JOURNAL OF COMPUTATIONAL PHYSICS 106, 62–76 (1993)

A Finite-Volume High-Order ENO Scheme for Two-Dimensional Hyperbolic Systems*

JAY CASPER

Vigyan, Inc., 30 Research Drive, Hampton, Virginia 23666

AND

H. L. ATKINS

Theoretical Flow Physics Branch, NASA Langley Research Center, Hampton, Virginia 23665

Received April 26, 1991; revised July 22, 1992

We consider here the finite-volume approach in developing a two-dimensional, high-order accurate, essentially non-oscillatory shock-capturing scheme. Such a scheme achieves high-order spatial accuracy by a piecewise polynomial approximation of the solution from its cell averages. High-order Runge–Kutta methods are employed for time integration, thus making such schemes best-suited for unsteady problems. The focal point in our development is a high-order spatial operator which will retain high-order accuracy in smooth regions, yet avoid the oscillatory behavior that is associated with interpolation across steep gradients. Such an operator is first presented within the context of a scalar function on a rectangular mesh and then extended to hyperbolic systems of equations and curvilinear meshes. Spatial and temporal accuracy are validated through grid refinement studies, involving the solutions of scalar hyperbolic equations and the Euler equations of gas dynamics. Through a control-volume approach, we find that this two-dimensional scheme is readily applied to inviscid flow problems involving solid walls and non-trivial geometries. Results of a physically relevant, numerical experiment are presented for qualitative and quantitative examination.

1. INTRODUCTION

In recent years, a class of numerical schemes for solving hyperbolic partial differential equations has been developed, which generalizes the first-order method of Godunov [1] to an arbitrary order of accuracy. High-order accuracy is obtained, wherever the solution is smooth, by an *essentially non-oscillatory* (ENO) piecewise polynomial reconstruction procedure, which yields high-order pointwise information from the cell averages of the solution, at a given point in time. When applied to piecewise smooth initial data, this reconstruction enables a flux computation which is of high-order accuracy, wherever the function is smooth, and avoids a Gibbs phenomenon at discontinuities.

* This work was supported by NASA Langley Research Center, Hampton, VA, under Grant NAG1-363 and Contract NAS1-18585.

The first step in extending Godunov's scheme beyond first-order accuracy was taken by Van Leer in [2], resulting in a second-order algorithm, to which many of the ideas for modern nonlinear schemes can be traced. Later, in [3, 4], Harten was able to achieve second-order accuracy and monotonicity by rigorously developing a class of *total-variation diminishing* (TVD) schemes. The schemes in this class are required to revert to first-order at local extrema, a result of ensuring that the solution's total variation is a non-increasing function in time. In order to achieve uniform second-order accuracy, Harten and Osher, in [5], were willing to accept a weaker version of the TVD criterion, thereby establishing a basis for what would become the class of high-order ENO schemes, to be fully developed for one-dimensional problems by Harten *et al.* [6].

It should be noted here the distinction between "high-order" and "high-resolution" schemes. High-resolution methods (e.g., [3, 4, 7]) are designed to capture discontinuities within a minimum number of points, while usually reverting to first order at smooth extrema. Therefore, such schemes are not the best choice for the resolution of smooth, low amplitude flow phenomena. On the other hand, the dual capacity for shock-capturing and achieving high-order accuracy in smooth regions makes ENO schemes well suited for the study of aeroacoustic and transition-related problems. They may therefore serve as an alternative to spectral methods in solving such problems when shocks or complex geometries are involved.

The application of high-order ENO schemes to areas of scientific and industrial interest obviously requires compressible flow solutions in more than one spatial dimension. In [8], Harten was the first to present a two-dimensional extension of the finite-volume ENO schemes in [6]. Because these results were preliminary, their scope was confined to solutions of scalar equations on a rectangular mesh.

0021-9991/93 $5.00

In this paper, we consider the finite-volume approach in developing multi-dimensional, high-order accurate ENO schemes for hyperbolic systems of conservation laws, as well as their application on curvilinear grids.

Other multi-dimensional ENO schemes have been proposed, most notably those of Shu and Osher in [9, 10]. These schemes are radically different from the ENO schemes in [6], in that the control-volume formulation is discarded in favor of one which involves only point values. The principal motivation for this approach is the greater computational efficiency of finite-difference algorithms, when accuracy above second order is desired. However, the finite-volume formulation is not without its attributes. Among those we will address are grid conservation, improved resolution of multi-dimensional flow features, and easier implementation of most boundary conditions.

In the two sections that follow, we formulate the discrete solution of a two-dimensional hyperbolic equation in terms of control volumes and the method of lines, and then we discuss the general design by which we can achieve high-order accuracy. In Section 4, we present a spatial operator, by which we can recover high-order pointwise information of a two-dimensional scalar function from its cell averages. This reconstruction operator is then extended to a vector-valued function in Section 5, in order that we may apply our ENO schemes to a system of two-dimensional conservation laws. Section 6 extends our design to a general curvilinear coordinate system, after which we present the results of numerical experiments which test all of the preceding ideas. In particular, we apply these ideas to the Euler equations of gas dynamics. These results demonstrate the first successful application of finite-volume high-order ENO schemes to inviscid flow problems involving non-trivial geometries and solid walls.

2. FINITE-VOLUME FORMULATION

We wish to design high-order accurate ENO schemes for the numerical approximation of weak solutions of a hyperbolic system of conservation laws

$$u_t + f(u)_x + g(u)_y = 0, \tag{1a}$$

subject to given initial conditions,

$$u(x, y, 0) = u^0(x, y). \tag{1b}$$

The function $u = (u^1, u^2, ..., u^m)^T$ is a state vector and the fluxes $f(u)$ and $g(u)$ are vector-valued differentiable functions of m components. We assume that the system (1a) is *hyperbolic* in the sense that the $m \times m$ Jacobian matrices

$$A(u) = \frac{\partial f}{\partial u}, \qquad B(u) = \frac{\partial g}{\partial u}$$

are such that any linear combination of A and B has m real eigenvalues $\{\lambda^k(u)\}$ and a complete set of m right eigenvectors $\{r^k(u)\}$ and left eigenvectors $\{l^k(u)\}$, which we assume to satisfy the orthonormal relation $l^i \cdot r^j = \delta_{ij}$.

We assume that the initial-value problem (IVP) (1) is well-posed in the sense that the solution u depends continuously on the initial data and that this solution is piecewise smooth, with at most a finite number of discontinuities. Appealing to well-known theoretical results, we know that a weak solution of the IVP (1) must satisfy (1a) in the sense of distribution theory. Furthermore, this is equivalent to requiring that u obey the integral form of (1a), where the limits of integration can reflect any smoothly bounded domain in the $x-y$ plane and any time interval (t^1, t^2). Seeking such a solution, let the set $\{\mathscr{C}_{ij}\}$, where

$$\mathscr{C}_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}],$$

denote a rectangular partition of the $x-y$ plane, with (x_i, y_j) denoting the centroid of each rectangle. With a semi-discrete formulation in mind, we note that, for every rectangle $\mathscr{C}_{ij}$, a weak solution of (1a) must satisfy

$$\frac{\partial}{\partial t}\bar{\bar{u}}_{ij}(t) = -\frac{1}{a_{ij}}\Big[\hat{f}_{i+1/2,j}(t) - \hat{f}_{i-1/2,j}(t) + \hat{g}_{i,j+1/2}(t) - \hat{g}_{i,j-1/2}(t)\Big], \tag{2a}$$

where $a_{ij} = \Delta_i x\, \Delta_j y$ is the area of $\mathscr{C}_{ij}$, and

$$\bar{\bar{u}}_{ij}(t) = \frac{1}{a_{ij}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} u(x, y, t)\, dx\, dy \tag{2b}$$

is the cell average of u over the control volume at time t. The fluxes $\hat{f}$ and $\hat{g}$ are given by

$$\hat{f}_{i+1/2,j}(t) = \int_{y_{j-1/2}}^{y_{j+1/2}} f(u(x_{i+1/2}, y, t))\, dy; \tag{2c}$$

$$\hat{g}_{i,j+1/2}(t) = \int_{x_{i-1/2}}^{x_{i+1/2}} g(u(x, y_{j+1/2}, t))\, dx. \tag{2d}$$

We now treat (2a) as a system of ordinary differential equations for the purpose of time discretization, using a "method-of-lines" approach. Along any $t =$ constant line, the right-hand side of (2a) is strictly a spatial operation in u, and we rewrite this equation, for fixed t, in the abstract operator-product form

$$\frac{\partial}{\partial t}\bar{\bar{u}}_{ij}(t) = (\mathscr{L}u(t))_{ij}, \tag{3}$$

thus effectively "separating" the spatial and temporal operations for computing weak solutions of (1).

In [9], Runge–Kutta methods are presented for the time discretization of ordinary differential equations which are of high-order accuracy and total-variation diminishing (TVD), in the sense that the temporal operator itself does not increase the total variation of the solution. We will employ these Runge–Kutta methods to solve (3) and achieve our desired accuracy in time. It remains to approximate the operator $\mathscr{L}$ to high order.

3. A HIGH-ORDER ACCURATE SCHEME

Given $\{\bar{\bar{u}}^n\}$, cell averages of a piecewise smooth solution $u(x, y, t^n)$ of (1), we desire a high-order accurate numerical solution operator E_h which will update these averages to time $t^{n+1} = t^n + \Delta t$. Specifically, we require that E_h be rth-order accurate in the sense of local truncation error; i.e., when applied once to the exact solution $\{\bar{\bar{u}}^n\}$, E_h satisfies

$$E_h \bar{\bar{u}}^n - \bar{\bar{u}}^{n+1} = O(h^{r+1}), \tag{4}$$

wherever u is sufficiently smooth, with $h = \max_{i,j}\{\Delta_i x, \Delta_j y\}$. Furthermore, we desire that our numerical update scheme avoid the development of spurious $O(1)$ oscillations near discontinuities in u. In order to achieve this property, we require our operator to be *essentially non-oscillatory*. Such an operator satisfies, in the one-dimensional scalar case,

$$TV(E_h \bar{u}) \leqslant TV(u) + O(h^{1+q}), \tag{5}$$

for some $q > 0$, where TV represents total variation in x. For details concerning the initial development of ENO schemes, the reader is referred to [6] and the references therein.

