DEPARTMENT OF MATHEMATICAL SCIENCES
COLLEGE OF SCIENCES
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529

P-39

FINITE-VOLUME APPLICATION OF HIGH ORDER ENO SCHEMES TO TWO-DIMENSIONAL BOUNDARY-VALUE PROBLEMS

By

J. Mark Dorrepaal, Principal Investigator

and

Jay Casper, Graduate Research Assistant

Final Report
For the period ended August 15, 1990

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Master Contract Agreement NAS1-18584
Task Authorization No. 56
James D. Keller, Technical Monitor
FLDMD-Theoretical Aerodynamics Branch

August 1990

DEPARTMENT OF MATHEMATICAL SCIENCES
COLLEGE OF SCIENCES
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529

FINITE-VOLUME APPLICATION OF HIGH ORDER ENO SCHEMES TO TWO-DIMENSIONAL BOUNDARY-VALUE PROBLEMS

By

J. Mark Dorrepaal, Principal Investigator

and

Jay Casper, Graduate Research Assistant

Final Report
For the period ended August 15, 1990

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Master Contract Agreement NAS1-18584
Task Authorization No. 56
James D. Keller, Technical Monitor
FLDMD-Theoretical Aerodynamics Branch

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, Virginia 23508-0369

August 1990

ABSTRACT

Finite–Volume Application of High Order ENO Schemes
to Multi–Dimensional Boundary–Value Problems

Jay Casper
Department of Mathematics
Old Dominion University
Norfolk, Virginia

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 to 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 that provides a time update of the solution which is of high-order accuracy, wherever the function is smooth, and avoids a Gibbs phenomenon at discontinuities.

The application of these schemes to areas of scientific and industrial interest obviously requires compressible flow solutions in more than one spatial dimension. The multi–dimensional ENO schemes proposed thus far in the literature are of finite–difference type, combining an adaptive high–order ENO interpolation procedure with a split–flux approach. Due to the design of these schemes, it is not clear, at present, how to apply them at boundaries, *e.g.* solid walls. Moreover, as with any finite–difference scheme in which stencils are not pre–determined, curvature and discontinuity of the computational grid can pose fundamental problems.

In this paper, we consider the finite–volume approach in developing multi–dimensional, high–order accurate ENO schemes. In particular, a two-dimensional extension is proposed for the Euler equations of gas dynamics. This requires a spatial reconstruction operator that attains formal high order of accuracy in two dimensions by taking account of cross gradients. Given a set of cell averages in two spatial variables, polynomial interpolation of a two-dimensional primitive function is employed in order to extract high-order pointwise values on cell interfaces. These points are appropriately chosen so that correspondingly high-order flux integrals are obtained through each interface by quadrature, at each point having calculated a flux contribution in an upwind fashion. The solution-in-the-small of Riemann's IVP that is required for this pointwise flux computation is achieved using Roe's approximate Riemann solver. Issues to be considered in this two-dimensional extension include the implementation of boundary conditions and application to general curvilinear coordinates.

Results of numerical experiments are presented for qualitative and quantitative examination. These results contain the first successful application of ENO schemes to boundary–value problems with solid walls.

FINITE-VOLUME APPLICATION OF HIGH ORDER ENO SCHEMES TO TWO-DIMENSIONAL BOUNDARY-VALUE PROBLEMS

Jay Casper

Department of Mathematics
Old Dominion University
Norfolk, Virginia

1. INTRODUCTION

We wish to design high-order accurate essentially non–oscillatory (ENO) schemes for the numerical approximation of weak solutions of hyperbolic systems 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 $u = (u^1, u^2, \ldots, u^m)^T$ 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 (1) is well-posed in the sense that the solution u depends continuously on the initial data, and that this solution is generically 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

$$x_{i-1/2} < x < x_{i+1/2}\,, \qquad y_{j-1/2} < y < y_{j+1/2}\,, \qquad -\infty < i,j < \infty\,,$$

denote a rectangular partition of the $x-y$ plane, with x_i and y_j denoting the centroids of their respective intervals. With a semi–discrete formulation in mind, we note that, for every every rectangle $(x_{i-1/2}, x_{i+1/2}) \times (y_{j-1/2}, y_{j+1/2})$, a weak solution of (1a) must satisfy

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

where a_{ij} is the area of the rectangle and

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

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

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

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

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 the unknown u, and we rewrite this equation, for fixed t, in the abstract operator–product form

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

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

In [8,9], Runge–Kutta methods are presented for the time discretization of ordinary differential equations which are of high–order accuracy and 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 in order to achieve our desired accuracy in time. The use of

the formulation (2a), rather than the finite–difference approach of [8,9], makes for a more immediate application of its numerical approximation to the solutions of boundary–value problems involving solid walls or non–trivial geometries.

2. Two–Dimensional ENO Schemes

Given $\{\bar{u}^n\}$, 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 r^{th}–order accurate in the sense of local truncation error, *i.e.* when applied to the exact solution, E_h satisfies

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

in a pointwise sense, wherever u is sufficiently smooth, with Δx, Δy, and Δt all assumed to be $O(h)$. 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*** (ENO). Such an operator satisfies, in the one–dimensional scalar case,

$$TV(\,E_h\,\bar{u}\,) \;=\; TV(\,u\,) \;+\; O(h^{1+p})\,, \tag{5}$$

for some $p > 0$, where TV represents total variation in x. For details concerning the initial development of ENO schemes, the reader is referred to [5] and the references therein.

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

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

$$\bar{u}_{ij}^{(0)} \;=\; \bar{u}_{ij}^{n}\,, \qquad\qquad \bar{u}_{ij}^{(p)} \;=\; \bar{u}_{ij}^{n+1}\;.$$

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 [8,9] for details.)

Now, if we assume that the scheme (6) achieves our desired r^{th}–order accuracy in time, then clearly, this scheme satisfies (4) if we can evaluate $(\,\mathcal{L}\,u(t)\,)_{ij}$, the exact spatial operation on the right–hand side of (2a). However, the calculation of the fluxes (2c–d) needed for this evaluation requires that we know the solution $u(x,y,t)$, pointwise, at a

given time t, whereas the information we have at any time t is that of the cell averages (2b). And clearly, since

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

wherever u is smooth, there is an inherent limit on the order of accuracy if we use the cell averages themselves in the flux calculation. Therefore, we replace the operator $\mathcal{L}$ with a discrete spatial operator L which acts upon the $\{\bar{u}(t)\}$ and approximates the pointwise operation of $\mathcal{L}$ to high order. To this end, we see that if

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

wherever u is smooth, then, upon replacement of $\mathcal{L}$ in (5), the local truncation error of our fully discrete scheme will satisfy (4), as required.

We define L explicitly by

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

where $\bar{f}$ and $\bar{g}$ are to be designed as high–order numerical approximations to $\hat{f}$ and $\hat{g}$ in (2c–d). We require that the numerical flux functions $\bar{f}$, $\bar{g}$

$$\begin{aligned}
\bar{f}_{i+1/2\,,j}(t) \;&=\; \bar{f}(\bar{u}_{i-k+1\,,j-m+1}(t), \ldots, \bar{u}_{i+k\,,j+m}(t))\;, \quad 1 \le k, m \le r\;, \\
\bar{g}_{i\,,j+1/2}(t) \;&=\; \bar{g}(\bar{u}_{i-l+1\,,j-n+1}(t), \ldots, \bar{u}_{i+l\,,j+n}(t))\;, \quad 1 \le l, n \le r\;,
\end{aligned} \tag{7c}$$

be Lipschitz continuous functions of their arguments and consistent with the true fluxes $f(u)$, $g(u)$ in the sense

$$\bar{f}(w, \ldots, w) \;=\; f(w)\;, \qquad \bar{g}(w, \ldots, w) \;=\; g(w)\;. \tag{7d}$$

The first and most important step in the high–order approximations of $\hat{f}$, $\hat{g}$ is the method by which we obtain high–order accurate pointwise information of the solution $u(x, y, t)$ from the given set of cell averages $\{\bar{u}(t)\}$. Let R^2 be a spatial operator which reconstructs this set of cell averages and yields a two–dimensional, piecewise polynomial $R^2(x, y\,; \bar{u}(t))$ 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

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

In general, the "reconstruction" operator R^2 which we have developed is a "natural" two–dimensional extension of the one–dimensional "reconstruction by primitive" presented in

[5]. The reconstruction procedure involves polynomial interpolation in combination with an adaptive stencil algorithm which, in tandem, achieve high–order accuracy in smooth regions while avoiding $O(1)$ oscillations near steep gradients. This operator is detailed in Section 3.

In order to include the more general case, where $f(u)$ cannot be integrated in closed form in (2c), 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 integration 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.* [10]) 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), with P_K being the polynomial of degree K in the orthogonal basis that spans the space of polynomials 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 $P_K(x) = (x-x_1)(x-x_2)\ldots(x-x_K)$ are real and distinct, making it clear that the above truncation error is $O(h^{2K+1})$. Relating this error to the $(r-1)$–st degree polynomial reconstruction (8), we see that for $r \le 2K$, 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

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

for some η in $(y_{j-1/2}, y_{j+1/2})$.

