NASA Contractor Report 4285

Some Aspects of Essentially Nonoscillatory (ENO) Formulations for the Euler Equations

Sukumar R. Chakravarthy

CONTRACT NAS1-17492
MAY 1990

NASA

NASA Contractor Report 4285

Some Aspects of Essentially Nonoscillatory (ENO) Formulations for the Euler Equations

Sukumar R. Chakravarthy
Rockwell International Science Center
Thousand Oaks, California

Prepared for
Langley Research Center
under Contract NAS1-17492

National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Division

1990

Page intentionally left blank

CONTENTS

SOME ASPECTS OF ESSENTIALLY NONOSCILLATORY (ENO) FORMULATIONS FOR THE EULER EQUATIONS

ABSTRACT

This report describes an essentially nonoscillatory (ENO) formulation for hyperbolic systems of conservation laws. ENO approaches are based on "smart interpolation" to avoid spurious numerical oscillations. ENO schemes are a superset of Total Variation Diminishing (TVD) schemes. In the recent past, TVD formulations were used to construct shock-capturing finite-difference methods. At extremum points of the solution, TVD schemes automatically reduce to being first-order accurate discretizations locally while away from extrema, they can be constructed to be of higher-order accuracy. This local effect, which is necessary to prevent the Total Variation from increasing, restricts the maximum global accuracy possible for TVD schemes to third order for steady-state solutions and second order for unsteady computations. The new framework helps construct essentially non-oscillatory finite-difference methods without recourse to local reductions of accuracy to first order. Thus arbitrarily high orders of accuracy can be obtained. The basic general ideas of the new approach can be specialized in several ways and this report describes one specific implementation based on a) the integral form of the conservation laws, b) reconstruction based on the primitive function, c) extension to multiple dimensions in a tensor product fashion, d) Runge-Kutta time integration. The resulting method is fourth-order accurate in time and space, and is applicable to uniform Cartesian grids. The construction of such schemes for scalar equations and systems in one and two space dimensions is described along with several examples which illustrate interesting aspects of the new approach.

Section 1.0
INTRODUCTION

Rockwell International's Science Center, under contract with NASA Langley Research Center (LaRC), has been developing very powerful numerical methods for the Euler equations. The research began with the development of upwind schemes of first-order accuracy based on the Osher scheme and progressed to higher-order TVD schemes encompassing a variety of upwind formulations including Osher's and Roe's approximate Riemann Solvers (Refs. 1-2). Eventually, LaRC-sponsored research also included certain aspects of ENO schemes which are described here within the overall context of a general framework.

The earlier algorithmic research on TVD schemes culminated in the development of the EMTAC (Euler Marching Scheme for Accurate Computations) code for steady inviscid supersonic flows including subsonic pocket treatment (Ref. 3). The EMTAC code was delivered to NASA Langley Research Center in 1987. Rockwell International also developed a multi-zone capability that was built into the EMTAC-MZ code which was initially Rockwell proprietary (Ref. 4). As part of an extension to the original contract with NASA Langley Research Center, Rockwell agreed to make EMTAC-MZ available to NASA for their use and dissemination. The EMTAC-MZ code and its usage are separately described in a user manual (Ref. 5).

This report introduces the general framework in which a new approach has been developed for constructing shock-capturing schemes of arbitrarily high orders of accuracy. The report also presents details of one specific implementation that results in a fourth-order accurate method. The new approach is based on ENO interpolation techniques, where ENO is an acronym for "Essentially Nonoscillatory".

Until higher-order ENO schemes were developed, TVD formulations were at the forefront of shock-capturing methods available to the Computational Fluid Dynamics (CFD) community. In recent years, many finite-difference methods have been developed using TVD formulations along with Riemann Solvers[5-9]. These methods manifest many properties desirable in numerical solution procedures. By design, they avoid numerical oscillations and "expansion shocks", while at the same time being higher-order (more than first-order) accurate. ("Expansion shocks" are shock waves which do not satisfy the *entropy inequality*). TVD formulations are also based on the principle of discrete or numerical conservation which is the numerical analog of physical conservation of mass, momentum, and energy. This results in TVD schemes being able to "capture" discontinuities with ease

and high resolution. The conservation property, the avoidance of numerical oscillations and "expansion shocks", the adherence to signal propagation principles (through the Riemann Solver), and the achievement of higher accuracy enable TVD formulations to closely follow the physical principles built into the mathematical framework of the governing hyperbolic systems of conservation laws.