Employing the formulation (3) for the numerical update of the solution $\{\bar{\bar{u}}^n\}$, we discretize the temporal operation by using a Runge–Kutta method, which we write in the form

$$\bar{\bar{u}}_{ij}^{(l)} = \sum_{m=0}^{l-1} [\alpha_{lm} \bar{\bar{u}}_{ij}^{(m)} + \beta_{lm} \Delta t (\mathscr{L} u^{(m)})_{ij}], \qquad l = 1, 2, \ldots, p,$$
$$\bar{\bar{u}}_{ij}^{(0)} = \bar{\bar{u}}_{ij}^{n}, \qquad \bar{\bar{u}}_{ij}^{(p)} = \bar{\bar{u}}_{ij}^{n+1}. \tag{6}$$

The order of accuracy achieved by this time discretization, as well as its TVD property, is determined by the values of the integer p and the coefficients α and β (see [9] for details).

Now, if we assume that the scheme (6) achieves our desired rth-order accuracy in time, then clearly this scheme satisfies (4) if we can evaluate $(\mathscr{L} u(t))_{ij}$, the exact spatial operation on the right-hand side of (2a). However, this will not be possible, in the general nonlinear case of (1a). Therefore, we replace the operator $\mathscr{L}$ with a discrete spatial operator L, which acts upon the averages $\{\bar{\bar{u}}(t)\}$ and approximates $\mathscr{L}$ to high order. To this end, we see that if

$$L\bar{\bar{u}}(t) = \mathscr{L} u(t) + O(h^r), \tag{7a}$$

then, upon replacement of $\mathscr{L}$ in (6), the local truncation error of our fully discrete scheme will satisfy (4), as required.

We define L explicitly by

$$(L\bar{\bar{u}}(t))_{ij} = -\frac{1}{a_{ij}} [\bar{f}_{i+1/2,j}(t) - \bar{f}_{i-1/2,j}(t) + \bar{g}_{i,j+1/2}(t) - \bar{g}_{i,j-1/2}(t)], \tag{7b}$$

where $\bar{f}$ and $\bar{g}$ are to be designed as high-order numerical approximations to $\hat{f}$ and $\hat{g}$ in (2c)–(2d). The first step in obtaining these high-order accurate numerical fluxes is the correspondingly high-order approximation of the solution $u(x, y, t)$. We note that this approximation must be pointwise, whereas the information we have at any time t is that of the cell averages (2b). And clearly, since

$$\bar{\bar{u}}_{ij}(t) = u(x_i, y_j, t) + O(h^2),$$

there is an inherent limit on the order of accuracy of our scheme, if we use the cell averages directly in the flux calculation. Therefore, let R^2 be a spatial operator which reconstructs the set of cell averages and yields a two-dimensional, piecewise polynomial $R^2(x, y; \bar{\bar{u}}(t))$ of degree $r-1$ which approximates $u(x, y, t)$, with a truncation error of $O(h^r)$, wherever u is sufficiently smooth. We write this relationship in the form

$$R^2(x, y; \bar{\bar{u}}(t)) = u(x, y, t) + e(x, y)\, h^r. \tag{8}$$

Such an operator is discussed in Section 4.

To include the more general case, where $f(u)$, in (2c), cannot be integrated in closed form, we will approximate this integration by Gaussian quadrature. In order to express the error made by this approximation, let $q(x)$ be a $C^{(2K)}$ function whose integral on $[a, b]$ we approximate by the "classical" Gaussian quadrature, i.e., relative to the unit weight function on the interval $[-1, 1]$. It can be shown (e.g., [11]) that the error made by this approximation with a K-point quadrature is given by

$$\int_a^b q(x)\, dx - \frac{b-a}{2} \sum_{k=1}^{K} c_k q(x_k) = \frac{q^{(2K)}(\xi)}{(2K)!} \int_a^b P_K^2(x)\, dx,$$

for some ξ in (a, b), where P_K is the polynomial of degree K in the orthogonal basis that spans the space of polynomials on $[a, b]$, of degree not exceeding K. This quadrature is exact when $q(x)$ is a polynomial of degree less than or equal to $2K-1$. The roots of $P_K(x) =$

$(x-x_1)(x-x_2)\cdots(x-x_K)$ are real, distinct, and lie on the interval (a, b), making it clear that the above truncation error is $O((b-a)^{2K+1})$. Correlating this error to the $(r-1)$th-degree polynomial reconstruction (8), we see that for $r \leqslant 2K$, this truncation error is at worst $O(h^{r+1})$ when $r-1$ is odd and $O(h^{r+2})$ when $r-1$ is even. Therefore, using the "larger" error, for fixed x and t, and sufficiently smooth f, the approximation of the flux integral (2c) by Gaussian quadrature satisfies

$$\begin{aligned}\hat{f}_{i+1/2,j}(t) &= \int_{y_{j-1/2}}^{y_{j+1/2}} f(u(x_{i+1/2}, y, t))\, dy \\ &= \frac{\Delta_j y}{2} \sum_{k=1}^{K} c_k f(u(x_{i+1/2}, y_k, t)) \\ &\quad + s(x_{i+1/2}, \eta)\, h^{r+1}, \end{aligned} \tag{9a}$$

for some η in $(y_{j-1/2}, y_{j+1/2})$, and s relates to the quadrature error in the previous equation.

Now, let $v_h(x, y)$ denote the (global) piecewise polynomial approximation to u which is determined by the reconstruction operator R^2 in (8) and therefore satisfies

$$v_h(x, y) = u(x, y, t) + e(x, y)\, h^r, \tag{9b}$$

for fixed t, wherever u is sufficiently smooth. Since f is assumed differentiable in u, it is therefore Lipschitz continuous in u, and thus, for fixed t,

$$f(u(x, y, t)) = f(v_h(x, y)) + d(x, y)\, h^r, \tag{9c}$$

where $d(x, y) = O(e(x, y))$. Finally, we substitute (9c) into the quadrature in (9a), and we see that

$$\begin{aligned}\hat{f}_{i+1/2,j}(t) = \frac{\Delta_j y}{2} \sum_{k=1}^{K} c_k [f(v_h(x_{i+1/2}, y_k)) \\ + d(x_{i+1/2}, y_k)\, h^r] + s(x_{i+1/2}, \eta)\, h^{r+1}. \end{aligned} \tag{9d}$$

Therefore, if we define our "abstract" numerical flux $\bar{f}_{i+1/2,j}(t)$ in (7b) by

$$\bar{f}_{i+1/2,j}(t) = \frac{\Delta_j y}{2} \sum_{k=1}^{K} c_k f(v_h(x_{i+1/2}, y_k)), \tag{10a}$$

then the error made by the approximate flux difference $\bar{f}_{i+1/2,j}(t) - \bar{f}_{i-1/2,j}(t)$ in the definition (7b) is given by

$$\begin{aligned}&\hat{f}_{i+1/2,j}(t) - \hat{f}_{i-1/2,j}(t) \\ &\quad = \bar{f}_{i+1/2,j}(t) - \bar{f}_{i-1/2,j}(t) \\ &\qquad + \frac{\Delta_j y}{2} \sum_{k=1}^{K} c_k [d(x_{i+1/2}, y_k) - d(x_{i-1/2}, y_k)]\, h^r \\ &\qquad + [s(x_{i+1/2}, \eta) - s(x_{i-1/2}, \eta)]\, h^{r+1}. \end{aligned} \tag{10b}$$

Clearly, if d and s are Lipschitz continuous on $[x_{i-1/2}, x_{i+1/2}]$ for each y, then the error relation in (10b) satisfies

$$\begin{aligned}&\hat{f}_{i+1/2,j}(t) - \hat{f}_{i-1/2,j}(t) \\ &\quad = \bar{f}_{i+1/2,j}(t) - \bar{f}_{i-1/2,j}(t) + O(h^{r+2}). \end{aligned} \tag{10c}$$

Moreover, a symmetrical argument can be used to show that

$$\begin{aligned}&\hat{g}_{i,j+1/2}(t) - \hat{g}_{i,j-1/2}(t) \\ &\quad = \bar{g}_{i,j+1/2}(t) - \bar{g}_{i,j-1/2}(t) + O(h^{r+2}), \end{aligned} \tag{10d}$$

where

$$\bar{g}_{i,j+1/2}(t) = \frac{\Delta_i x}{2} \sum_{k=1}^{K} c_k\, g(v_h(x_k, y_{j+1/2})). \tag{10e}$$

Noting that the area a_{ij} is $O(h^2)$, we see that upon substitution of the numerical fluxes (10a) and (10e), which satisfy the error relations (10c) and (10d), into (7b), we have thus designed the spatial operator L that satisfies (7a), and therefore, when substituted for $\mathscr{L}$ in (6), yields a numerical solution operator E_h which is rth-order accurate in the sense of local truncation error. We note here that the desired truncation errors given by (10c)–(10d) are achieved only if the functions representing the errors due to the solution approximation (8) and the quadrature (9a) are Lipschitz continuous on $\mathscr{C}_{ij}$.

We now wish to modify the "abstract" numerical fluxes (10a) and (10e) such that (4) still holds in regions where the solution is smooth and, in addition, these fluxes will account for possible discontinuities in u. This modification follows naturally from the reconstruction step (8), by which the function $v_h(x, y)$, in (9b), can discontinuous at cell interfaces. Let $\mathscr{P}_{ij}(x, y)$ denote the local representation of $v_h(x, y)$ within a cell $\mathscr{C}_{ij}$. The differing representations of $\mathscr{P}_{ij}(x, y)$, from cell to cell, will create discontinuities at the cell boundaries. The relative size of these local "jumps" is on the level of the interpolation error in smooth regions and is $O(1)$ near discontinuities in u. Therefore, in order to resolve these discontinuities, the flux integrands in (10a) and (10e) are replaced by

$$f^{\mathrm{Rm}}[\mathscr{P}_{i,j}(x_{i+1/2}, y_k), \mathscr{P}_{i+1,j}(x_{i+1/2}, y_k)], \tag{11a}$$

$$g^{\mathrm{Rm}}[\mathscr{P}_{ij}(x_k, y_{j+1/2}), \mathscr{P}_{i,j+1}(x_k, y_{j+1/2})], \tag{11b}$$

where $f^{\mathrm{Rm}}[u_1, u_2]$ denotes the flux, across $x = 0$, associated with the solution to the Riemann problem whose initial states are u_1 and u_2.