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

$$v_h(x,y,t) \;=\; u(x,y,t) \;+\; e(x,y)\,h^r\;, \qquad (9b)$$

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,

$$f\big(u(x\,,\,y\,,\,t)\big) \;=\; f\big(v_h(x\,,\,y\,,\,t)\big) \;+\; d(x\,,\,y)\,h^r\;, \qquad (9c)$$

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

$$\hat{f}_{i+1/2\,,j}(t) \;=\; \frac{\Delta y_j}{2} \sum_{k=1}^{K} c_k \,[\, f(\,v_h(x_{i+1/2}\,,\,y_k\,,\,t)\,) \;+\; d(x_{i+1/2}\,,\,y_k)\,h^r\,] \qquad (9d)$$
$$+\; s(x_{i+1/2}\,,\,\eta)\,h^{r+1}\,.$$

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

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

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

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

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 10(b) satisfies

$$\hat{f}_{i+1/2\,,j}(t) - \hat{f}_{i-1/2\,,j}(t) \;=\; \bar{f}_{i+1/2\,,j}(t) - \bar{f}_{i-1/2\,,j}(t) \;+\; O(h^{r+2})\,. \qquad (10c)$$

Moreover, a symmetrical argument can be used to show

$$\hat{g}_{i\,,j+1/2}(t) - \hat{g}_{i\,,j-1/2}(t) \;=\; \bar{g}_{i\,,j+1/2}(t) - \bar{g}_{i\,,j-1/2}(t) \;+\; O(h^{r+2})\,, \qquad (10d)$$

where

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

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 $\mathcal{L}$ in (6), yields a numerical solution operator E_h which is r^{th}– order accurate in the sense of local truncation error. We note here that the desired truncation errors given by (10c–d) are achieved only if the functions representing the errors due to the quadrature (9a) and the solution approximation (8) are Lipschitz continuous on $[x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$.

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 is largely due to the nature of the reconstruction step, at which the pointwise behavior of $u(x,y,t)$ is approximated in a piecewise polynomial fashion within each cell. (See Section 3.) In particular, the piecewise polynomial generated by the reconstruction operator can be discontinuous at cell interfaces, due to an adaptive interpolation procedure. 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 (2c–d) are approximated by

$$f(\,u(x,\,y,\,t)\,) \;\approx\; f^{\mathrm{Rm}}[\,R^2(\,x-0\,,\,y\,;\,\bar{u}(t)\,)\,,\,R^2(\,x+0\,,\,y\,;\,\bar{u}(t)\,)\,]\,, \tag{11a}$$

$$g(\,u(x,\,y,\,t)\,) \;\approx\; g^{\mathrm{Rm}}[\,R^2(\,x\,,\,y-0\,;\,\bar{u}(t)\,)\,,\,R^2(\,x\,,\,y+0\,;\,\bar{u}(t)\,)\,]\,, \tag{11b}$$

where $f^{\mathrm{Rm}}[u_1,u_2]$ denotes the flux across $x=0$ of the solution to the Riemann problem whose initial states are u_1 and u_2. We use the notation $q(x+0)\,,\,q(x-0)$ to denote the limiting values of q at x from the right and left, respectively. When the solution u is sufficiently smooth, the "jumps" in the discontinuities in the approximate solution at cell interfaces will be $O(h^r)$ in which case our previous conclusions concerning high–order truncation error are not altered by the modification (11).

In fully–discrete form, we write our scheme as the Runge–Kutta method

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

$$\bar{u}_{ij}^{(0)} \;=\; \bar{u}_{ij}^{n}\,, \qquad\qquad \bar{u}_{ij}^{(p)} \;=\; \bar{u}_{ij}^{n+1}\,,$$

where

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

and

$$\bar{f}_{i+1/2\,,\,j}^{(m)} \;=\; \frac{\Delta y_j}{2}\,\sum_{k=1}^{K}\,c_k\,f^{\mathrm{Rm}}[\,R^2(x_{i+1/2}-0\,,\,y_k\,;\,\bar{u}^{(m)})\,,\,R^2(x_{i+1/2}+0\,,\,y_k\,;\,\bar{u}^{(m)})\,]\,, \tag{12c}$$

$$\bar{g}_{i,j+1/2}^{(m)} \;=\; \frac{\Delta x_i}{2}\,\sum_{k=1}^{K}\,c_k\,g^{\mathrm{Rm}}[\,R^2(x_k\,,\,y_{j+1/2}-0\,;\,\bar{u}^{(m)})\,,\,R^2(x_k\,,\,y_{i+1/2}+0\,;\,\bar{u}^{(m)})\,]\,. \tag{12d}$$

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 r^{th}–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 $\{\bar{v}_{ij}^N\}$, approximations to the cell averages of u at time t^N which satisfy

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

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

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

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.

In Section 7, we present results which support the claim of cumulative error represented by (13). We will also extend the scheme (12) to general curvilinear co–ordinates and perform numerical experiments which apply this scheme to boundary–value problems. Some of these results are shown in Section 7, and are representative of the first successful attempt to employ high–order ENO schemes to two–dimensional problems with solid walls.

3. RECONSTRUCTION

Before proceeding to a discussion of the implementation of the scheme (12) in solving boundary–value problems, we turn now to describe more fully the high–order spatial reconstruction operator R^2 which is crucial to the scheme's accuracy. For the purpose of clarity, we discuss the finer points of this procedure within the framework a scalar function defined on a rectangular computational mesh. Furthermore, as there are many ways to approach the development of this high–order reconstruction, for simplicity, we choose to describe the implementation of R^2 as a composition of two applications of a one–dimensional operator R, where the latter is the "reconstruction–by–primitive" operator in [5].

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 n^{th}–degree piecewise polynomial function of x that interpolates w at the points $\{x_i\}$, *i.e.*

$$Q_n(x_i\,; w) = w(x_i) , \tag{14a}$$

$$Q_n(x\,; w) \equiv q_{n,i}(x\,; w) , \qquad x_i \le x \le x_{i+1} , \tag{14b}$$

where $q_{n,i}$ is the (unique) n^{th}–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 chose the other $n-1$ points, and we do so subject to the condition that $w(x)$ be "smoothest" on the chosen stencil in some asymptotic sense.

It can be argued that information relevant to the smoothness of w can be extracted from a table of divided differences of w. Employing the standard notation, the k^{th} divided difference of w can be defined recursively by

$$w[x_j] \;=\; w(x_j)\;;$$

$$w[x_j,\ldots,x_{j+k}] \;=\; \frac{w[x_{j+1},\ldots,x_{j+k}] - w[x_j,\ldots,x_{j+k-1}]}{x_{j+k}-x_j}\;.$$

In particular, it can be argued that the magnitude of a k^{th} divided difference, provides an asymptotic measure of the smoothness of w in (x_j, x_{j+k}), in the following sense. Suppose that w is smooth in the interval (x_0, x_k) but is discontinuous in (x_1, x_{k+1}). Then for h sufficiently small, we expect

$$|\,w[x_0,\ldots,x_k]\,| < |\,w[x_1,\ldots,x_{k+1}]\,|\;,$$

and hence these divided differences can serve as a tool to compare the relative smoothness of w in various stencils. Since we always assume any stencil we choose to be contiguous, we can assign a particular stencil by determining its left–most index, which we denote by $j(i)$. We choose $j(i)$ with the following hierarchical algorithm.

Let $j_k(i)$ denote the left–most index of a chosen "smoothest" $(k+1)$–point stencil which includes the interval (x_i, x_{i+1}). Denote this stencil

$$\{\,x_{j_k(i)},\ldots,x_{j_k(i)+k}\,\},\qquad k=1,2,\ldots,n \tag{15a}$$

Since any stencil must include $\{x_i, x_{i+1}\}$, our recursive algorithm begins $(k=1)$ by setting

$$j_1(i) \;=\; i\;, \tag{15b}$$

In order to choose $j_{k+1}(i)\,, k=1,\ldots,n-1$, we consider as candidates the two stencils

$$\{\,x_{j_k(i)-1},\ldots,x_{j_k(i)+k}\,\}\quad\text{or}\quad\{\,x_{j_k(i)},\ldots,x_{j_k(i)+k+1}\,\}\;, \tag{15c}$$

which are obtained by adding a point to the left or right, respectively, of the previously determined stencil. We select the one in which w is relatively smoother, *i.e.* the one in which the $(k+1)$–th order divided difference is smaller in magnitude, thus

$$j_{k+1}(i) \;=\; \begin{cases} j_k(i)-1\,, & \text{if } |\,w[x_{j_k(i)-1},\ldots,x_{j_k(i)+k}]\,| \;<\; |\,w[x_{j_k(i)},\ldots,x_{j_k(i)+k+1}]\,|\;, \\ j_k(i)\,, & \text{otherwise}\;. \end{cases} \tag{15d}$$

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

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

i.e. $Q_n(x\,;\,w)$ is essentially non–oscillatory. Henceforth, we will assume that $Q_n(x\,;\,w)$ is an ENO interpolating polynomial of degree n whose construction involves the adaptive–stencil algorithm (15).

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

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

where $\Delta x_i \Delta y_j = a_{ij} = (x_{i+1/2} - x_{i-1/2})\,(y_{j+1/2} - y_{j-1/2})$. For $y_{j+1/2} < y < y_{j-1/2}$, define the *primitive function* $\bar{W}_j(x)$ associated with w by

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

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

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

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

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

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

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

This suggests that if we approximate $\bar{W}_j(x)$ by $Q_r(x\,;\bar{W}_j)$, the ENO picewise polynomial of degree r which interpolates $\bar{W}_j$ at the values given by (17c), we can then obtain an approximation of $\bar{w}_j(x)$ by defining the first step in our reconstruction procedure as

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

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

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

wherever $\bar{w}_j(x)$ is sufficiently smooth in 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 value of $\bar{w}_j(x)$ in x is equivalent in form to a one–dimensional cell average on the interval $[y_{j+1/2}, y_{j-1/2}]$. Therefore, for a fixed x, the remainder of our reconstruction procedure becomes equivalent to the one–dimensional method in [5], applied to the set $\{\bar{v}_j(x)\}$.