While TVD formulations are very reliable, versatile, and quite accurate, they do have certain inherent accuracy limitations. At extremum points of the solution, TVD schemes automatically reduce to being first-order accurate discretizations locally while away from extrema, they can be constructed to be of higher-order accuracy. This local effect, which is necessary to prevent the Total Variation from increasing, restricts the maximum global accuracy possible for TVD schemes to third order for steady-state solutions and second order for unsteady computations. These inherent limits were the motivating factor behind the development of ENO schemes which can achieve arbitrarily high orders of accuracy while, at the same time, essentially avoiding numerical oscillations. TVD schemes restricted the total variation from increasing. Higher-order ENO schemes depart from this by permitting the variation to possibly increase but in a bounded fashion. The ENO framework therefore includes TVD schemes as a subset and provides a natural unification of design principles for the construction of good shock-capturing numerical methods. Uniformly accurate ENO schemes were introduced by Harten and further developed by his colleagues in Refs. 10-13.

The ENO formulation can be used to possibly obtain greater computational efficiency (same accuracy with fewer numbers of grid points and less work), greater resolution (more accuracy for a given number of mesh points), and in general to greatly extend the boundaries of what is achievable utilizing CFD methods. Greater accuracy, in itself, is not necessarily difficult to obtain. For fluid flow problems exhibiting a sufficient degree of smoothness, finite-difference schemes of any desirable order of accuracy (provided that a sufficient number of grid points are available to realize this) can be constructed using either Taylor series methods or spectral methods. The real difficulty arises in constructing very highly accurate schemes to capture shock waves and to resolve non-smooth high-gradient regions. ENO schemes provide an eminent solution to this problem.

In this report we consider ENO schemes which are based on piecewise polynomial interpolation. We also assume that these methods use a Riemann Solver at each cell interface to construct the numerical flux based on the left and right states of the dependent variables at the cell interface. Other approaches are possible but not considered here. Assuming that an exact or a "good" approximate Riemann Solver is used, it is the construction of

the left and right states that endows the ENO formulation with its desirable properties. For every time step, the left and right states are constructed using piecewise polynomial interpolation of discrete data related to the approximate cell averages of the dependent variables.

The discussion begins with an introduction to hyperbolic systems of conservation laws followed by a presentation of the integral form on which shock-capturing numerical methods may be constructed. In such approaches, interpolation plays a direct role and "smart interpolation" helps avoid spurious numerical oscillations while also achieving high accuracy. In approaches based on the integral form, the cell-average values of the dependent variables are updated from one time level to the next. Piecewise polynomial pointwise behavior of the dependent variables is reconstructed from these cell avarages. Out of many possible reconstruction techniques, one based on the "primitive function" approach (RP) is explained for both one and two dimensions. Implementation details including high order quadrature formulae for integration of flux along cell boundaries, time-stepping method, and extension to systems of equations are all covered followed by detailed presentation of several one and two dimensional examples.

Section 2.0
HYPERBOLIC SYSTEMS OF CONSERVATION LAWS

In this report, we study the use of ENO uniformly high-order accurate schemes for the numerical approximation of weak solutions of hyperbolic systems of conservation laws.

In one spatial dimension, such systems may be written as

$$q_t + f(q)_x = 0. \tag{2.1}$$

Here, $q = (q_1, \cdots, q_m)^{\mathrm{T}}$ is a state vector and $f(q)$, the flux, is a vector valued function of m components. The system is hyperbolic in the sense that the m x m Jacobian matrix

$$A(q) = \partial f / \partial q$$

has m real eigenvalues

$$a_1(q) \leq a_2(q) \leq \cdots \leq a_m(q)$$

and a complete set of m linearly independent (right-) eigenvectors.

For multiple spatial dimensions, the system of equations may be written as

$$q_t + \vec{\nabla} \cdot \vec{F} = 0 \tag{2.2}$$

where $\vec{\nabla}$ is the gradient operator

$$\vec{\nabla} = \hat{\imath} \frac{\partial}{\partial x} + \hat{\jmath} \frac{\partial}{\partial y} + \hat{k} \frac{\partial}{\partial z} \tag{2.3}$$

and

$$\vec{F} = f_1 \hat{\imath} + f_2 \hat{\jmath} + f_3 \hat{k} \tag{2.4}$$

If we define the Jacobians $A_i = \partial f_i / \partial q$, the system (2.2) may be considered hyperbolic if the coefficient matrix

$$A = \sum_i \lambda_i A_i$$

has m real eigenvalues and a complete set of m linearly independent (right-) eigenvectors for any real choices of constants λ_i.