To summarize, we write our scheme as the Runge–Kutta method,

$$\bar{\bar{u}}_{ij}^{(l)} = \sum_{m=0}^{l-1} [\alpha_{lm} \bar{\bar{u}}_{ij}^{(m)} + \beta_{lm} \Delta t (L\bar{\bar{u}}^{(m)})_{ij}], \qquad l = 1, 2, ..., p,$$
$$\bar{\bar{u}}_{ij}^{(0)} = \bar{\bar{u}}_{ij}^{n}, \qquad \bar{\bar{u}}_{ij}^{(p)} = \bar{\bar{u}}_{ij}^{n+1}, \tag{12a}$$

where

$$(L\bar{\bar{u}}^{(m)})_{ij} = -\frac{1}{a_{ij}} [\bar{f}_{i+1/2,j}^{(m)} - \bar{f}_{i-1/2,j}^{(m)} + \bar{g}_{i,j+1/2}^{(m)} - \bar{g}_{i,j-1/2}^{(m)}] \tag{12b}$$

and

$$\bar{f}_{i+1/2,j}^{(m)} = \frac{\Delta_j y}{2} \sum_{k=1}^{K} c_k f^{\mathrm{Rm}} [\mathscr{P}_{i,j}(x_{i+1/2}, y_k), \mathscr{P}_{i+1,j}(x_{i+1/2}, y_k)], \tag{12c}$$

$$\bar{g}_{i,j+1/2}^{(m)} = \frac{\Delta_i x}{2} \sum_{k=1}^{K} c_k g^{\mathrm{Rm}} [\mathscr{P}_{i,j}(x_k, y_{j+1/2}), \mathscr{P}_{i,j+1}(x_k, y_{j+1/2})], \tag{12d}$$

with

$$\mathscr{P}_{ij}(x, y) = R^2(x, y; \bar{\bar{u}}^{(m)}), \qquad x, y \in \mathscr{C}_{ij}. \tag{12e}$$

Assuming the error functions $d(x, y)$ and $s(x, y)$ in (10b) to be globally Lipschitz continuous, the numerical solution operator E_h, defined by (12), is formally rth-order accurate in the sense of local truncation error as given by (4). Furthermore, if these error functions remain Lipschitz continuous for N time steps, where $N = t/\Delta t = O(1/h)$, we *assume* the cumulative error to be $O(h^r)$. Thus, at the end of such a computation, we have a set $\{\bar{\bar{v}}_{ij}^{N}\}$, approximations to the cell averages of u at time t^N which satisfy

$$\bar{\bar{v}}^N - \bar{\bar{u}}^N = O(h^r). \tag{13a}$$

If we desire our high-order accurate output in pointwise form, we simply perform one final reconstruction which, by (8) and (13a), will yield

$$R^2(x, y; \bar{\bar{v}}^N) = u(x, y, t^N) + O(h^r). \tag{13b}$$

In addition to the accuracy properties (13), we desire that if u should develop discontinuities, then the scheme (12) will avoid $O(1)$ spurious oscillations, and we will design the reconstruction operator R^2 to do so.

4. RECONSTRUCTION

The first and most important step in the high-order approximations of $\hat{f}$, $\hat{g}$ is the method by which we obtain a high-order accurate pointwise representation of the solution $u(x, y, t)$ from the given set of cell averages $\{\bar{\bar{u}}(t)\}$. For the purpose of clarity, we discuss the finer points of this procedure within the framework of a scalar function defined on a rectangular computational mesh.

In preparation, we find it necessary to briefly review the notion of essentially non-oscillatory interpolation. Let $w(x)$ be a piecewise smooth function in x whose values are known to us only at the discrete points $\{x_i\}$. We introduce $Q_n(x; w)$, an nth degree piecewise polynomial function of x, which interpolates w at the points $\{x_i\}$, i.e.,

$$Q_n(x; w) \equiv \bigcup_i q_{n,i}(x; w), \tag{14a}$$

$$Q_n(x; w) = q_{n,i}(x; w) \qquad \text{for} \quad x_i \leqslant x \leqslant x_{i+1}, \tag{14b}$$

$$Q_n(x_i; w) = w(x_i), \qquad \text{all } i, \tag{14c}$$

where $q_{n,i}(x)$ is the (unique) nth-degree polynomial that interpolates $w(x)$ at $n+1$ successive points $\{x_j\}$ which include x_i and x_{i+1}. We are, therefore, free to choose the other $n-1$ points, and we do so subject to the condition that $w(x)$ be "smoothest" on the chosen interval in some asymptotic sense.

It is suggested in [6] that information relevant to the smoothness of w can be extracted from a table of divided differences of w. For simplicity, we will use the notation $\mathscr{S}_j^k$ to denote the set of $k+1$ contiguous points whose left-most members is x_j, i.e.,

$$\mathscr{S}_j^k = \{x_j, x_{j+1}, ..., x_{j+k}\}.$$

We will then use $w[\mathscr{S}_j^k]$ to denote the kth-divided difference of w on the "stencil" $\mathscr{S}_j^k$. A divided difference table for w can be recursively constructed by

$$w[x_j] = w(x_j);$$
$$w[\mathscr{S}_j^k] = \frac{w[\mathscr{S}_{j+1}^{k-1}] - w[\mathscr{S}_j^{k-1}]}{x_{j+k} - x_j}.$$

It is argued in [6] that the magnitudes of these divided differences can serve as a tool to compare the relative smoothness of w on various stencils. Since we always assume any stencil we choose to be contiguous, we can identify a particular stencil by specifying its left-most index, which we denote by $j(i)$. In [6], Harten *et al.* suggest that $j(i)$ be chosen by the following hierarchical algorithm, which begins ($k = 1$) by setting

$$j_1(i) = i. \tag{15a}$$

In order to choose $j_{k+1}(i)$, $k = 1, ..., n-1$, we consider as candidates the two stencils which are obtained by adding a point to the left or right of the previously determined stencil.

We select the one in which the $(k+1)$th-order divided difference is smaller in magnitude, thus

$$j_{k+1}(i)=\begin{cases} j_k(i)-1, & \text{if } |w[\mathscr{S}^{k+1}_{j_k(i)-1}]|<|w[\mathscr{S}^{k+1}_{j_k(i)}]|,\\ j_k(i), & \text{otherwise.}\end{cases} \tag{15b}$$

Finally, we set $j(i)=j_n(i)$.

For h sufficiently small, the algorithm (15) will always be able to determine an $(n+1)$-point stencil of smoothness between any two discontinuities. Therefore if each $q_{n,i}(x; w)$ in (14b) is constructed with the aid of this adaptive-stencil type of interpolation, then wherever $w(x)$ is smooth, $Q(x; w)$ will satisfy

$$TV(Q_n(x; w)) \leqslant TV(w(x)) + O(h^{n+1}); \tag{16}$$

i.e. $Q_n(x; w)$ is an essentially non-oscillatory approximation to $w(x)$. Henceforth, $Q_n(x; w)$ will denote the ENO interpolating polynomial of degree n whose construction involves the adaptive-stencil algorithm (15).

We now describe the two-dimensional reconstruction operator. It might seem plausible to extend the reconstruction operator R in [6] to two dimensions by simply *summing* two one-dimensional operators, which is most commonly done in conventional finite-volume algorithms. As will become apparent, this approach will inherently limit a numerical solution operator to second-order accuracy, as it applies to the solution of a nonlinear equation. Nonetheless, we can describe the implementation of R^2 as a *product* of two one-dimensional, "reconstruction-by-primitive" operators in [6].

We are initially given a discrete set of cell averages $\{\bar{\bar{w}}_{ij}\}$ of a piecewise smooth function $w(x, y)$,

$$\bar{\bar{w}}_{ij}=\frac{1}{\Delta_i x\,\Delta_j y}\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{j-1/2}}^{y_{j+1/2}} w(x, y)\,dy\,dx. \tag{17a}$$

For $y_{j-1/2}<y<y_{j+1/2}$, we define a *primitive function* $\bar{W}_j(x)$ associated with w by

$$\bar{W}_j(x)=\int_{x_0}^{x}\frac{1}{\Delta_j y}\int_{y_{j-1/2}}^{y_{j+1/2}} w(\xi, y)\,dy\,d\xi. \tag{17b}$$

Seeking a relationship between pointwise values of $\bar{W}_j(x)$ and the discrete values $\{\bar{\bar{w}}_{ij}\}$, we see immediately from the definitions (17a)–(17b) that

$$\Delta_i x\,\bar{\bar{w}}_{ij}=\bar{W}_j(x_{i+1/2})-\bar{W}_j(x_{i-1/2}),$$

and we can, therefore, establish a relationship at the cell interfaces:

$$\bar{W}_j(x_{i+1/2})=\sum_{k=i_0}^{i}\Delta_k x\,\bar{\bar{w}}_{kj}. \tag{17c}$$

Now, if we interpret the notation $\bar{w}_j(x)$ as the line average in y of $w(x, y)$ for a fixed x, then the definition (17b) clearly implies

$$\frac{d}{dx}\bar{W}_j(x)=\frac{1}{\Delta_j y}\int_{y_{j-1/2}}^{y_{j+1/2}} w(x, y)\,dy\equiv\bar{w}_j(x). \tag{17d}$$

This suggests that if we approximate $\bar{W}_j(x)$ by $Q_r(x; \bar{W}_j)$, we can then obtain an approximation of $\bar{w}_j(x)$ by defining the first step in our reconstruction procedure as

$$(R(x; \bar{\bar{w}}))_j=\frac{d}{dx}Q_r(x; \bar{W}_j)\equiv\bar{v}_j(x). \tag{17e}$$

Then $\bar{v}_j(x)$ is a piecewise polynomial in x of degree $r-1$ which satisfies

$$\bar{v}_j(x)=\bar{w}_j(x)+O(h_x^r), \tag{17f}$$

wherever $\bar{w}_j(x)$ is sufficiently smooth in x, and $h_x=\max_i\{\Delta_i x\}$.