For 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

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

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

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

wherever $v(x,y)$ is sufficiently smooth.

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 $\{\bar{w}_{ij}\}$, *i.e.*

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

We note here that the high–order relationship (19) cannot be achieved for $r > 2$ by simply applying the operator R to two "overlapped" one–dimensional stencils.

In addition to the high–order truncation error in regions of smoothness, 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.*

$$\frac{1}{\Delta x_i \Delta y_j} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} R^2(x,y\,;\,\bar{w})\,dy\,dx \;=\; \bar{w}_{ij}\;, \tag{20}$$

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 adequate resolution of a discontinuity, near which the "jumps" in $R^2(x,y\,;\,\bar{w})$ 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 occurr 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 see that the cumulative error of our numerical scheme is $O(h^{r-1})$ in the L_∞ norm and $O(h^r)$ in the L_1 norm.

4. Systems of Conservation Laws

In this section, we extend the scalar reconstruction procedure of Section 3 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 $\{\bar{u}_{ij}\}$ of vector–valued cell averages to high–order pointwise accuracy. Clearly, in this context, $\bar{u}_{ij}$ denotes a vector whose components are the cell averages of the scalar components of

u. It would therefore seem natural, purely from an approximation theory viewpoint, to reconstruct the set $\{\bar{u}_{ij}\}$ 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, in the solution of more than one dependent variable, a component–wise 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 dimension can be found in [5]. In the following, we describe an algorithm to reconstruct the vector–valued solution u from its cell averages which attempts to avoid this difficulty.

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

$$u_t + A\,u_x + B\,u_y = 0\,, \tag{21a}$$

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

We also assume, for now, that our reconstruction takes place within the context of a Cartesian mesh. We note that the eigenvalues $\{\lambda^k(A)\}$ of A and $\{\lambda^k(B)\}$ of B are constant as are the right eigenvectors $\{r^k(A)\}$, $\{r^k(B)\}$ and left eigenvectors $\{l^k(A)\}$, $\{l^k(B)\}$. We assume that these eigenvectors are suitably normalized so that

$$l^i(A)\cdot r^j(A) = l^i(B)\cdot r^j(B) = \delta_{ij}\,. \tag{22a}$$

If we define the k^{th} characteristic variables $w^k(A)$ and $w^k(B)$ by the dot products

$$w^k(A) = l^k(A)\cdot u\,, \quad w^k(B) = l^k(B)\cdot u\,, \qquad k = 1,2,\ldots,m\,, \tag{22b}$$

then it follows from (22a) that

$$u = \sum_{k=1}^{m} w^k(A)\,r^k(A)\,. = \sum_{k=1}^{m} w^k(B)\,r^k(B)\,. \tag{22c}$$

It is argued in [5] that, in the constant–coefficient case, it makes sense 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 (21a) becomes an uncoupled set of equations, thereby rendering any discontinuity in a particular characteristic variable "undetectable" by another. (See [5] for details.)

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

$$\bar{w}^k_{ij}(A) \;=\; l^k(A)\cdot\bar{u}_{ij}\;, \qquad \text{for } j \text{ fixed,} \quad \text{all } i\;, \tag{23a}$$

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

$$\mathbf{R}(x;\bar{u}) \;=\; \sum_{k=1}^{m} R(x;\bar{w}^k(A))\,r^k(A) \;\equiv\; v_j(x)\;, \tag{23b}$$

the right–hand side of which is the vector–valued analogy of $v_j(x)$ in (17f). In analogy with the two–step procedure in Section 3, 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

$$\bar{w}^k_j(B) \;=\; l^k(B)\cdot v_j\;. \tag{23c}$$

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

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

which completes our reconstruction for the linear case and we write

$$\begin{aligned}\mathbf{R}^2(x,y;\bar{u}) \;&\equiv\; \mathbf{R}(\,y;\mathbf{R}(x;\bar{u})\,)\\ &=\; \sum_{k=1}^{m} R\left(\,y\;;\,l^k(B)\cdot\sum_{p=1}^{m} R(\,x;l^p(A)\cdot\bar{u}\,)\,r^p(A)\right)\,r^k(B)\;.\end{aligned} \tag{23e}$$

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 $\{\lambda^k(A(u))\}$,$\{\lambda^k(B(u))\}$, and the eigenvectors $\{r^k(A(u))\}$, $\{r^k(B(u))\}$, $\{l^k(A(u))\}$, and $\{l^k(B(u))\}$. Our extension will require the use of locally defined characteristic variables, in the following manner. In order to reconstruct $u(x,y)$ on the region $(x_{i-1/2},x_{i+1/2})\times(y_{j-1/2},y_{j+1/2})$, we first derive a set of local average characteristic variables $\{\bar{w}^k_{nj}(\bar{u}_{ij})\}$, where n varies in the x–direction. We do this, for a fixed j, by computing dot products of $l^k(A(\bar{u}_{ij}))$ with the cell averages $\{\bar{u}_{nj}\}$ associated with intervals in the "immediate vicinity" of $(x_{i-1/2},x_{i+1/2})$, *i.e.*

$$\bar{w}^k_{nj}(\bar{u}_{ij}) \;=\; l^k(A(\bar{u}_{ij}))\cdot\bar{u}_{nj}\;, \qquad \text{for} \quad n=i-q,\ldots,i+q\;, \tag{24a}$$

where q is the degree of the reconstruction polynomial. We then apply the scalar reconstruction operator R to this set $\{\bar{w}^k_{nj}(\bar{u}_{ij})\}$ of $2q+1$ variables in which the left eigenvector has been locally "frozen" at the i^{th} location of the j^{th} row of cells. This "local linearization" allows us to apply locally the linear vector reconstruction described in (23). The first step in our "nonlinear" vector reconstruction procedure then becomes

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

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

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

We complete our "nonlinear" reconstruction by applying the scalar operator R to the set $\{\bar{w}^k_n(\bar{v}_j)\}$ of local variables in (24c) which results in

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

Thus, our vector reconstruction operator $\mathbf{R}^2$ is a composition of (24b) and (24d) and, for a polynomial within a cell (i,j), we write *abstractly*

$$\mathbf{R}^2(x,y;\,\bar{u})_{ij} = \sum_{k=1}^{m} R\left(\;y\;;\,l^k(B(\bar{v}_j(x))\cdot\sum_{p=1}^{m} R(\,x\,;l^p(A(\bar{u}_{ij}))\cdot\bar{u}\,)\,r^p(A(\bar{u}_{ij}))\right)\,r^k(B(\bar{v}_j(x))\;, \tag{24e}$$

and satisfies

$$\mathbf{R}^2(\,x,y;\,\bar{u}(t)\,) \;=\; u(x,y,t)\;+\;O(h^r)\;, \tag{24f}$$

wherever u is sufficiently smooth, and which we hope will avoid the oscillatory behavior associated with colliding discontinuities.

5. Curvilinear Co–ordinates

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

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

denote a smooth transformation from the physical x–y plane to the rectangular ξ–η plane, with Jacobian determinant J given by

$$J \;=\; x_\xi\,y_\eta \;-\; y_\xi\,x_\eta\;. \tag{25b}$$

Let C_{ij} denote a discrete region in physical space which is mapped into a rectangle $(\xi_{i-1/2}, \xi_{i+1/2}) \times (\eta_{j-1/2}, \eta_{j+1/2})$ 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) :

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

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

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

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

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

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

where

$$\tilde{F}(u) \;=\; y_\eta\, f(u) - x_\eta\, g(u) \;, \qquad \tilde{G}(u) \;=\; x_\xi\, g(u) - y_\xi\, f(u) \;, \tag{26e}$$

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

Having defined the necessary terms of our finite–volume formulation in a curvilinear co–ordinate system, we now wish to discretize our spatial and temporal operations. Again, with our Runge–Kutta time discretization, the extension is straight forward. 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 $\bar{u}_{ij}$, as defined by (26b), we see that the quantity $a_{ij}\,\bar{u}_{ij}$ can be interpreted as a "cell average" of the function $u(\xi,\eta)\,J(\xi,\eta)$ on the corresponding computational cell, having unit area. Therefore if we choose, instead, to reconstruct the set $\{a_{ij}\,\bar{u}_{ij}\}$ of cell averages which are "scaled" by their respective areas, we transform the reconstruction procedure to rectangular co–ordinates, and can simply use the procedures outlined in Sections 3 and 4. In the case of vector reconstruction (Section 4), 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 polynomial $p(\xi,\eta)$ we construct by this approach will be a high–order approximation to the function $u(\xi,\eta)\,J(\xi,\eta)$, and therefore must be "re–scaled" by $J(\xi,\eta)$ in order to

yield the pointwise values we need to approximate the fluxes (26c–d). Thus, we define our curvilinear reconstruction operator $\tilde{R}^2$ by

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

with the vector analogy $\tilde{\mathbf{R}}^2$ defined in an identical manner.