In particular, we will be considering the linear wave equation and the Euler equations of fluid dynamics in 1 and 2 space dimensions.

Scalar wave equation in 1-d:

$$u_t + a\,u_x = 0 \tag{2.5}$$

Scalar wave equation in 2-d:

$$u_t + a\,u_x + b\,u_y = 0 \tag{2.6}$$

Euler equations in 1-d:

$$\frac{\partial q}{\partial t} + \frac{\partial f_1}{\partial x} = 0, \tag{2.7a}$$

with

$$q = \begin{pmatrix} e \\ \rho \\ \rho u \end{pmatrix}, f_1 = \begin{pmatrix} (e+p)u \\ \rho u \\ \rho u^2 + p \end{pmatrix}. \tag{2.7b}$$

Euler equations in 2-d:

$$\frac{\partial q}{\partial t} + \frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} = 0, \tag{2.8a}$$

with

$$q = \begin{pmatrix} e \\ \rho \\ \rho u \\ \rho v \end{pmatrix}, f_1 = \begin{pmatrix} (e+p)u \\ \rho u \\ \rho u^2 + p \\ \rho v u \end{pmatrix}, f_2 = \begin{pmatrix} (e+p)v \\ \rho v \\ \rho u v \\ \rho v^2 + p \end{pmatrix}. \tag{2.8b}$$

In the Euler equations above, p is pressure, ρ is density, and the Cartesian velocity components are u and v in the x and y directions, respectively. The total energy per unit volume, e, is given by $e = p/(\gamma - 1) + \rho(u^2 + v^2)/2$ where γ is the ratio of specific heats assuming "perfect gas".

It is desirable that numerical methods devised to solve the above equations a) provide the desirable order of accuracy, b) mimic the signal propagation properties of hyperbolic systems and c) permit weak solutions (piecewise continuous) to be obtained.

Section 3.0
INTEGRAL FORM FOR CONSERVATION LAWS

Methods based on the integral form of the system of conservation laws can satisfy the three goals just outlined. We can derive the integral form by starting from the conservation laws described in their differential form in the previous section. For example, beginning with Eq. (2.1) and integrating with respect to x and i, we obtain

$$\int_t \int_x (q_t + f(q)_x)\, dx\, dt = 0 \tag{3.1}$$

$$\int_x \left(\int_t q_t\, dt \right) dx + \int_t \left(\int_x f_x\, dx \right) dt = 0 \tag{3.2}$$

$$(\bar{q}_j^{n+1} - \bar{q}_j^n)\, \Delta x + (\bar{f}_{j+1/2}^n - \bar{f}_{j-1/2}^n)\, \Delta t = 0 \tag{3.3}$$

where t^n, t^{n+1} and $x_{j+1/2}$, $x_{j-1/2}$ define the limits of integration,

$$\bar{q}_j = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} q\, dx \tag{3.4a}$$

is the cell average of the dependent variables and

$$\bar{f}_{j\pm 1/2} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f\, dt \tag{3.4b}$$

is the average flux along cell boundaries over an interval of time.

Eq. (3.3) is the fully discrete integral form of the 1-d system of conservation laws presented as Eq. (2.1) and can also be written as

$$\frac{\bar{q}_j^{n+1} - \bar{q}_j^n}{\Delta t} + \frac{\bar{f}_{j+1/2}^n - \bar{f}_{j-1/2}^n}{\Delta x} = 0. \tag{3.5}$$

Even though Eq. (3.5) resembles a finite-difference formula, it must be noted that it is an exact relation that must be satisfied by any exact solution of the differential equations. The integral form of the equations does not demand the existence of derivatives but only weaker conditions of integrability and solutions of Eq. (3.5) can also therefore include "weak solutions."

The semi-discrete version of Eq. (3.5) can be written as

$$\frac{\partial \bar{q}}{\partial t} + \frac{\hat{f}_{j+1/2} - \hat{f}_{j-1/2}}{\Delta x} = 0 \tag{3.6}$$

and can either be obtained by integrating Eq. (2.1) only with respect to x or by taking the limit $\Delta t \to 0$ in Eq. (3.5). In Eq. (3.6), $\hat{f}_{j\pm1/2}$ is the flux at cell boundary $x_{j\pm1/2}$.