If the procedure (17) is performed for all j, then we have a set of piecewise polynomials $\{\bar{v}_j(x)\}$, each of which is a high-order approximation to each $\bar{w}_j(x)$. Clearly, from the definition (17d), the form of $\bar{w}_j(x)$ is equivalent to that of a one-dimensional cell average on the interval $[y_{j-1/2}, y_{j+1/2}]$. Therefore, for a fixed x, we now treat the set $\{\bar{v}_j(x)\}$ as one-dimensional cell averages in y of a piecewise smooth function $v(x, y)$, which we wish to reconstruct to high-order pointwise accuracy in y. Analogous to the method (17), we define another primitive function $W(x, y)$ associated with $v(x, y)$ by

$$W(x, y)=\int_{y_0}^{y} v(x, y)\,dy, \tag{18a}$$

whose pointwise values we know at cell interfaces

$$W(x, y_{j+1/2})=\sum_{k=j_0}^{j}\Delta_k y\,\bar{v}_k(x). \tag{18b}$$

Fitting the point values (18b) of $W(y)$ with a piecewise polynomial $Q_r(y; W)$ of degree r by ENO interpolation, we can obtain a high-order pointwise approximation to $v(x, y)$, in y, by defining the second reconstruction step

$$R(y; \bar{v}(x))=\frac{d}{dy}Q_r(y; W)\equiv p(x, y), \tag{18c}$$

where, for fixed x, $p(x, y)$ is a polynomial in y of degree $r-1$ that satisfies

$$p(x, y)=v(x, y)+O(h_y^r), \tag{18d}$$

wherever $v(x, y)$ is sufficiently smooth in y, and $h_y = \max_j \{\Delta_j y\}$.

Noting the reconstruction definitions and the error relationships above, we can see that the values obtained in (18c) are the high-order pointwise approximations to $w(x, y)$ which we desired from the initial cell averages $\{\bar{\bar{w}}_{ij}\}$, i.e.,

$$R^2(x, y; \bar{\bar{w}}) \equiv R(y; R(x; \bar{\bar{w}})) = w(x, y) + O(h^r). \quad (19)$$

We note here that the high-order relationship (19) cannot be achieved, for $r > 2$, by simply summing two one-dimensional operators in a "directionally split" fashion. For instance, upon reviewing equation (17e), which represents a purely one-dimensional reconstruction, we recover, not point values, but $\{\bar{v}_j\}$, line averages in y. Though, as a function in x, these line averages are high-order approximations to $\{\bar{w}_j\}$, as given by (17f), the pointwise truncation error in y is, at best, $O(h_y^2)$. A symmetrical argument can be made for a pointwise error in x of $O(h_x^2)$ for a 1D-reconstruction in the y direction. Thus, the directionally split approach to the two-dimensional reconstruction problem would yield a summation operator

$$R^+(x, y; \bar{\bar{w}}) \equiv R(x; \bar{\bar{w}}) + R(y; \bar{\bar{w}}) = w(x, y) + O(h^2). \quad (20)$$

The implementation of (20) in (12), for the solution of a nonlinear equation, yields a second-order scheme, regardless of the one-dimensional truncation error of R. Figure 1 depicts an example of a reconstruction stencil, for each of the operators (19) and (20), as based on a fourth-order one-dimensional operator R. The entire shaded area represents a fourth-order stencil, which might result from the product (19), and the darker shading, a stencil determined by the summation (20).

As for the high-order operator R^2, we also note that each of the one-dimensional reconstruction operators (17e) and (18c) is essentially non-oscillatory, due to the nature of the interpolating polynomial Q_r, as given by (16). Furthermore, we note that R^2 is "conservative," in the sense that the cell-averaging operator defined by the right-hand side of (17a) is its left-hand inverse, i.e.,

$$\frac{1}{\Delta_i x\, \Delta_j y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} R^2(x, y; \bar{\bar{w}})\, dy\, dx = \bar{\bar{w}}_{ij},$$

which is necessary in order that our numerical scheme (12) remain conservative. This property results directly from the various definitions in the reconstruction (17)–(18), and the fact that Q_r is an interpolating polynomial. It is the adaptive-stencil algorithm (15) that enables this reconstruction (for sufficiently small h) to be high-order accurate on any domain, where $w(x, y)$ is smooth, even if that region is near one in which $w(x, y)$ is discontinuous. Furthermore, algorithm (15) is ultimately responsible for the desired non-oscillatory behavior near a discontinuity, where the "jumps" in $R^2(x, y; \bar{\bar{w}})$ at cell boundaries become large relative to the mesh spacing.

We further note that the error coefficient $e(x, y)$ in (8), due to this reconstruction, becomes discontinuous at points where there is a change of orientation in the stencil of the associated interpolation. This discontinuity may occur at critical points of the function and/or its derivatives. It is clear that, when $e(x, y)$ fails to be Lipschitz continuous at a point, the truncation error of the approximate flux difference in (10c) is only $O(h^{r+1})$. We therefore expect the cumulative pointwise error due to N applications of the operator E_h to be only $O(h^{r-1})$ at such points, but to remain $O(h^r)$ away from these points. Owing to the essentially non-oscillatory nature of E_h, it is reasonable to expect the number of points at which $e(x, y)$ fails to be Lipschitz continuous to remain bounded as $h \to 0$. In this case, we expect the cumulative error of our numerical scheme to be $O(h^{r-1})$ in the L_∞ norm and $O(h^r)$ in the L_1 norm.

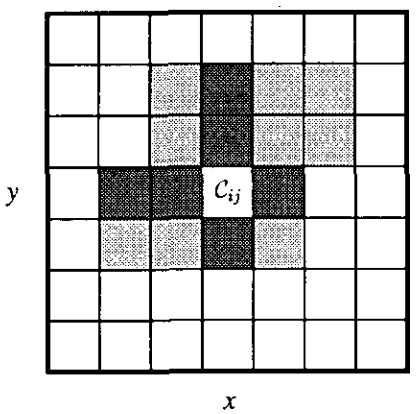

FIG. 1. Possible reconstruction stencils for the cell $\mathcal{C}_{ij}$, as determined by R^2 and R^+.

5. SYSTEMS OF CONSERVATION LAWS

In this section, we extend the scalar reconstruction procedure of Section 4 to solutions of hyperbolic systems of conservation laws. To this end, we now reconsider the IVP (1), whose solution u is a vector of m components, as are the fluxes $f(u)$ and $g(u)$. We now wish to develop a vector reconstruction operator, denoted by $\mathbf{R}^2$, which will reconstruct a set $\{\bar{\bar{u}}_{ij}\}$ of vector-valued cell averages to high-order pointwise accuracy. Clearly, in this context, $\bar{\bar{u}}_{ij}$ denotes a vector whose components are the cell averages of the scalar components of u. It would therefore seem natural, purely from the viewpoint of approximation theory, to reconstruct the set $\{\bar{\bar{u}}_{ij}\}$ by applying the scalar reconstruction R^2 in a component-wise fashion. However, this approach is valid only if we assume that $u(x, y, t)$ remains

free of discontinuities for the duration of a desired calculation. But in the more general case, when $u(x, y, t)$ is a solution of a non-linearly coupled system of equations, such a solution can admit the collision of discontinuities of the same or of different families, as well as their collision with boundaries, e.g., solid walls. In the vicinity of such collisions, a componentwise reconstruction may develop spurious oscillations during this brief encounter which do not dissipate as the discontinuities then distance themselves from one another, if indeed they ever do. Numerical experiments to demonstrate this potential problem in one spatial dimension can be found in [6]. In the following, we describe a vector reconstruction procedure which attempts to avoid this difficulty.

We begin by considering the constant coefficient case of (1): $f(u) = Au$, $g(u) = Bu$, where A and B are constant $m \times m$ matrices,

$$u_t + Au_x + Bu_y = 0, \tag{21a}$$

$$u(x, y, 0) = u^0(x, y). \tag{21b}$$

We also assume, for now, that our reconstruction takes place within the context of a Cartesian mesh. We note that the eigenvalues $\{\lambda_A^k\}$ of A and $\{\lambda_B^k\}$ of B are constant as are the right eigenvectors $\{r_A^k\}$, $\{r_B^k\}$ and left eigenvectors $\{l_A^k\}$, $\{l_B^k\}$. We assume that these eigenvectors are suitably normalized so that

$$l_A^i \cdot r_A^j = l_B^i \cdot r_B^j = \delta_{ij}. \tag{22a}$$

If we define the kth characteristic variables w_A^k and w_B^k by the dot products

$$w_A^k = l_A^k \cdot u, \qquad w_B^k = l_B^k \cdot u, \qquad k = 1, 2, ..., m, \tag{22b}$$

then it follows from (22a) that

$$u = \sum_{k=1}^{m} w_A^k r_A^k = \sum_{k=1}^{m} w_B^k r_B^k. \tag{22c}$$

It is argued in [6] that, in the constant-coefficient case, it is advantageous to use these characteristic variables (22b) in the reconstruction procedure, rather than u itself. This is due to the fact that, under the transformation (22b), the coupled system (21b) becomes an independent set of equations, thereby rendering any discontinuity in a particular characteristic variable "undetectable" by another (see [6] for details).