As for our second modification, we note that when we desire accuracy that is higher than second–order, a piecewise linear interpretation of a two–dimensional grid will no longer 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–d), we can no longer assume that the quantities $\{x_\xi, x_\eta, y_\xi, y_\eta\}$ 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. It should also be noted that the curvature of cell boundaries also affects the lengths of cell interfaces, which become the curvilinear analogy of Δy_j , Δx_i in (12c–d). Our numerical flux can then be written

$$\bar{f}_{i+1/2\,,j}(t) \;=\; \frac{|\Delta\eta|_{i+1/2\,,j}}{2}\sum_{k=1}^{K} c_k\,\tilde{F}^{\mathrm{Rm}}\big[\;\tilde{R}^2(\xi_{i+1/2}-0\,,\,\eta_k\,;\,\bar{u}(t))\,,\;\tilde{R}^2(\xi_{i+1/2}+0\,,\,\eta_k\,;\,\bar{u}(t))\;\big]\,, \tag{28}$$

where $\tilde{R}^2$ is given by (27), and $|\Delta\eta|_{i+1/2\,,j}$ represents the arclength of the of the boundary of C_{ij} along $\xi=\xi_{i+1/2}$. The description of $\hat{g}_{i\,,j+1/2}(t)$ follows from symmetry. 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 r^{th}–order accurate as defined by (4) in smooth regions and 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 equations (25a) from which to determine all the grid quantities necessary for the flux computation (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 is approximated by polynomial interpolation, and all the necessary mesh quantities are calculated from these polynomials. As the grid metrics $\{x_\xi, x_\eta, y_\xi, y_\eta\}$ represent the components of outward normals at cell boundaries, a simple calculus argument will enable us to compute these quantities at the desired quadrature points. Cell areas $\{a_{ij}\}$ and the arclengths of cell faces $\{|\Delta\eta|_{i+1/2\,,j}, |\Delta\xi|_{i\,,j+1/2}\}$ are also straight forward, once the four polynomials defining each

control volume are determined. An approximate Jacobian determinant, which we will denote by J', required for the curvilinear reconstruction procedure (27) 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. Thus, having first determined approximations to the cell areas by polynomial interpolation of the grid points, we apply our scalar reconstruction algorithm to this set and define J' by

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

We note here 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. We do, however, "test" this approach by implementing it in a numerical experiment rather than applying the exact grid quantities from the known transformation. The results of this test case are reported in Section 7.

6. Implementation

Before proceeding to numerical test cases, we find it prudent to make a few remarks concerning the implementation of some of the ideas in the preceding sections. As these notions are presented in a general setting, their implementation in a given problem is subject to individual interpretation and the suggestions made in this section are those which are employed by the author in the calculations presented in the next section.

Our first topic concerns the first step in our scheme, namely the reconstruction procedure. Though it is not necessary from an approximation theory viewpoint, we will implement the reconstruction in two separate stages, depending upon the orientation of the cell boundary along which we desire to approximate u. For the evaluation of the flux $\hat{f}_{i+1/2\,,j}(t)$ in (12c), the pointwise approximation of u at the Gauss points $\{y_k\}$ along $x = x_{i+1/2}$ is determined by

$$u(x_{i+1/2} \pm 0, y_k, t) \;\approx\; R(\,y_k\,;\,R(\,x_{i+1/2} \pm 0\,;\bar{u}(t))\,)\;, \tag{29a}$$

and for the evaluation of $\hat{f}_{i+1/2\,,j}(t)$ in (12d), we employ the composite operation (29a) in "reversed" form and achieve the required pointwise values by

$$u(x_k, y_{j+1/2} \pm 0, t) \;\approx\; R(\,x_k\,;\,R(\,y_{j+1/2} \pm 0\,;\bar{u}(t))\,)\;. \tag{29b}$$

Also pertaining to the reconstruction, we have taken the position that a high–order approximation must reflect available information. Therefore, near boundaries, we restrict our interpolation stencil to remain within the computational domain. It was found during numerical experimentation that problems could result from a high–order, one–sided interpolation procedure. In particular, if a solution developed a shock which reflected from a solid wall, oscillatory behavior was noticed in the smooth region near the reflection point between the shock and the wall. Attempting to eliminate such numerical noise, one might suggest some sort of test for a "desirable reconstruction" which, when failed, will result in a local reduction in the order of interpolation. Owing to the recursive nature of the Newton interpolation procedure, such a test can be readily applied during the actual "building" of the polynomial. One such test simply checks for "over–shoots" or "under–shoots." This is equivalent to requiring the endpoint values of the polynomial $p_i(x)$ which approximates $u(x)$ on $[x_{i-1/2}, x_{i+1/2}]$ not to over–shoot a larger adjacent cell average, or to under–shoot a smaller one. We can so restrict $p_i(x)$ by requiring that it satisfy

$$(\bar{u}_{i+1} - \bar{u}_i)(\bar{u}_{i+1} - p_i(x_{i+1/2})) \geq 0 \tag{30}$$

$$(\bar{u}_i - \bar{u}_{i-1})(p_i(x_{i-1/2}) - \bar{u}_{i-1}) \geq 0\,. \tag{31}$$

We define any $p_i(x)$ which satisfies (30) to be a "desirable" reconstruction. If either of (30–31) are not satisfied, then the degree of $p_i(x)$ is reduced. Clearly, this restriction will never require any less than a locally linear reconstruction, as such a reconstruction is always locally monotone.

As for the application of the boundary conditions themselves, such conditions as inflow or outflow are handled in a standard manner. As it turns out, a wall condition is relatively simple to handle. Because we actually approximate the solution on the boundary of each cell, those point values $\{u_w\}$ along a solid wall are treated with an appropriate boundary condition and the numerical flux $\hat{f}_w$ on the wall is simply

$$\hat{f}_w = f(WC(u_w))\,, \tag{32}$$

with $WC(\cdot)$ denoting that a wall condition has been applied.

7. 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 co–ordinates. We have performed experiments on scalar equations in order to test for the computational order of accuracy,

and for our problems containing solid walls we solve the two–dimensional Euler equations of gas dynamics.

In the tests involving the measurement of accuracy, we use r_c to denote the "computational order of accuracy." This value is calculated by assuming a linear accumulation of error as given by (13). Also, in all of our results, when referring to a CFL coefficient, we imply its use in the "conventional" explicit time–step restriction, which in the scalar case is given by

$$\Delta t \;=\; \mathrm{CFL}\,\frac{\Delta x\,\Delta y}{\max_u\left(\,|f'(u)|\,\Delta y + |g'(u)|\,\Delta x\right)}\;,$$

and is easily extended to a hyperbolic system.

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

$$u_t + u_x + u_y \;=\; 0\;, \qquad t > 0\;, \tag{33a}$$

with initial data

$$u(x,y,0) \;=\; \tfrac{1}{2}\cos\pi\,(x+y) + \tfrac{1}{2}\;. \tag{33b}$$

The solution of (33) is 2–periodic in x and y for all time. By restricting our computational domain to $-1 \le x, y \le 1$, we thereby make the boundaries 2–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 (33a) is still non–linear, due to the adaptive stencil algorithm of the reconstruction operator R^2.

The exact solution of (33) can be easily calculated one–dimensionally in terms of the variable $\xi = x+y$, and can be written

$$u(x,y,t) \;=\; \tfrac{1}{2}\cos\pi\,(x+y-2t) + \tfrac{1}{2}\;. \tag{34}$$

However, because we computationally solve (33) on a cartesian grid, our application is truly two–dimensional. 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 (34) is smooth for all time, we apply our scheme for one period in time, *i.e.* $t = 2.0$, at a CFL of $2/3$. The number of iterations required to do this on a given grid is high enough to expect a significant accumulation of error. We have performed this test on five consecutively refined meshes for the scheme (12) with orders of accuracy $r = 1,2,3,4$. 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, along with their corresponding computational orders of accuracy "r_c," are presented in Table 1, and are

calculated with respect to the L_∞ and L_1 norms. 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 3. 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

$$u_t + (\tfrac{1}{2}u^2)_x + (\tfrac{1}{2}u^2)_y = 0\,, \qquad t>0\,, \tag{35a}$$

again with initial data

$$u(x,y,0) = \tfrac{1}{2}\cos\pi(x+y) + \tfrac{1}{2}\,. \tag{35b}$$

We solve the IVP (35) on the same domain as the previous example, and again apply periodic boundary conditions. In this case, due to the non–linearity of equation (35a), gradients immediately begin to steepen upon the first time step, until a shock eventually forms at $t = 2/\pi$. We therefore apply the scheme using a CFL of $3/4$, up to $t = 0.15$, when the solution remains smooth. Table 2 illustrates 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. Our next numerical experiment involves the solution of the two–dimensional Euler equations of gas dynamics in a cartesian frame of reference, which we write in conservative form

$$U_t + F(U)_x + G(U)_y = 0\,, \tag{36a}$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}\,, \quad F(U) = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ (\rho E + P)u \end{bmatrix}\,, \quad G(U) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ (\rho E + P)v \end{bmatrix}\,. \tag{36b}$$

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 $\vec{V}$. We close the system (36a) of four equations with the polytropic equation of state :

$$P = (\gamma - 1)\rho(E - \tfrac{1}{2}V^2)\,, \tag{36c}$$

where γ is the ratio of specific heats.

We take our test case from [11]. A simple two–dimensional inlet with a step is installed onto a uniform cartesian grid, and the problem begins with a uniform Mach 3 flow directed towards the step, from left to right. As the length scales of the inlet's configuration are identical to those in [11], we omit this description and and refer the interested reader to the cited reference. Scaling the time variable by the height of the inlet's entrance, the solution reaches a steady state at approximately $t = 12.0$. However, the structure of this steady state is relatively uninteresting, and therefore we compute the solution in a time accurate manner up to $t = 4.0$, when the flow–field contains a complicated shock structure.