The corresponding semi-discrete and fully-discrete integral forms of the multidimensional system of conservation laws (2.2) can also be easily obtained. For example, integrating Eq. (2.2) with respect to the spatial coordinates, we obtain

$$\int_V (q_t + \vec{\nabla} \cdot \vec{F})\, dV = 0 \tag{3.7}$$

where V denotes the volume of the multidimensional conservation cell under consideration. Eq. (3.7) reduces to

$$\frac{\partial}{\partial t} \int_V q\, dV + \int_S \vec{F} \cdot \hat{n}\, dS = 0 \tag{3.8}$$

where S denotes the surface that encloses the conservation cell and $\hat{n}$ is the unit normal at any point of the surface. We will see later how the boundary integration can be approximated by numerical quadrature in the context of numerical methods to solve Eq. (3.8). We now further reduce Eq. (3.8) to

$$\frac{\partial}{\partial t}(\bar{q}V) + \int_S \vec{F} \cdot \hat{n}\, dS = 0 \tag{3.9}$$

where the cell average $\bar{q}$ is defined in the usual way for a multidimensional formulation.

We have already seen that a) no derivatives appear in the fully discrete integral form and b) no spatial derivatives appear in the semi-discrete form. We observe additionally that: c) the dependent variables in the integral form are the cell averages of the original dependent variables of the differential form of the equations; d) while we will develop numerical methods directly for the integral forms, methods can also be developed directly from the differential form in a fashion that would still ensure that weak solutions can be computed (by obeying principles of discrete conservation).

Section 4.0
NUMERICAL METHODS BASED ON THE INTEGRAL FORM

We observed in the last section that Eq. (3.3) resembled a finite-difference formula that may be used to advance the cell averages $\bar{q}$ from one time level to the next if the cell boundary (face) values of the fluxes can be defined. Eq. (3.9) for the multidimensional equations is similar. The key is being able to obtain the cell face values of the flux from known values of $\bar{q}$. This may be accomplished using piecewise polynomial interpolation.

Consider the initial value problem for Eq. (3.3) defined by adding to that equation the initial conditions $\bar{q}_j^0, j = 1, \cdots, J$. Then, a numerical algorithm to solve Eq. (3.3) may be defined as follows:

a) Interpolate $\bar{q}_j$ to obtain piecewise polynomial pointwise behavior of q within each cell.

b) Each polynomial (within each cell) may be evaluated at $x_{j\pm1/2}$ for that cell. Collecting all such values, we find that we have, at each cell face, left and right values $(q_L, q_R)_{j+1/2}$.

c) Resolve the discontinuity at each cell face using solutions to the Riemann problem. (The solution procedure is usually referred to as the Riemann Solver.) This will result in a knowledge of $\hat{f}_{j+1/2}$ which we shall henceforth call the numerical flux.

d) Substitute $\hat{f}_{j\pm1/2}$ into Eq. (3.3) to advance the solution to the next time level. Proceed to step (a) and repeat.

Notes:

1) In step (a), we must construct piecewise polynomials that match the given cell averages. This is different from the usual interpolation of discrete pointwise values. There are at least three different ways of performing such interpolation in order to reconstruct the pointwise behavior of the original dependent variables in each cell — i) reconstruction by deconvolution (RD), ii) reconstruction using the primitive function formulation (RP), and iii) reconstruction by matching cell averages directly (RM). The second method, RP, will be explained in this report.

2) One may wonder why piecewise polynomials should be used, especially when one sees in step (b) that this will lead to discontinous behavior at cell interfaces. In fact, the choice of piecewise polynomials is particularly apt for just that reason. After all, we

must allow our interpolation model to permit discontinuities since our goal is to be able to compute "weak" solutions. It is true, however, that with piecewise polynomials of the type described in this report, the approximation always pushes any discontinuity to be at the cell interfaces. Further refinements have already been devised to enable discontinuities to be located even within the cell ("subcell resolution" — Ref. 14) but it is beyond the scope of this report to delve into such advances.

3) When dealing with systems of equations, questions arise regarding the choice of variables to interpolate: should the reconstruction techniques be based on matching the basic conservation variables' averages, "primitive" variables', "characteristic" variables', etc.? These issues are briefly revisited in a later section.