Therefore, in the case of a linear system (21a), we can describe our vector reconstruction as follows. Given $\{\bar{\bar{u}}_{ij}\}$, cell averages of a vector u, we begin by defining the cell averages of the characteristic variables in the x-direction by

$$\bar{\bar{w}}_{A,ij}^k = l_A^k \cdot \bar{\bar{u}}_{ij} \qquad \text{for } j \text{ fixed, all } i, \tag{23a}$$

and then perform the scalar reconstruction given by (17e) on these averages. Using the result (22c), we can define the first step of our linear vector reconstruction procedure by

$$\mathbf{R}(x; \bar{\bar{u}}) = \sum_{k=1}^{m} R(x; \bar{\bar{w}}_A^k)\, r_A^k \equiv \bar{v}_j(x), \tag{23b}$$

the right-hand side of which is the vector-valued analogy of the resulting line averages in (17f). In analogy with the two-step procedure in Section 4, the reconstruction (23b) is performed for all j. For x fixed, we then proceed by approximating the line-average characteristic variables in the y-direction by the dot product

$$\bar{w}_{B,j}^k = l_B^k \cdot \bar{v}_j. \tag{23c}$$

The scalar reconstruction (18c) is applied to the values (23c) and, for fixed x, we have a polynomial in y

$$\mathbf{R}(y; \bar{v}(x)) = \sum_{k=1}^{m} R(y; \bar{w}_B^k)\, r_B^k, \tag{23d}$$

which completes our reconstruction for the linear case and we write

$$\mathbf{R}^2(x, y; \bar{\bar{u}}) \equiv \mathbf{R}(y; \mathbf{R}(x; \bar{\bar{u}})) = \sum_{k=1}^{m} R\left(y; l_B^k \cdot \sum_{p=1}^{m} R(x; l_A^p \cdot \bar{\bar{u}})\, r_A^p\right) r_B^k. \tag{23e}$$

We now wish to generalize the reconstruction procedure (23) to the case of a nonlinear system. In the nonlinear case of (21a), the matrices $A(u)$, $B(u)$ are now functions of u, as are the eigenvalues $\{\lambda_A^k(u)\}$, $\{\lambda_B^k(u)\}$, and the eigenvectors $\{r_A^k(u)\}$, $\{r_B^k(u)\}$, $\{l_A^k(u)\}$, $\{l_B^k(u)\}$. Our extension will require the use of locally defined characteristic variables, in the following manner. In order to reconstruct $u(x, y)$ on the rectangle $\mathscr{C}_{ij}$, we first derive a set of local average characteristic variables $\{\bar{\bar{w}}_{nj}^k\}$, where n varies in the i-direction. We do this, for a fixed j, by computing dot products of $l_A^k(\bar{\bar{u}}_{ij})$ with the cell averages $\{\bar{\bar{u}}_{nj}\}$ associated with intervals in the immediate vicinity of $(x_{i-1/2}, x_{i+1/2})$; i.e.,

$$\bar{\bar{w}}_{nj}^k = l_A^k(\bar{\bar{u}}_{ij}) \cdot \bar{\bar{u}}_{nj}, \qquad n = i - q, ..., i + q, \tag{24a}$$

where q is the degree of the reconstruction polynomial. We then apply the scalar reconstruction operator R to this set of $2q + 1$ variables in (24a). This "local linearization" allows us to apply the linear vector reconstruction described in (23). The first step in our nonlinear vector reconstruction procedure then becomes

$$\mathbf{R}(x; \bar{\bar{u}}) = \sum_{k=1}^{m} R(x; \bar{\bar{w}}_{nj}^k)\, r_A^k(\bar{\bar{u}}_{ij}) \equiv \bar{v}_j(x). \tag{24b}$$

Upon performing (24b) for all j, then, for x fixed, we define a set of local "line-average" characteristic variables $\{\bar{w}_n^k\}$ in the y-direction:

$$\bar{w}_n^k = l_B^k(\bar{v}_j) \cdot \bar{v}_n \qquad \text{for} \quad n = j-q, \ldots, j+q. \tag{24c}$$

We complete our nonlinear reconstruction by applying the scalar operator R to the set of local variables in (24c) which results in

$$\mathbf{R}(y;\ \bar{v}(x)) = \sum_{k=1}^{m} R(y; \bar{w}^k(\bar{v}_j))\, r_B^k(\bar{v}_j). \tag{24d}$$

Thus, our vector reconstruction operator $\mathbf{R}^2$ is a product of (24b) and (24d), which approximates the vector averages, pointwise, to high-order,

$$\begin{aligned}\mathbf{R}^2(x, y; \bar{\bar{u}}(t)) &= \mathbf{R}(y; \mathbf{R}(x; \bar{\bar{u}}(t))) \\ &= u(x, y, t) + O(h^r).\end{aligned} \tag{24e}$$

and is designed to avoid the oscillatory behavior associated with colliding discontinuities.

6. CURVILINEAR COORDINATES

We now wish to generalize the spatial domain of solutions of the IVP (1). To this end let

$$x = x(\xi, \eta), \qquad y = y(\xi, \eta), \tag{25a}$$

denote a smooth transformation from the physical $x-y$ plane to the rectangular $\xi-\eta$ plane. The differential transformation is then

$$\begin{aligned} dx &= x_\xi\, d\xi + x_\eta\, d\eta \\ dy &= y_\xi\, d\xi + y_\eta\, d\eta, \end{aligned} \tag{25b}$$

whose Jacobian determinant J is

$$J = x_\xi y_\eta - y_\xi x_\eta. \tag{25c}$$

Let $\mathscr{C}_{ij}$ denote a discrete control volume in physical space which is mapped into a rectangle

$$\mathscr{D}_{ij} = (\xi_{i-1/2}, \xi_{i+1/2}) \times (\eta_{j-1/2}, \eta_{j+1/2}) \tag{25d}$$

by the transformation (25a). Then for any such region, our semi-discrete formulation for a weak solution of (1a) remains identical in form to (2a):

$$\begin{aligned}\frac{\partial}{\partial t}\bar{\bar{u}}_{ij}(t) = -\frac{1}{a_{ij}}\Big[&\hat{f}_{i+1/2,j}(t) - \hat{f}_{i-1/2,j}(t) \\ &+ \hat{g}_{i,j+1/2}(t) - \hat{g}_{i,j-1/2}(t)\Big].\end{aligned} \tag{26a}$$

Under the transformation (25), we interpret the cell average $\bar{\bar{u}}_{ij}(t)$ at time t as

$$\bar{\bar{u}}_{ij}(t) = \frac{1}{a_{ij}} \int_{\xi_{i-1/2}}^{\xi_{i+1/2}} \int_{\eta_{j-1/2}}^{\eta_{j+1/2}} u(\xi, \eta, t)\, J(\xi, \eta)\, d\xi\, d\eta, \tag{26b}$$

where a_{ij} is the area of $\mathscr{C}_{ij}$. The fluxes $\hat{f}$ and $\hat{g}$ are given by

$$\hat{f}_{i+1/2,j}(t) = \int_{\eta_{j-1/2}}^{\eta_{j+1/2}} \tilde{F}(u(\xi_{i+1/2}, \eta, t))\, d\eta, \tag{26c}$$

$$\hat{g}_{i,j+1/2}(t) = \int_{\xi_{i-1/2}}^{\xi_{i+1/2}} \tilde{G}(u(\xi, \eta_{j+1/2}, t))\, d\xi, \tag{26d}$$

where

$$\begin{aligned} \tilde{F}(u) &= y_\eta f(u) - x_\eta g(u), \\ \tilde{G}(u) &= x_\xi g(u) - y_\xi f(u), \end{aligned} \tag{26e}$$

and $f(u)$ and $g(u)$ are the Cartesian flux vectors in (1a).

Having defined the necessary terms of our finite-volume formulation in a curvilinear coordinate system, we now wish to discretize our spatial and temporal operations. Concerning our Runge–Kutta time discretization, the extension is straightforward. As for the spatial operation, we note two basic modifications that must be made.

The first alteration involves the reconstruction operator. Having clearly and consistently defined this procedure in terms of a rectangular mesh, we therefore would like to be able to perform the reconstruction procedure in $\xi-\eta$ space. Given a cell average $\bar{\bar{u}}_{ij}$, as defined by (26b), we see that the quantity $a_{ij}\bar{\bar{u}}_{ij}$ can be interpreted as a "cell average" of the function $u(\xi, \eta)\, J(\xi, \eta)$ on a corresponding computational cell of unit area. Therefore if we choose, instead, to reconstruct the set $\{a_{ij}\bar{\bar{u}}_{ij}\}$ of cell averages which are "scaled" by their respective areas, we transform the reconstruction procedure to rectangular coordinates and can simply use the procedures outlined in Sections 4 and 5. The piecewise polynomial we construct by this approach will be a high-order approximation to the product of $u(\xi, \eta)$ and $J(\xi, \eta)$, and therefore must be "re-scaled" by $J(\xi, \eta)$ in order to yield the pointwise approximation to u we need to evaluate the fluxes (26c)–(26d). Thus, we define our curvilinear reconstruction operator $\tilde{R}^2$ by

$$\tilde{R}^2(\xi, \eta; \bar{\bar{u}}) = \frac{1}{J(\xi, \eta)} R^2(\xi, \eta; a\bar{\bar{u}}), \tag{27}$$

with the vector analogy $\tilde{\mathbf{R}}^2$ defined in a similar manner. We do note that, in the case of vector reconstruction the various eigenvalues and eigenvectors required for this purpose are now the corresponding quantities of the matrices

$$\tilde{A} = y_\eta A - x_\eta B, \qquad \tilde{B} = x_\xi B - y_\xi A.$$

The second modification concerns the computational mesh itself. Within the context of the operator R^2 in Sections 4 and 5, we note that when we desire accuracy that is higher than second-order, a piecewise linear interpretation of a two-dimensional grid will not suffice. This is due to the fact that, once we require more than one point in the quadrature which will approximate the flux integrals (26c)–(26d), we can no longer assume that the quantities $\{x_\xi, x_\eta, y_\xi, y_\eta\}$ in (26e) are constant along a given cell boundary. We must, therefore, assume that the mesh is "truly curvilinear" and account for any change in these grid metrics at each point required in the quadrature.