At the problem's outset, we assume that the inlet is filled with air which we model as an ideal gas, with $\gamma = 1.4$, with normalized initial free–stream conditions

$$\rho_\infty = 1.4\,, \quad P_\infty = 1.0\,, \quad u_\infty = 3.0\,, \quad v_\infty = 0\,. \tag{37}$$

We implement the scheme (37) with a CFL of 0.8 on a 120×40 grid. For the flux computation in (12c-d), we approximate the required "Riemann fluxes" by the method developed by Roe [7], combined with an appropriate entropy correction (See *e.g.* [4].) The inflow boundary condition is specified by (12) and held fixed. Because the outflow is supersonic, the exit boundary condition has no effect on the flow, and therefore we assume all gradients to vanish at this boundary. At the walls, we apply the tangency condition

$$\vec{V} \cdot \hat{n}_w = 0\,, \tag{38}$$

where $\hat{n}_w$ is the unit vector normal to the wall. The nature of this solution is such that the corner of the step is the center of a rarefaction fan, and hence is a singular point of the flow. We therefore apply a special treatment at this corner, as described in [11], in order to avoid large numerical errors generated in the neighborhood of this point which would hinder our qualitative comparison.

Because second–order schemes are the current "state of the art," we present second– and fourth–order accurate solutions at $t = 4.0$ in Figures 1a through 1d, choosing density and Mach number as the variables of comparison. Both variables are plotted using thirty equally spaced contours. Sharper discontinuities are the most notable improvement in the fourth–order case, perhaps most notably the slip line emanating from the triple point near

the top wall. The Mach stem in this area is also more correct in its length and its position upstream. The weak shock from the corner of the step is also more pronounced in the fourth–order case, as is the other weaker slip line formed as this shock intersects the shock reflecting from the top of the step. This weaker slip line is virtually undetectable in the the second–order accurate solution on this computational mesh.

EXAMPLE 4. 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.* [6]), and we therefore omit any general 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 [1,2,12] 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 Figure 2a. The flow orientation is chosen to facilitate comparison with existing experimental interferograms. The problem becomes truly two–dimensional when the shock encounters the wall and forms a reflection whose structure can be quite complex. Analysis shows that the resulting similarity solution can be entirely determined by the parameters M_S and θ_w .

We examine two cases, both of which are double Mach reflections. The wall angle θ_w is 40 degrees in both cases, and the shock Mach numbers we examine are 2.87 and 3.72. This type of reflection exhibits a complex structure containing shock diffractions and slip lines, and is particularly demanding of any computational algorithm. For this reason, these computations are most commonly performed with the use of a self–similar transformation which effectively removes the time dependence of the solution. The resulting equations that are then solved resemble the steady Euler equations with the addition of source terms. However, because we wish to examine the temporal as well as the spatial accuracy of our scheme, we choose to compute our solutions in a time–accurate manner.

In addition to the demanding nature of the solution itself, such computations are also made difficult by geometric concerns, largely due to the presence of a sharp corner. Though 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 Figure 2b, and is generated by a Schwarz–Chrisoffel transformation. We could use this transformation to derive all of the necessary grid quantities referred to in Section 5. 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 our all of our mesh variables from the approximate grid lines we generate by polynomial interpolation.

The equations we solve are the time–dependent Euler equations of gas dynamics in general curvilinear co–ordinates, which we write

$$U_t + \tilde{F}(U)_\xi + \tilde{G}(U)_\eta = 0 , \tag{39a}$$

where

$$\tilde{F} = y_\eta F - x_\eta G , \qquad \tilde{G} = x_\xi G - y_\xi F , \tag{39b}$$

and F and G are the cartesian flux vectors in (36b). Our initial conditions are the two constant states U_0 and U_1 which determine the desired plane shock, oriented as in Figure 2a. We normalize these conditions with respect to the still–air initial state U_0. These conditions then are

$$\rho_0 = 1.0 , \qquad P_0 = 1.0 , \qquad u_0 = v_0 = 0 . \tag{40}$$

Using these conditions and the shock Mach number M_S, we can then determine the state U_1 by means of the Rankine–Hugoniot jump condition. These initial states are then conservatively interpolated onto the computational mesh.

Provided we ensure that the entire reflection remains within the limits of the grid, the boundary conditions are relatively simple. The initial conditions U_0 ,U_1 are applied on the left and right boundaries, while the only concern at the far–field is adequately accounting for the movement of the plane shock. On the wall, the tangency condition (38) is imposed. We apply our scheme in its curvilinear version to the two test cases for 400 time steps using a CFL of 0.8 on a 180×40 grid. Our purpose is not only to compare our numerical solutions by their order of accuracy, but also to compare each of them with experimental results, which we obtain from [3].

Figure 3a is an inteferogram resulting from an experiment designed to photographically exhibit the density structure for the case $M_S = 2.87$. (The alphabetical labeling of this picture is not relevant to our presentation.) Density contour plots for the second– and fourth–order numerical solutions are compared in Figures 3b and 3c. Overall, crisper discontinuities are observed in the fourth–order solution. The shock structure itself is also more correct in Figure 3c, in that the more perpendicular orientation of the incident Mach stem with respect to the wall is more in line with the experimental observation. The "toeing out" of this Mach stem in Figure 3b appears to be due the poorer resolution of the contact discontinuity emanating from the primary triple point.

In Figure 3d, we plot the fourth–order density solution on the wall in order to make a comparison with experimental measurements. 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 the corner at $x = 1$. Overall agreement with the experimental data is as good if not better than the similar comparison in [3] in which the numerical solution was achieved in a self–similar fashion and on a much finer grid.

In Figures 4a through 4d, we have the analogous results for the case $M_S = 3.72$. In Figure 4b, the resolution of the contact discontinuity is much worse than in Figure 3b, causing a serious error in the formation of the incident Mach stem. There is no such problem, however, in the fourth–order case (Figure 4c) where the resolution of the contact discontinuity is excellent. Finally, the density wall plot in Figure 4d appears to be an improvement over the similar result in [3].

8. Concluding Remarks

The usefulness of shock–capturing schemes which are of formal high–order accuracy in more than one spatial dimension appears certain. For instance, the results in EXAMPLE 3 suggest that these types of schemes might provide better results than are currently available in scientific areas where the resolution of weak waves is crucial, *e.g.* acoustics. EXAMPLE 4 illustrates the effect higher–order accuracy can have in the more accurate resolution of unsteady compressible flow solutions which develop complicated shock structures. Furthermore, when compressible flow solutions are also required to be viscous, these schemes can play a major role in the computation of such flows, in which there are regions containing many smooth local extrema. In addition, when such flows also develop shocks, *e.g.* shock–turbulence interactions, the need for these newly–developed schemes is undeniable.

9. Acknowledgements

The author would like to thank the management and all involved personel of the Computational Aerodynamics Branch at the NASA Langley Research Center in Hampton, Virginia for their support of this work.

REFERENCES

[1] I. I. Glas, "Some Aspects of Shock–Wave Research," *AIAA Journal*, Vol. 25, No. 2, (1987), pp. 214–229. A 398 (1985), pp. 117–140.

[2] H. M. Glaz, P. Colella, I. I. Glass, and R. L. Deschambault, "A Numerical Study of Oblique Shock–Wave Reflections with Experimental Comparisons," *Proc. R. Soc. London*, A 398 (1985), pp. 117–140.

[3] 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). (January, 1986).

[4] A. Harten and J. M. Hyman, "A Self–Adjusting Grid for the Computation of Weak Solutions of Hyperbolic Conservation Laws," *J. Comp. Phys.*, Vol. 50 (1983), pp. 235–269.

[5] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, "Uniformly High–Order Accurate Essentially Non–Oscillatory Shock–Capturing Schemes III," *J. Comp. Phys.*, Vol. 71 (1987), pp. 231–323.

[6] D. H. Jones, P. M. Martin, E. Thornhill, "A Note on the Pseudo–Stationary Fow Behind a Strong Shock Diffracted or Reflected at a Corner," . *Proc. R. Soc. London*, A 209 (1951), pp 238–247.

[7] P.L. Roe, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comp. Phys.*, V. 43 (1981), pp 357–372.

[8] C. Shu and S. Osher, "Efficient Implementation of Essentially Non–Oscillatory Shock–Capturing Schemes," *J. Comp. Phys.*, Vol. 77 (1988), pp. 439–471.

[9] C. Shu, "TVD Time Discretizations," *UCLA Mathematics Department Report*, Los Angeles, CA 90024 (May, 1986).

[10] J. Stoer, R. Bulirsch, Intorduction to Numerical Analysis, Springer–Verlag, New York, (1980).

[11] P. Woodward and P. Colella, "The Numerical Simulation of Two–Dimensional Fluid Flow with Strong Shocks," *J. Comp. Phys.*, Vol. 54 (1984) pp. 115–173.

[12] V. Young, H. Yee, "Numerical Simulation of Shock Wave Diffraction by TVD Schemes," *AIAA–87–0112*, AIAA 25th Aerospace Sciences Meeting, January 1987.