4) In the case of interpolation with piecewise-constant polynomials, if we consider two neighboring cells, we have two sets of constant values, one to the left of and one to the right of each cell interface. This resembles the classical Riemann problem. When higher degree polynomials are chosen, the left and right states are not constant but a Riemann Solver may still be used to construct the solution at the instant of initial contact between the discontinuities. More sophisticated Riemann Solvers may also be sought — those that resolve piecewise linear left and right state variations, etc. In this report, the semi-discrete formulation is utilized to construct higher-order time-accurate schemes. It may be observed by looking at Eq. (3.6) that if we use a method-of-lines approach and embed the semi-discrete form in a Runge-Kutta time-integration scheme, for example, then only the pointwise values of the cell interface fluxes are required. These can be obtained using a Riemann Solver based on local values of left and right states.

5) Going back to note (2), we can also add that for smooth data, the magnitude of the difference between q_L and q_R behaves with $O(\Delta x^{r+1})$ where r is the degree of the interpolating polynomial. Thus, the piecewise polynomial approach is appropriate for obtaining both smooth solutions and solutions with discontinuities. For smooth solutions, the need for using "good" Riemann Solvers becomes decreasingly important with increasing degree of polynomial approximation.

6) The procedure for multidimensional flows is similar to that for one-dimensional problems. Interpolation in step (a) must be carried out in a suitable multidimensional way. The boundary integration of Eq. (3.9) can be replaced by a suitable quadrature. The need for a multidimensional Riemann Solver can be obviated by exploiting a suitable combination of several pointwise (in time and space) Riemann problems.

We now use a set of figures to help visualize the above concepts. Figure 1 simply outlines the integration limits for the 1-d integral form. Figure 2 shows how an initial value problem (IVP) for Eq. (2.1) can be replaced by the corresponding one for cell averages given by Eq. (3.5). Assuming that piecewise constant reconstruction was used, Figure 3 zooms in on one local Riemann problem (IVP with piecewise constant states) and Figure 4 helps visualize how the individual Riemann problems, taken together, provide the means to update the cell averages to the next time level.

In the following sections, we consider the two important steps of the solution procedure in more detail: 1) the Riemann Solver, and 2) interpolation.

$$\left(\bar{q}_j^{\,n+1} - \bar{q}_j^{\,n}\right)\Delta x + \left(\hat{f}_{j+1/2} - \hat{f}_{j-1/2}\right)\Delta t = 0$$

$$\frac{\bar{q}_j^{\,n+1} - \bar{q}_j^{\,n}}{\Delta t} + \frac{\left(\hat{f}_{j+1/2} - \hat{f}_{j-1/2}\right)}{\Delta x} = 0$$