Then, our flux calculation at a quadrature point in the computational space begins with the high-order representation of the solution by a polynomial $\mathscr{P}_{ij}(\xi, \eta)$, in each cell $\mathscr{D}_{ij}$, using the reconstruction (27). Our numerical fluxes can then be written

$$\bar{f}_{i+1/2,j}(t) = \frac{\Delta_j \eta}{2} \sum_{k=1}^{K} c_k \tilde{F}^{\mathrm{Rm}}[\tilde{\mathscr{P}}_{i,j}(\xi_{i+1/2}, \eta_k), \tilde{\mathscr{P}}_{i+1,j}(\xi_{i+1/2}, \eta_k)] \tag{28a}$$

$$\bar{g}_{i,j+1/2}(t) = \frac{\Delta_i \xi}{2} \sum_{k=1}^{K} c_k \tilde{G}^{\mathrm{Rm}}[\tilde{\mathscr{P}}_{i,j}(\xi_k, \eta_{j+1/2}), \tilde{\mathscr{P}}_{i,j+1}(\xi_k, \eta_{j+1/2})]. \tag{28b}$$

Now, assuming that the transformation (25a) is sufficiently smooth, the numerical scheme resulting from the curvilinear formulation (26) and the modifications mentioned above will be rth-order accurate in smooth regions, as defined by (4), and will avoid oscillations near steep gradients.

We make one further generalization. It so happens that there are a lot of applications of structured computational meshes for which a closed-form transformation is not available. For example, a set of grid points may be initially generated as a solution of a system of differential equations, after which they may then be subjected to some smoothing operator, e.g., Laplacian. In such a case we do not have a set of Eqs. (25a) from which to determine all the grid quantities necessary for the flux calculations (28).

However, given such a set of points, we might consider the equivalent of a locally defined set of transformation equations which are derived by polynomial approximation. By this we mean that each "grid line" through a set of points in physical space is approximated by polynomial interpolation and that all the necessary mesh quantities are calculated from these polynomials. For instance, on the face $\eta = \eta_{j-1/2}$ of $\mathscr{D}_{ij}$ in (25d), we need only determine the dependence of x and y on ξ. If, say, we desire a cubic polynomial representation of the transformation on this computational face, then we might choose the physical points (x, y) corresponding to the set $\{\xi_{i-3/2}, \xi_{i-1/2}, \xi_{i+1/2}, \xi_{i+3/2}\}$ along $\eta = \eta_{j-1/2}$ as our interpolation set. The distribution of ξ as a function of i can be arbitrarily taken as uniform. Denote by $x'(\xi)$ and $y'(\xi)$ the polynomials determined by this interpolation. The metrics that we need on this face are approximations to x_ξ and y_ξ, which are easily obtained by differentiating $x'(\xi)$ and $y'(\xi)$. This procedure is extended by symmetry to the determination of polynomials $x'(\eta)$ and $y'(\eta)$ on a $\xi = \text{const}$ face of $\mathscr{D}_{ij}$. Clearly, the physical points (x, y) we choose to approximate a grid line along a given cell face may have to reflect a one-sided interpolation near a computational boundary.

These same polynomials approximating the transformation along the four sides of $\mathscr{D}_{ij}$ can also be used to compute an approximate area of the physical cell $\mathscr{C}_{ij}$ by

$$a'_{ij} = \tfrac{1}{2} \oint_{\Gamma_{ij}} x'\, dy' - y'\, dx',$$

where Γ_{ij} is the boundary of $\mathscr{D}_{ij}$, and dx' and dy' are the corresponding approximations to (25b). An approximate Jacobian determinant, which we will denote by J', is also obtainable by the following reasoning: If the transformation (25a) existed, then clearly, we would have the relationship

$$a_{ij} = \int_{\xi_{i-1/2}}^{\xi_{i+1/2}} \int_{\eta_{j-1/2}}^{\eta_{j+1/2}} J(\xi, \eta)\, d\xi\, d\eta,$$

and therefore we may interpret a local area as a rectangular "cell average" of $J(\xi, \eta)$. Thus, having first determined approximations $\{a'_{ij}\}$ to the cell areas as described above, we apply our scalar reconstruction algorithm to this set and define J' by

$$J'(\xi, \eta) = R^2(\xi, \eta; a'). \tag{29}$$

We note that, without a known transformation (25a) from which to mathematically determine "sufficient smoothness," we can no longer make the claim concerning formal order of accuracy when this polynomial grid approximation is employed. In fact, a high-order polynomial approximation of a grid which is insufficiently smooth can result in the definition of control volumes whose boundaries are in serious error with a desired geometry.

7. COMMENTS

At this point, a few remarks are in order, the first of which concerns the stability of the scheme (12). We assume our scheme to be stable under the "conventional" explicit CFL time-step restriction. In the discrete scalar case, this is given by

$$\frac{\Delta t}{\mathrm{CFL}} = \min_{i,j} \left[\frac{a_{ij}}{|f'(u_{ij})|\, \Delta_j y + |g'(u_{ij})|\, \Delta_i x} \right]$$

and is easily extended to a hyperbolic system. The positive number "CFL" is bounded above by the appropriate value required for the stability of the chosen Runge–Kutta method in [9]. For orders of accuracy $r = 2$, 3, and 4, the upper bounds on the values of the CFL coefficients are 1.0, 1.0, and 0.87, respectively.

Regarding the cost of high-order finite-volume schemes, it should be noted that the use of these algorithms will be more costly than a correspondingly high-order finite-difference algorithm. The lower cost of a finite-difference scheme is largely due to the fact that it never involves cell averages. When pointwise values are used, the formal order of accuracy can be achieved with spatial operators of a summation type, thereby requiring less flow-field information for a flux computation. However, it is expected that a finite-volume scheme may be capable of better resolution of multi-dimensional flow features, due specifically to the truly multi-dimensional stencil of information which is utilized, as discussed in Section 4, regarding Fig. 1.

Generally speaking, grid conservation is a well-known advantage of finite-volume schemes. However, it should be noted that the curvilinear extension presented in Section 6, which utilizes the exact mesh quantities of a known transformation, will not exactly preserve a uniform flow, but will preserve it to high-order. This high-order truncation error is due to the reconstruction (27) and the Gaussian quadrature in the flux integration (28). However, if, for a given problem, exact freestream preservation is deemed essential, then it can be accomplished in the following way. Although we assume that a smooth transformation (25a) is known, we do not use it to compute the necessary metric quantities $\{x_\xi, y_\xi, x_\eta, y_\eta\}$, but instead we approximate them to high-order by the polynomial interpolation procedure discussed in the latter portion of Section 6. In addition, the pointwise distribution of the Jacobian determinant is approximated by the procedure (29), using either exact cell areas or high-order accurate approximations. This will cause exact cancellation of the Jacobian determinant in the reconstruction (27), when $u(\xi, \eta)$ is a constant, leaving only the integration of the outward face normal on the boundary of $\mathcal{D}_{ij}$ in (28). Having locally approximated the dependence of x and y on ξ and η by polynomials of degree $r-1$ on each face, the Gaussian quadrature in (28) will be exact, thereby ensuring that the discrete contour integral vanishes. In addition to the exact preservation of freestream, in this case of local approximation of a known smooth transformation, our curvilinear scheme retains its formal high-order accuracy as well.

As for boundary conditions, the finite-volume schemes appear to have an advantage, if only in the philosophy of their implementation. The application of boundary conditions in a finite-difference scheme usually requires a modification of the governing equations and/or extrapolation from the interior of the computational domain to the exterior. Boundary treatments in the finite-volume setting are less subjective, in the sense that the calculation of a flux across a cell face, which is coincident with a boundary of the computational domain, is fundamentally no different from that of any other flux within the domain. The solution of the Riemann problem is determined by two solution states on the boundary and the problem description there. One of the required boundary states arises naturally from the reconstruction procedure on the interior. The exterior value is obtained from knowledge of the flow physics on the boundary.

8. NUMERICAL EXPERIMENTS

In the following, we present a few examples of numerically computed solutions using the scheme (12) as well as its extension to curvilinear coordinates. We have performed grid refinement studies on scalar equations and the Euler equations of gas dynamics, in order to test for the computational accuracy of our scheme. We also present the numerical solution of a physical problem, for which experimental results are available for qualitative and quantitative comparison.

EXAMPLE 1. In order to test our high-order ENO scheme (12) for its accuracy, we solve the two-dimensional linear advection equation

$$u_t + u_x + u_y = 0, \qquad t > 0, \tag{30a}$$

with initial data

$$u(x, y, 0) = \tfrac{1}{2}\cos \pi(x + y) + \tfrac{1}{2}. \tag{30b}$$

The solution of (30) is two-periodic in x and y for all time. By restricting our computational domain to $-1 \leqslant x, y \leqslant 1$, we thereby make the boundaries two-periodic also, effectively removing them from consideration. We note here that even though we are solving a linear equation, the high-order scheme (12) applied to (30a) is still non-linear, due to the adaptive stencil algorithm of the reconstruction operator R^2.

The exact solution of (30) is

$$u(x, y, t) = \tfrac{1}{2}\cos \pi(x + y - 2t) + \tfrac{1}{2}.$$

This solution is truly two-dimensional, since its "wave front" is at an angle of 45° to the x and y axes. We further emphasize the two-dimensionality of the numerical solution by discretizing the computational domain so that $\Delta x \neq \Delta y$.