TABLE 1

Solution Error for IVP (8)

$t = 2.0$ CFL= $\frac{2}{3}$

	L_∞ error				L_1 error			
Grid	$r = 3$	r_c	$r = 4$	r_c	$r = 3$	r_c	$r = 4$	r_c
8×12	1.559 E-1		8.764 E-2		3.890 E-1		2.078 E-1	
16×24	2.331 E-2	2.74	1.021 E-2	3.10	5.678 E-2	2.78	1.605 E-2	3.69
32×48	3.002 E-3	2.97	1.239 E-3	3.04	7.627 E-3	2.97	1.764 E-3	3.20
64×96	3.749 E-4	3.00	1.390 E-4	3.16	9.130 E-4	2.99	1.237 E-4	3.82
128×192	4.666 E-5	3.01	1.544 E-5	3.17	1.142 E-4	3.00	8.658 E-6	3.84

TABLE 2

Solution Error for IVP (10)

$t = 0.15$ CFL= $\frac{3}{4}$

	L_∞ error				L_1 error			
Grid	$r = 3$	r_c	$r = 4$	r_c	$r = 3$	r_c	$r = 4$	r_c
8×12	3.765 E-2		2.632 E-2		4.938 E-2		3.010 E-2	
16×24	9.549 E-3	1.98	4.373 E-3	2.59	8.661 E-3	2.51	3.834 E-3	2.97
32×48	2.111 E-3	2.18	4.192 E-4	3.38	1.237 E-3	2.81	3.245 E-4	3.56
64×96	4.264 E-4	2.31	4.374 E-4	3.26	1.844 E-4	2.75	2.497 E-5	3.70
128×192	8.988 E-5	2.25	4.361 E-6	3.33	2.783 E-5	2.73	1.968 E-6	3.67

STEPPED INLET

$M_\infty = 3.0$

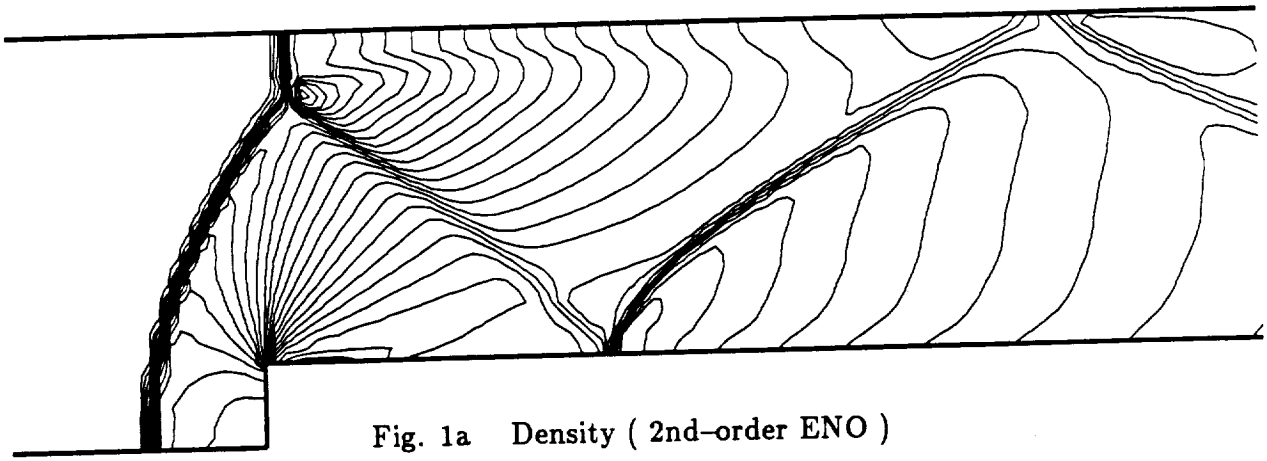

Fig. 1a Density (2nd–order ENO)

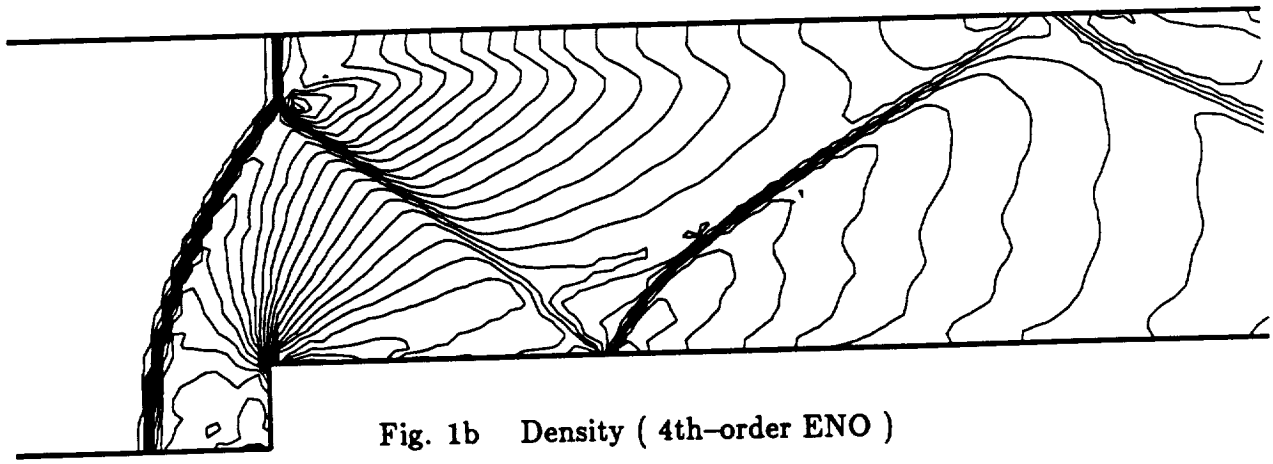

Fig. 1b Density (4th–order ENO)

STEPPED INLET

$M_\infty = 3.0$

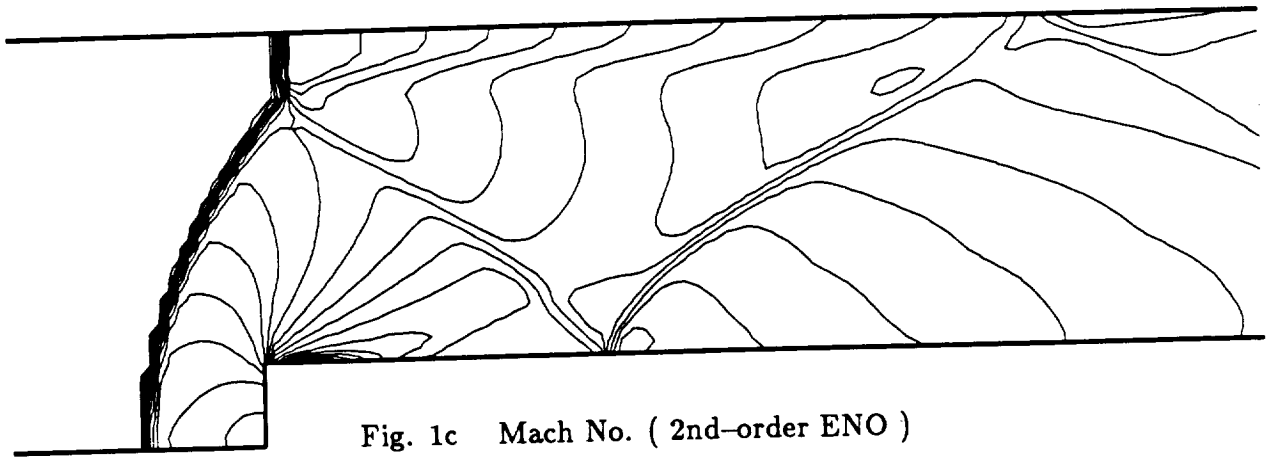

Fig. 1c Mach No. (2nd–order ENO)

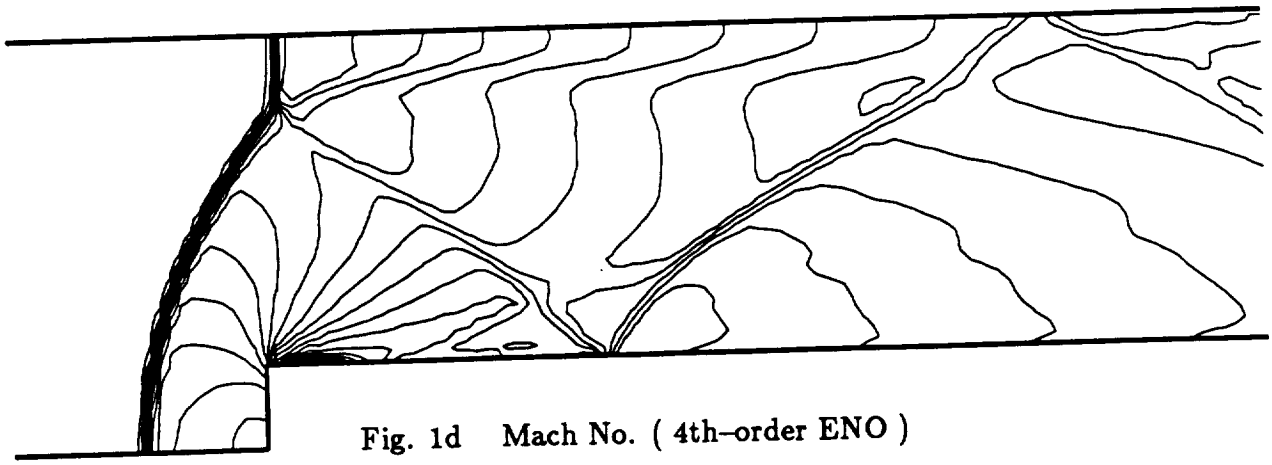

Fig. 1d Mach No. (4th–order ENO)