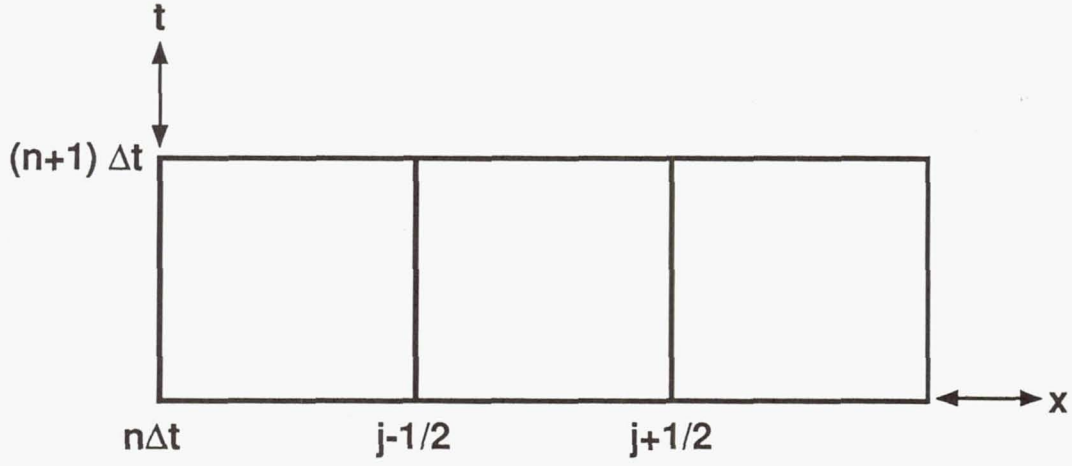


Figure 1. 1-d integration cell limits

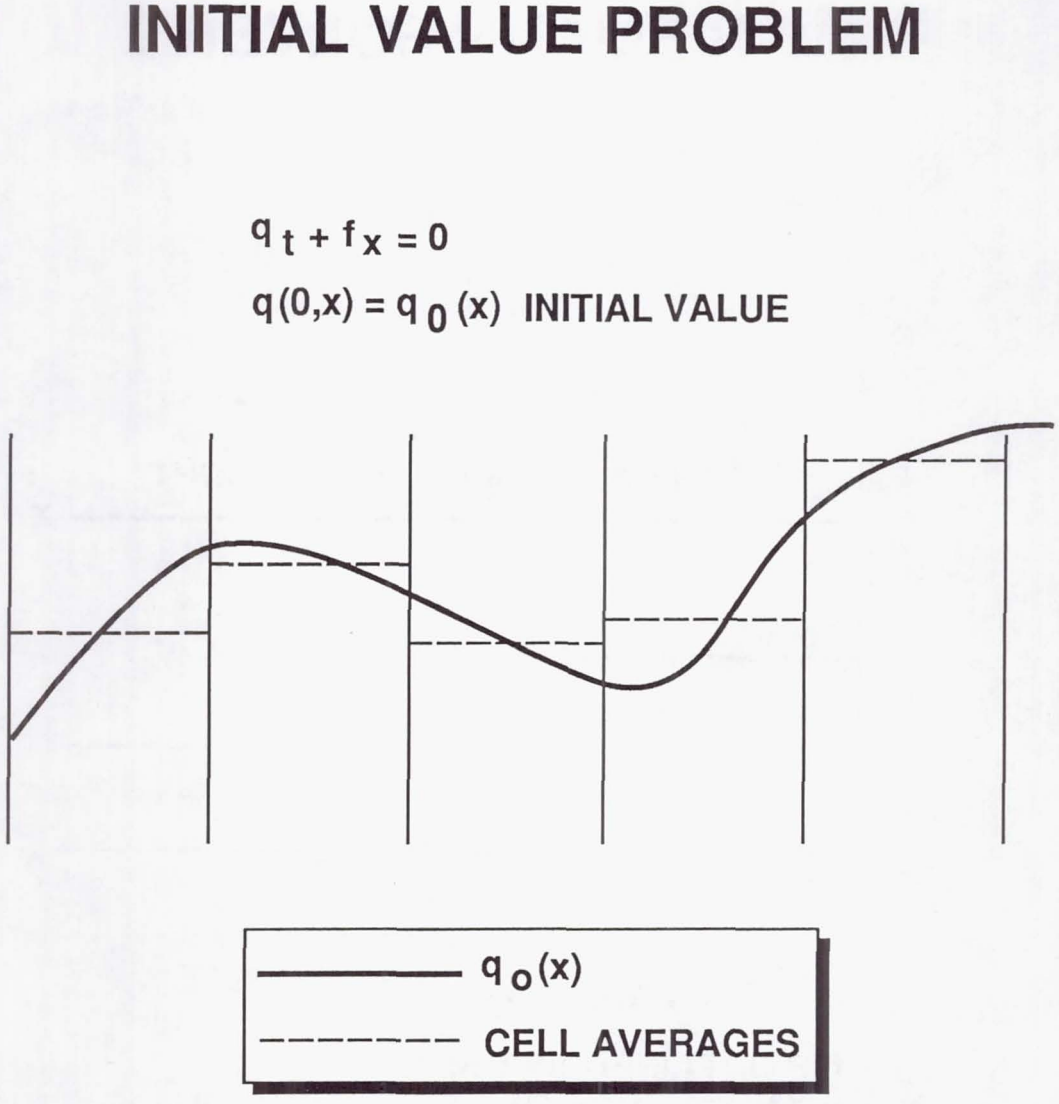
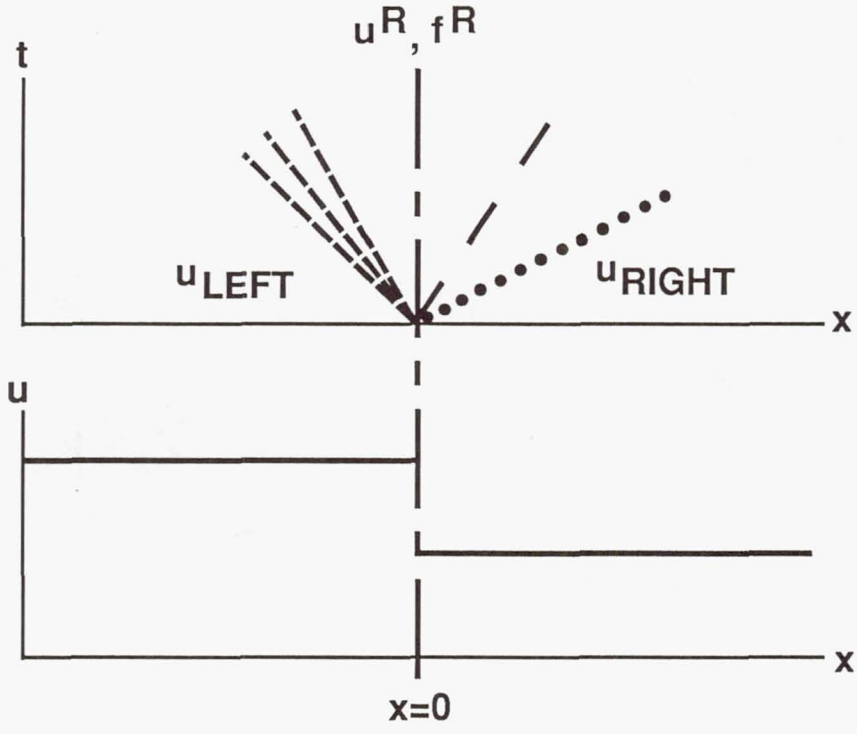