Since the solution is smooth for all time, we apply our scheme for one period in time, i.e., to time $t = 2.0$, at a CFL of $\frac{2}{3}$. The number of iterations required to do this on a given grid is high enough to expect a significant accumulation of

TABLE I

Solution Error—Scalar Advection—3rd Order

Grid	$\|e\|_\infty$	r_c	$\|e\|_1$	r_c
8×12	1.559 E − 1		9.725 E − 2	
16×24	2.331 E − 2	2.74	1.420 E − 2	2.78
32×48	3.002 E − 3	2.97	1.907 E − 3	2.90
64×96	3.749 E − 4	3.00	2.283 E − 4	3.06
128×192	4.666 E − 5	3.01	2.855 E − 5	3.00

TABLE III

Solution Error—Burgers Equation—3rd Order

Grid	$\|e\|_\infty$	r_c	$\|e\|_1$	r_c
8×12	3.765 E − 2		1.235 E − 2	
16×24	9.549 E − 3	1.98	2.165 E − 3	2.51
32×48	2.111 E − 3	2.18	3.093 E − 4	2.81
64×96	4.264 E − 4	2.31	4.610 E − 5	2.75
128×192	8.988 E − 5	2.25	6.958 E − 6	2.73

error. We have performed this test on five consecutively refined meshes for the scheme (12) with orders of accuracy $r = 1$, 2, 3, and 4. Though good results were obtained in all four cases, we are particularly interested in the "higher-order" cases $r = 3$ and $r = 4$. These errors are presented in Tables I and II and are calculated with respect to the L_∞ and L_1 norms. The values labeled "r_c" are the "computational orders of accuracy." These values are computed by assuming a linear accumulation of error between two successive meshes. The exact and computational solutions were compared by cell-centered pointwise output. We see "one less order" of accuracy in the L_∞ norm for $r = 4$. We therefore expect that there are points in the solution where the reconstruction stencil is discontinuous, as referred to in the closing paragraph of Section 4. However, the third-order error for this particular problem is uniform.

EXAMPLE 2. We now test a nonlinear equation, namely the two-dimensional Burgers equation

$$u_t + (\tfrac{1}{2}u^2)_x + (\tfrac{1}{2}u^2)_y = 0, \qquad t > 0, \tag{31a}$$

again with initial data

$$u(x, y, 0) = \tfrac{1}{2}\cos\pi(x + y) + \tfrac{1}{2}. \tag{31b}$$

We solve the IVP (31) on the same domain as the previous example and again apply periodic boundary conditions. In this case, due to the non-linearity of Eq. (31a), gradients immediately begin to steepen upon the first time step, until a shock eventually forms at time $t = 1/\pi$. We therefore apply the scheme, using a CFL of 0.75, up to $t = 0.15$, when the solution remains smooth. Tables III and IV illustrate the accumulated errors for this test case for $r = 3$ and $r = 4$. The exact solution is computed by using Newton–Raphson iterations to solve the characteristic relation

$$u(x, y, t) = \tfrac{1}{2}\cos\pi(x + y - 2ut) + \tfrac{1}{2}.$$

In this case, we do notice the "drop" in the order of accuracy, with respect to the L_∞ norm, in the third-order case. The solution was then computed to and past the point of shock formation, with no visible oscillatory behavior near the discontinuity.

EXAMPLE 3. In order to validate our extensions to hyperbolic systems and curvilinear coordinates, our final grid refinement study involves the solution of the two-dimensional Euler equations of gas dynamics, which we write in the form

$$U_t + F(U)_x + G(U)_y = 0, \tag{32a}$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \qquad F(U) = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ (\rho E + P)u \end{bmatrix}, \qquad G(U) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ (\rho E + P)v \end{bmatrix}. \tag{32b}$$

TABLE II

Solution Error—Scalar Advection—4th Order

Grid	$\|e\|_\infty$	r_c	$\|e\|_1$	r_c
8×12	8.764 E − 2		5.195 E − 2	
16×24	1.021 E − 2	3.10	4.013 E − 3	3.69
32×48	1.239 E − 3	3.04	4.410 E − 4	3.19
64×96	1.390 E − 4	3.16	3.093 E − 5	3.83
128×192	1.544 E − 5	3.17	2.165 E − 6	3.84

TABLE IV

Solution Error—Burgers Equation—4th Order

Grid	$\|e\|_\infty$	r_c	$\|e\|_1$	r_c
8×12	2.632 E − 2		7.525 E − 3	
16×24	4.373 E − 3	2.59	9.585 E − 4	2.97
32×48	4.192 E − 4	3.38	8.113 E − 5	3.56
64×96	4.374 E − 5	3.26	6.243 E − 6	3.70
128×192	4.361 E − 6	3.33	4.920 E − 7	3.67

Here ρ, P, and E are the density, pressure, and total specific energy, respectively, and u and v are the Cartesian components of the velocity vector $\mathbf{V}$. We close the system (32a)–(32b) of four equations with the polytropic equation of state

$$P = (\gamma - 1)\,\rho(E - \tfrac{1}{2}V^2), \tag{32c}$$

where γ is the ratio of specific heats.

The problem we solve is that of entropy advection. The pressure and velocity fields are uniform, with an initial density distribution given by

$$\rho(x, y, 0) = 1 - \frac{\alpha}{2} + \alpha[H(x)\,H(y)]^2, \tag{33a}$$

where

$$H(x) = \frac{1}{2}\left[1 + \cos\left(\pi \sin\left(\frac{\pi x}{2}\right)\right)\right] \tag{33b}$$

and α is a parameter. As a time-dependent solution of (32), the density distribution (33) is two-periodic for all time and convects with the uniform velocity. We choose to solve this problem on a curvilinear mesh, and our transformation is given by

$$x = \xi - \frac{1}{4\pi}\sin(\pi(\xi - \eta)), \tag{34a}$$

$$y = \eta - \frac{1}{4\pi}\sin(\pi(\eta - \xi)). \tag{34b}$$

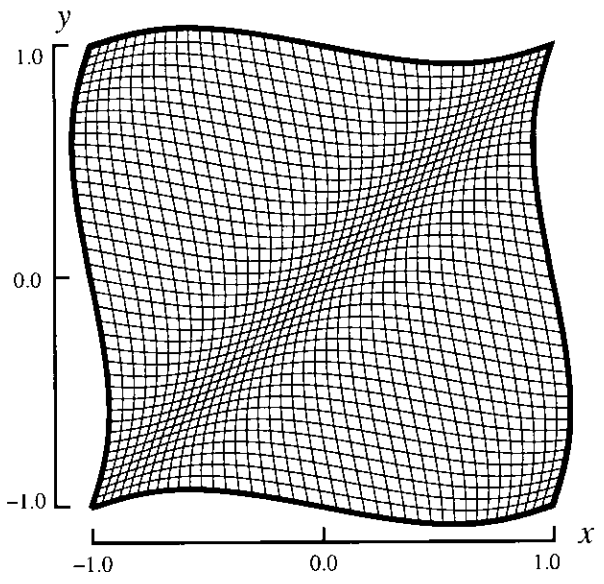

FIG. 2. Curvilinear mesh determined by the transformation (34).

TABLE V

Solution Error—Euler Advection—3rd Order

Grid	$\|e\|_\infty$	r_c	$\|e\|_1$	r_c
8×8	9.543 E − 2		3.098 E − 2	
16×16	1.706 E − 2	2.48	5.673 E − 3	2.45
32×32	2.132 E − 3	3.00	7.651 E − 4	2.89
64×64	2.999 E − 4	2.83	9.806 E − 5	2.96
128×128	5.486 E − 5	2.45	1.236 E − 5	2.99
256×256	1.559 E − 5	1.82	6.958 E − 6	3.00

Due to the fact that the mixed second derivatives of this mapping are non-trivial, there is significant skewness in the grid generated by this transformation, particularly along the diagonal, as shown in Fig. 2.

We begin with the density distribution (33), with $\alpha = 0.01$, to perturb a Mach 1.0 flow, which we orient along the mesh diagonal. Because the $x - y$ coordinates determined by (34) are two-periodic, as is the solution of (32)–(33), we restrict our computational domain to the square $[-1 \leqslant \xi \leqslant 1] \times [-1 \leqslant \eta \leqslant 1]$ and apply periodic boundary conditions.

In its rigorous form, the scheme (12) requires the exact solution of the Riemann problem, in order to determine the fluxes (12c)–(12d). In applying the scheme to the solution of the Euler equations, we approximate this step by using the approximate Riemann solver developed by Roe in [13]. We compute the solution for one period in time, at a CFL of 0.8, and compare with the exact solution. The solution errors for this grid refinement study are shown in Tables V and VI, for third- and fourth-order applications of the scheme (12). The resulting computational orders r_c for the L_1 error are comparable to those of Examples 1 and 2, which were calculated on a rectangular mesh.

EXAMPLE 4. We turn now to a more physically pertinent problem. Again, we solve the two-dimensional Euler equations of gas dynamics, this time as the solution pertains to the reflection of a moving shock wave from an inclined wall. The self-similar nature of such a solution lends itself to a rigorous analysis which is well documented in the literature (e.g., [14]), and we therefore omit any general

TABLE VI

Solution Error—Euler Advection—4th Order

Grid	$\|e\|_\infty$	r_c	$\|e\|_1$	r_c
8×8	5.782 E − 2		1.818 E − 2	
16×16	7.481 E − 3	2.95	1.582 E − 3	3.53
32×32	9.377 E − 4	3.00	1.857 E − 4	3.09
64×64	1.082 E − 4	3.12	1.400 E − 5	3.73
128×128	1.209 E − 5	3.16	1.011 E − 6	3.79
256×256	1.338 E − 6	3.18	7.105 E − 8	3.83

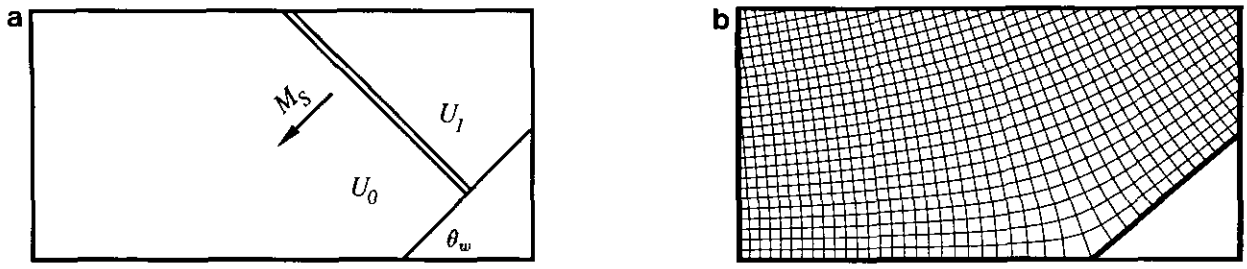

FIG. 3. (a) Initial conditions for the Mach reflection problem. (b) Computational mesh for the Mach reflection problem.

discussion of this phenomenon. These oblique shock reflections have been the subject of extensive experimental and computational research and the interested reader is referred to [15–17] and the references therein.