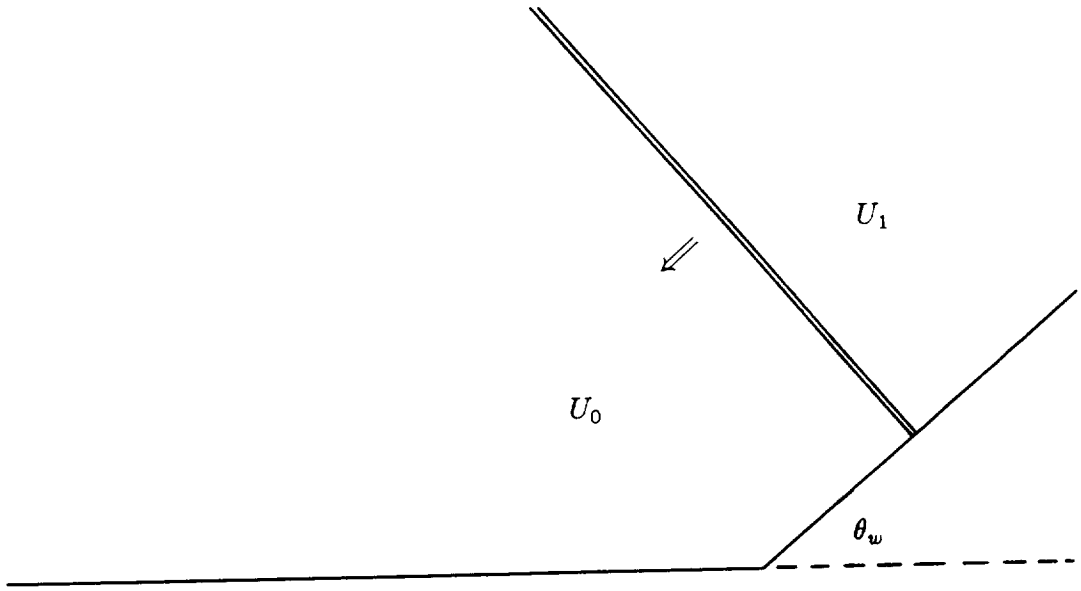

Fig. 2a Initial conditions for EXAMPLE 4

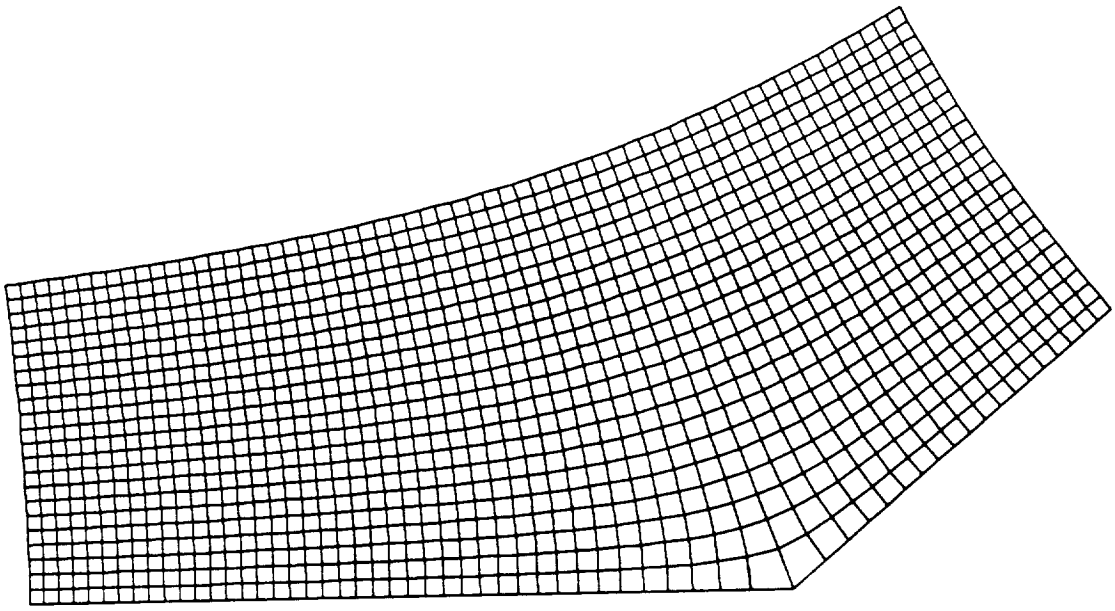

Fig. 2b Computational grid for EXAMPLE 4

MACH REFLECTION

$M_S = 2.87 \quad \Theta_w = 40°$

Figure 3a. Experimental Isopycnics (Ref. [3])

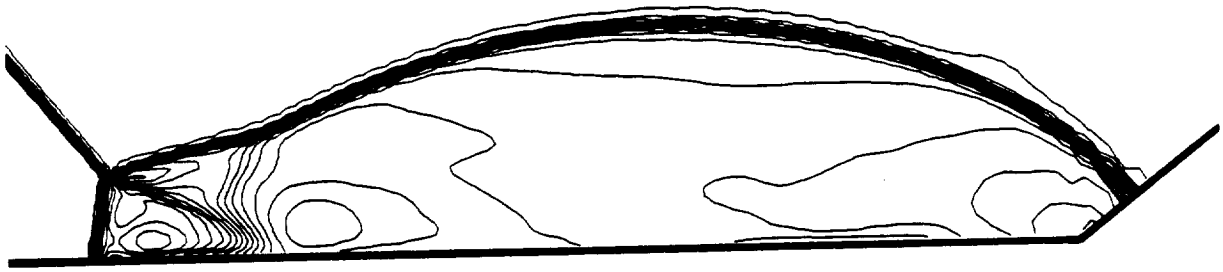

Figure 3b. Density Contours (2nd–order ENO)

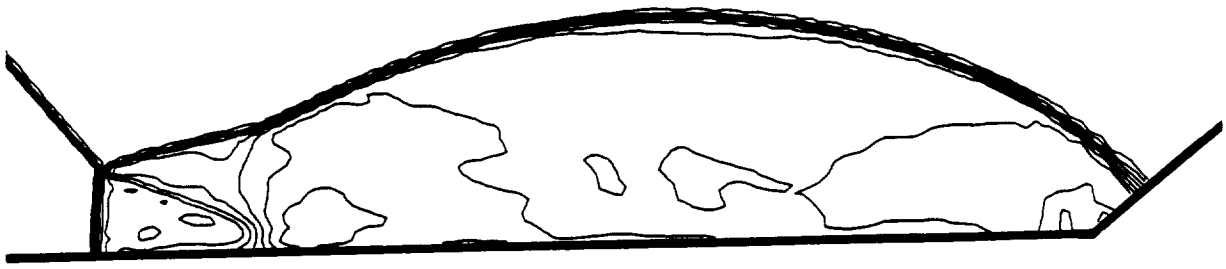

Figure 3c. Density Contours (4th–order ENO)

