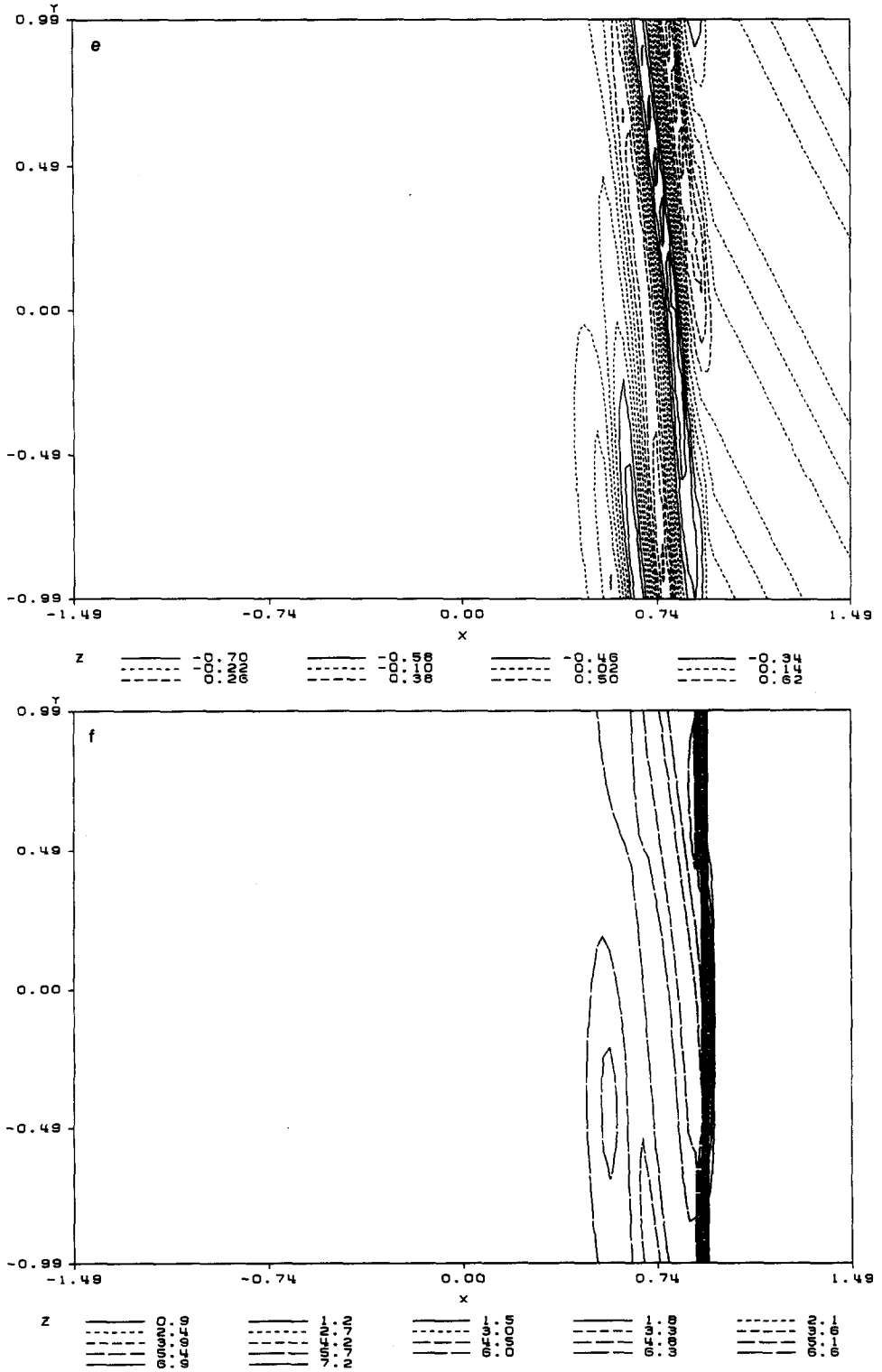


Fig. 16. (*Continued*)

ACKNOWLEDGMENTS

We thank David Gottlieb, Ami Harten, Lawrence Sirovich, Bjorn Sjogreen, Huanan Yang, and Thomas Zang for helpful discussions and/or for suggesting numerical examples. We also thank the referees for some constructive comments on the first draft of this paper.

REFERENCES

1. M. CRANDALL AND A. MAJDA, *Math. Comput.* **34**, 1 (1980).
2. A. HARTEN, "Preliminary Results on the Extension of ENO Schemes to Two-Dimensional Problems," in *Proceedings of the International Conference on Hyperbolic Problems*, Saint-Etienne, January 1986.
3. A. HARTEN AND S. OSHER, *SIAM J. Numer. Anal.* **24**, 279 (1987).
4. A. HARTEN, B. ENGQUIST, S. OSHER, AND S. CHAKRAVARTHY, *J. Comput. Phys* **71**, 231 (1987).
5. A. HARTEN, S. OSHER, B. ENGQUIST, AND S. CHAKRAVARTHY, *J. Appl. Numer. Math.* **2**, 347 (1986).
6. A. HARTEN, ICASE Report 87-56, August 1987 (unpublished).
7. A. HARTEN, in preparation.
8. J. MCKENZIE AND K. WESTPHAL, *Phys. Fluids* **11**, 2350 (1968).
9. S. OSHER, *SIAM J. Numer. Anal.* **22**, 947 (1985).
10. S. OSHER AND S. CHAKRAVARTHY, *SIAM J. Numer. Anal.* **21**, 955 (1984).
11. P. ROE, *J. Comput. Phys.* **43**, 357 (1981).
12. C.-W. SHU AND S. OSHER, ICASE Report 87-33; *J. Comput. Phys.* **77**, 439 (1988).
13. D. WAGNER, *SIAM J. Math. Anal.* **14**, 534 (1983).
14. P. WOODWARD AND P. COLELLA, *J. Comput. Phys.* **54**, 115 (1984).
15. H. YANG, Ph. D. thesis, UCLA Math. Dept., 1988; *J. Comput. Phys.* (1990) to appear.
16. S. ZALESAK, "A Preliminary Comparison of Modern Shock-Capturing Schemes: Linear Advection," in *Advances in Computer Methods for Partial Differential Equations, VI*, edited by R. Vichnevetsky and R. Stepleman (IMACS, New Brunswick, NJ, 1987).
17. T. ZANG, M. HUSSAINI, AND D. BUSHNELL, *AIAA J.* **22**, 13 (1984).
18. T. ZANG, D. KOPRIVA, AND M. HUSSAINI, "Pseudospectral Calculation of Shock Turbulence Interaction," in *Proceedings of Numerical Methods Conference*, edited by C. Taylor (Pineridge Press, Swansea, U.K., 1984), pp. 210–222.