Our problem begins with a plane shock, whose Mach number we denote by M_S, which is moving into still air towards a wall inclined by an angle θ_w to the direction of the shock's motion, as shown in Fig. 3a. This flow orientation is chosen to facilitate comparison with existing experimental interferograms in [18]. The problem becomes two-dimensional when the shock encounters the wall and forms a reflection whose structure can be quite complex. We examine a case which results in a double Mach reflection. The wall angle θ_w is 40° and the shock Mach number is 3.72. This type of reflection exhibits a complex structure containing shock diffractions and slip lines, and it is particularly demanding of any computational algorithm.

In addition to the demanding nature of the solution itself,

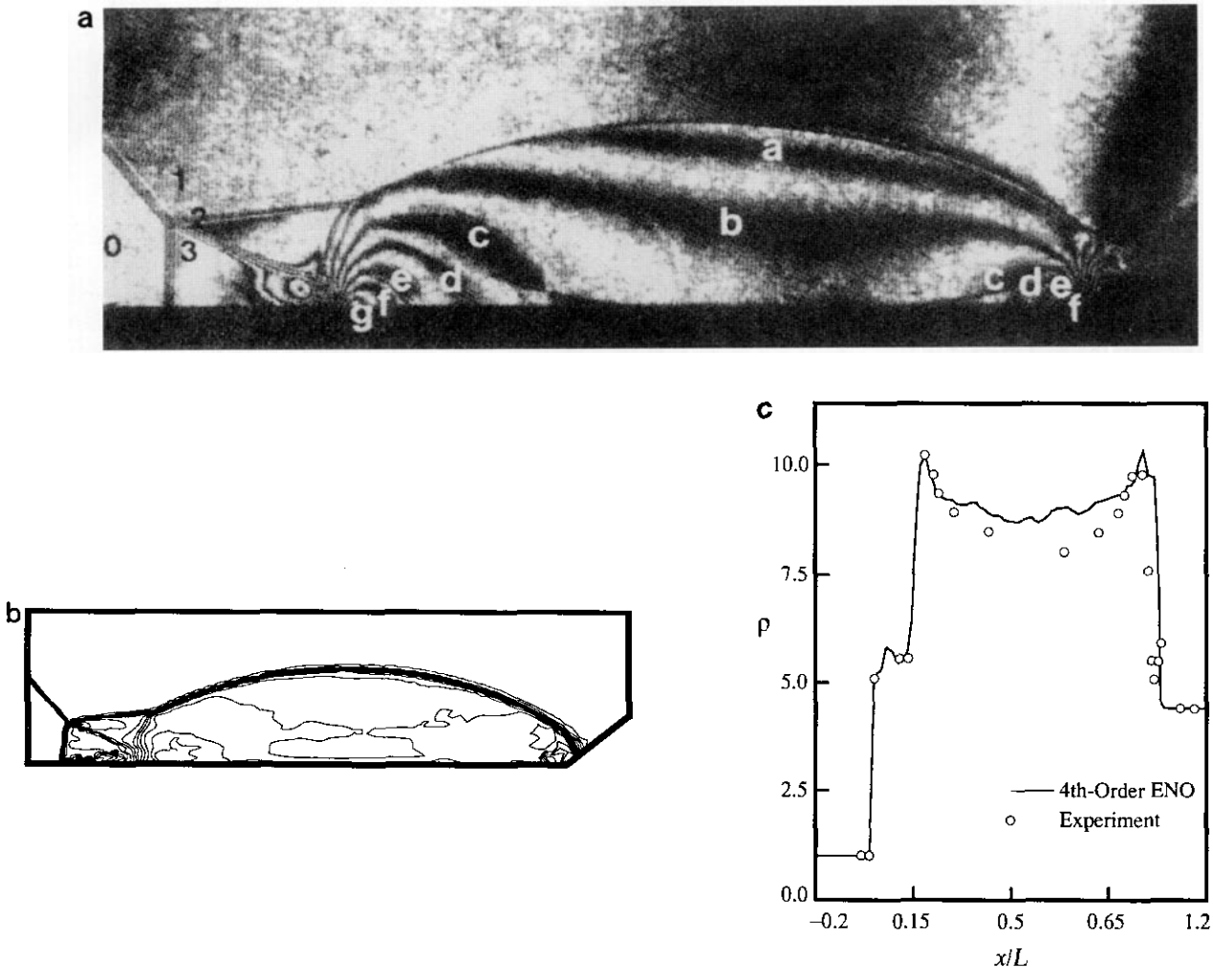

FIG. 4. (a) Experimental isopynics (Ref. [18]). (b) Density contours—fourth-order ENO. (c) Wall density (experimental data from Ref. [18]).

this problem is also made difficult by geometric concerns, largely due to the presence of a sharp corner. Although there is no way to rid ourselves of the corner itself, we attempt to mitigate its presence by using a curvilinear grid transformation. A portion of our particular mesh is shown in Fig. 3b, and it is generated by a Schwarz–Christoffel transformation. We could use this transformation to derive all of the necessary grid quantities referred to in Section 6. However, we would like to test our scheme in its most general form. Therefore, given a collection of points generated by this transformation, we calculate all of our mesh variables from the approximate grid lines we generate by polynomial interpolation.

Our initial conditions are the two constant states U_0 and U_1 which determine the desired plane shock, oriented as in Fig. 3a. These conditions are normalized with respect to the still-air initial state U_0 and are conservatively interpolated onto the computational mesh. This initial shock is then allowed to propagate through the flow field in a time-accurate manner. Provided we ensure that the entire reflection remains within the limits of the grid, the boundary conditions are relatively simple. The only concern at the far-field is adequately accounting for the movement of the plane shock. On the wall, a condition of tangency is imposed. We apply our scheme, in its curvilinear version, using Roe's approximate Riemann solver, to this test case for 400 time steps using a CFL of 0.8 on a 180×40 grid.

Figure 4a is an interferogram, from [18], resulting from an experiment designed to photographically exhibit the density structure for this test case. (The alphabetical labeling of this picture is not relevant to our presentation.) Figure 4b depicts density contours for a fourth-order numerical solution. We note good qualitative agreement with the experiment. Although the solution contains two triple points, the characteristic reconstruction procedure of Section 5 appears more than adequate for the resolution of this complicated shock structure, without oscillations.

In Fig. 4c, we plot the fourth-order density solution on the wall in order to make a comparison with experimental measurements in [18]. The x-axis is scaled by the distance L from the incident Mach stem on the wall to the corner, the Mach stem location being $x = 0$, and the corner at $x = 1$. Overall agreement with the experimental data is as good if not better than a similar comparison in [18], in which a second-order solution was computed on a 420×100 grid.

9. CONCLUDING REMARKS

A finite-volume high-order shock-capturing scheme has been developed for the computation of solutions of two-dimensional hyperbolic systems. In particular, the finite-volume schemes in [6] have been extended to the two-dimensional Euler equations of gas dynamics for applications on rectangular and smoothly varying grids. Third- and fourth-order algorithms have been numerically validated for accuracy and successfully applied to a physical problem. The usefulness of shock-capturing schemes which are of formal high-order accuracy in more than one spatial dimension appears certain. The results in Example 3 suggest that these types of schemes may be useful in scientific areas where the time-accurate resolution of weak waves is crucial, such as in the area of aeroacoustics. Example 4, on the other hand, illustrates the capability of a high-order algorithm to resolve a complicated shock structure, without oscillatory behavior. Such schemes may provide a promising alternative to spectral methods, in the study of aeroacoustic and transition related problems, in which shocks or complicated geometries are involved.

REFERENCES

1. S. K. Godunov, *Mat. Sb.* **47**, 271 (1959).
2. B. van Leer, *J. Comput. Phys.* **32**, 101 (1979).
3. A. Harten, *J. Comput. Phys.* **49**, 357 (1983).
4. A. Harten, *SIAM J. Numer. Anal.* **21**, 1 (1984).
5. A. Harten and S. Osher, MRC Technical Summary Report No. 2823, 1985 (unpublished).
6. A. Harten, B. Engquist, S. Osher, and S. Chakravarthy, *J. Comput. Phys.* **71**, 231 (1987).
7. P. L. Roe, "Some Contributions to the Modelling of Discontinuous Flows," in *Large Scale Computations in Fluid Mechanics*, Lectures in Appl. Math., Vol. 22-2 (Amer. Math. Soc., Providence, RI, 1985).
8. A. Harten, "Preliminary Results on the Extension of ENO Schemes to Two-Dimensional Problems," in *Proceedings, International Conference on Nonlinear Hyperbolic Problems, Saint-Etienne, 1986*, Lecture Notes in Mathematics, edited by C. Carasso *et al.* (Springer-Verlag, Berlin, 1987), p. 23.
9. C. Shu and S. Osher, *J. Comput. Phys.* **77**, 439 (1988).
10. C. Shu and S. Osher, *J. Comput. Phys.* **83**, 32 (1989).
11. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* (Springer-Verlag, New York, 1980).
12. P. Woodward and P. Colella, *J. Comput. Phys.* **54**, 115 (1984).
13. P. L. Roe, *J. Comput. Phys.* **43**, 357 (1981).
14. D. H. Jones, P. M. Martin, and E. Thornhill, *Proc. R. Soc. London A* **209**, 238 (1951).
15. I. I. Glass, *AIAA J.* **25** (2), 214 (1987).
16. H. M. Glaz, P. Colella, I. I. Glass, and R. L. Deschambault, *Proc. R. Soc. London A* **398**, 117 (1985).
17. V. Young and H. Yee, AIAA-87-0112, AIAA 25th Aerospace Sciences Meeting, January 1987 (unpublished).
18. H. M. Glaz, P. Colella, I. I. Glass, and R. L. Deschambault, "A Detailed Numerical, Graphical, and Experimental Study of Oblique Shock Wave Reflections," UTIAS Report No. 285, Institute for Aerospace Studies, University of Toronto, 1986 (unpublished).