WALL DENSITY

$M_S = 2.87 \quad \Theta_w = 40°$

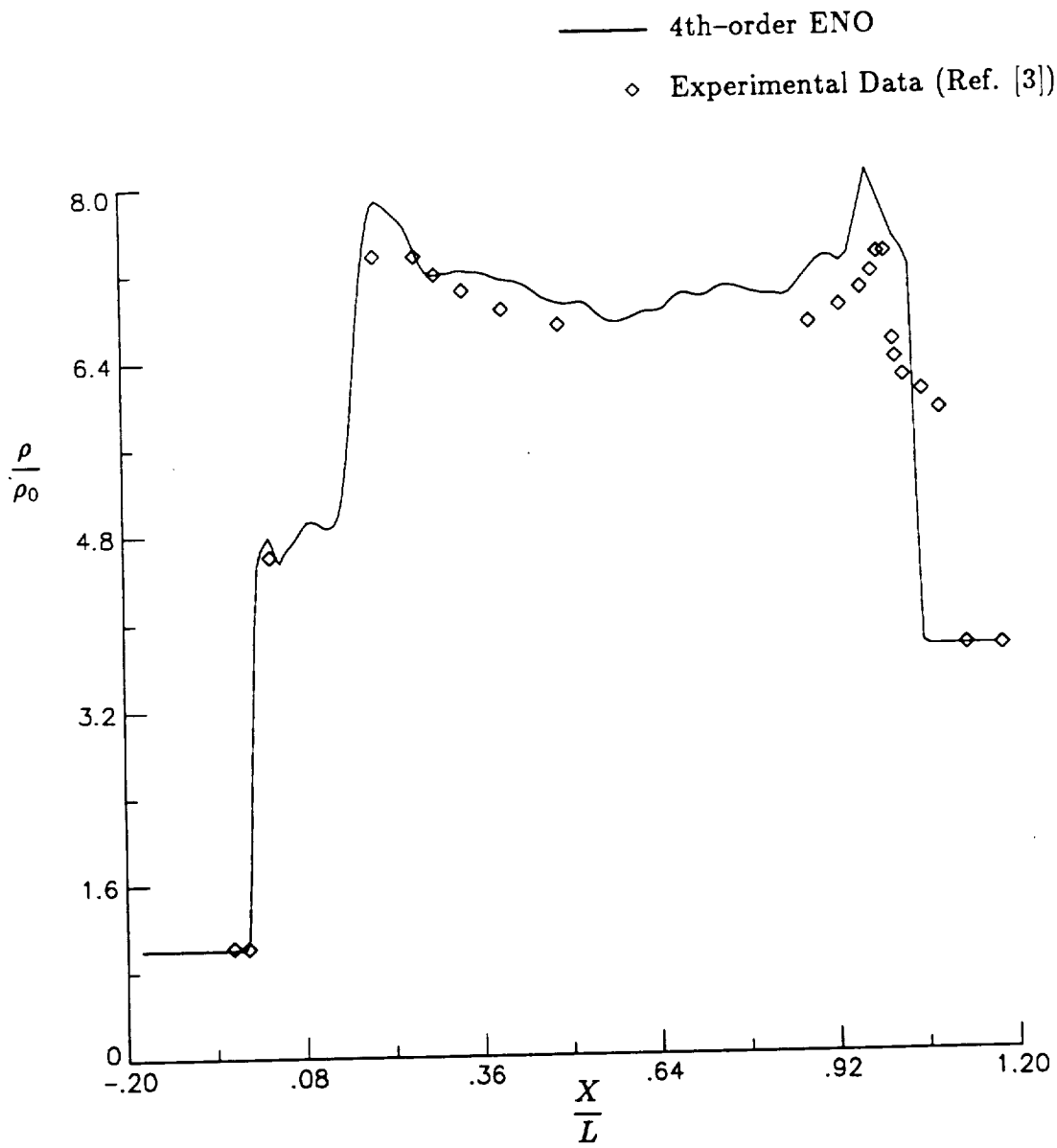

Fig. 3d Comparison of 4th–order ENO solution and experimental data

MACH REFLECTION

$M_S = 3.72 \quad \Theta_w = 40°$

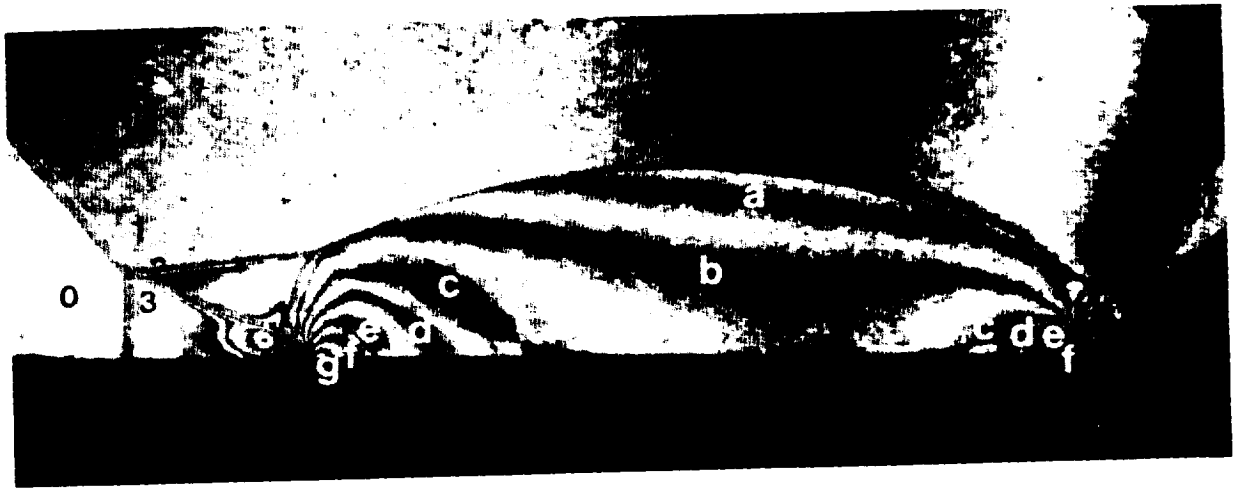

Figure 4a. Experimental Isopycnics (Ref. [3])

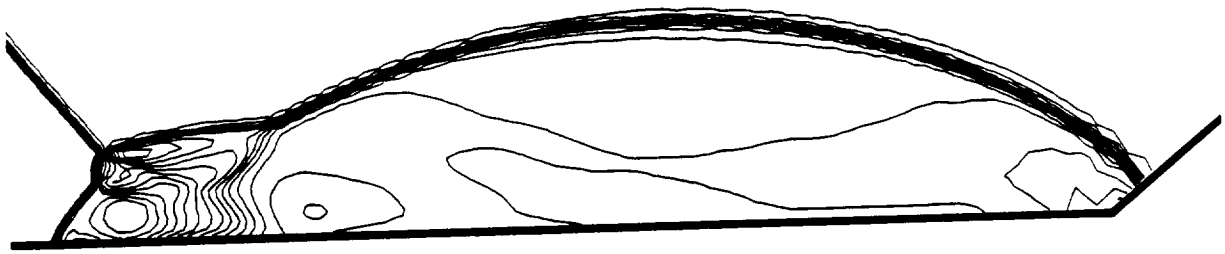

Figure 4b. Density Contours (2nd–order ENO)

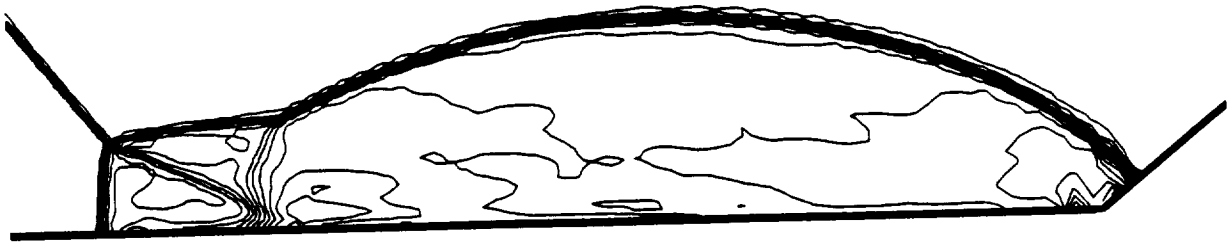

Figure 4c. Density Contours (4th–order ENO)

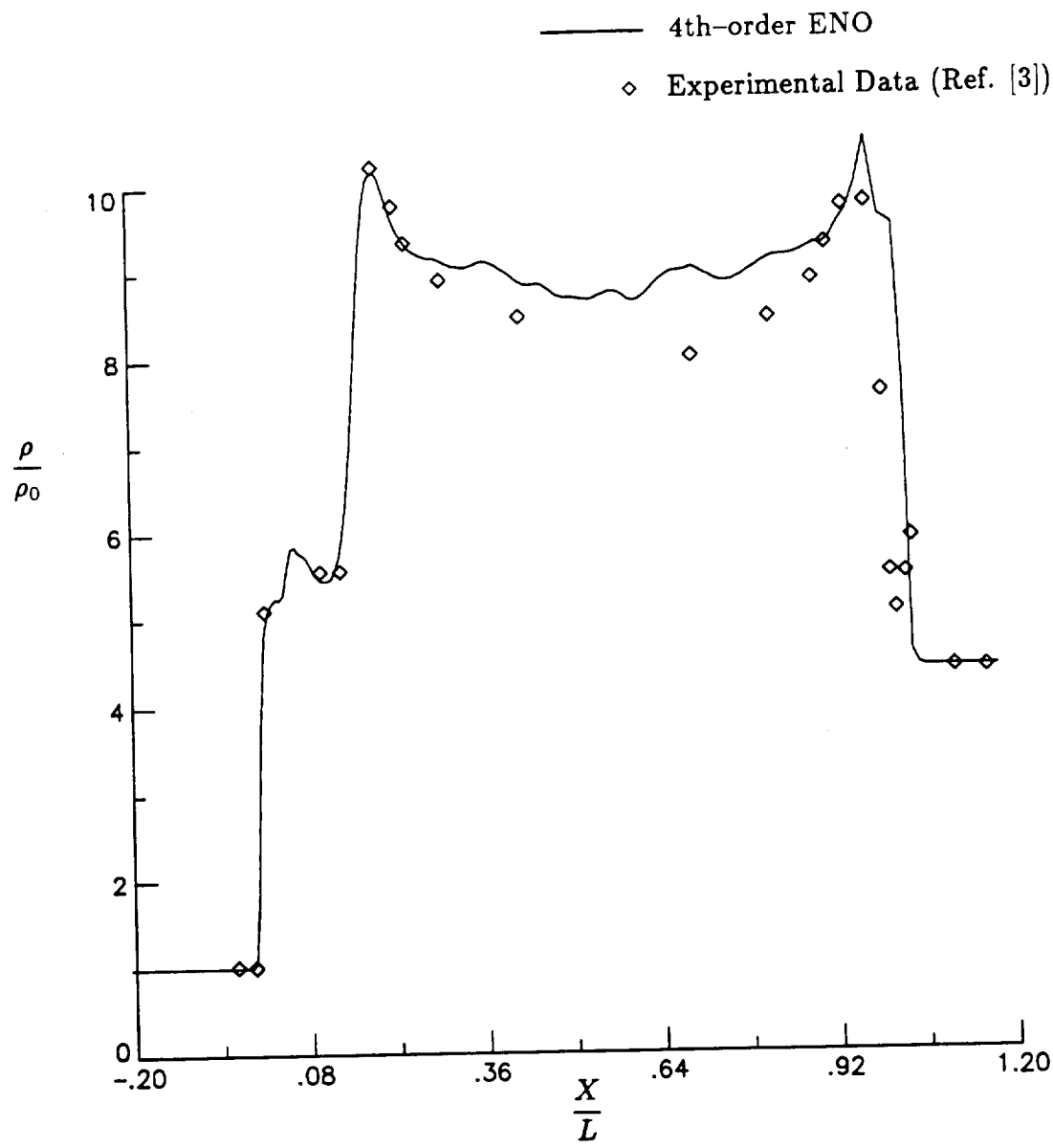

Fig. 4d Comparison of 4th–order ENO solution and experimental data