Figure 2. IVP for cell averages

RIEMANN SOLVER



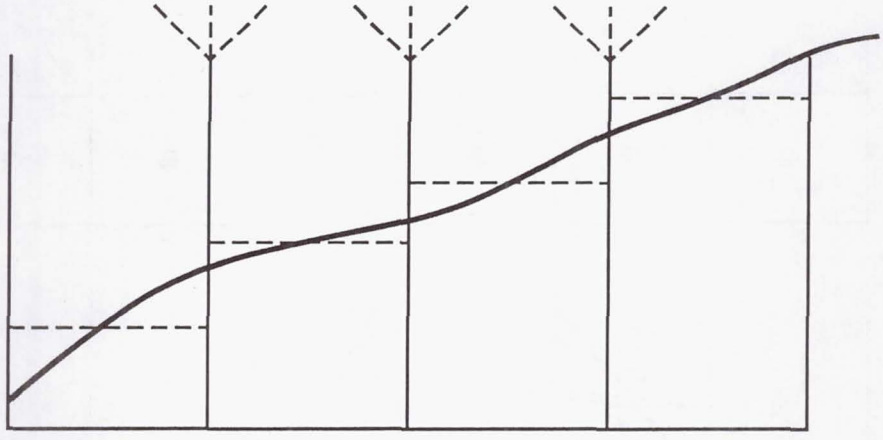
f^R = FLUX AT- $x_{j+1/2,\ t=0+\varepsilon}$

f IS OBTAINED BY USING
A KNOWLEDGE OF DECOMPOSITION
OF INITIAL DISCONTINUITIES

Figure 3. Riemann Solver

USING RIEMANN SOLVER

$$\frac{\bar{q}^{n+1} - \bar{q}^{n}}{\Delta t} + \frac{\hat{f}_{j+1/2} - \hat{f}_{j-1/2}}{\Delta x} = 0$$



WE DEFINE $\hat{f} = f^{RIEMANN}$

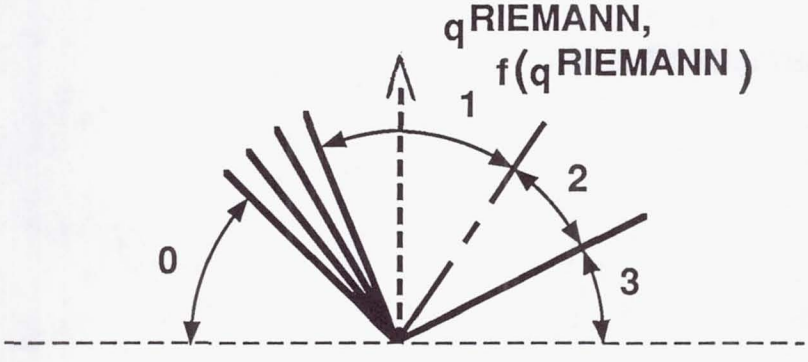


Figure 4. Using Riemann problem solutions

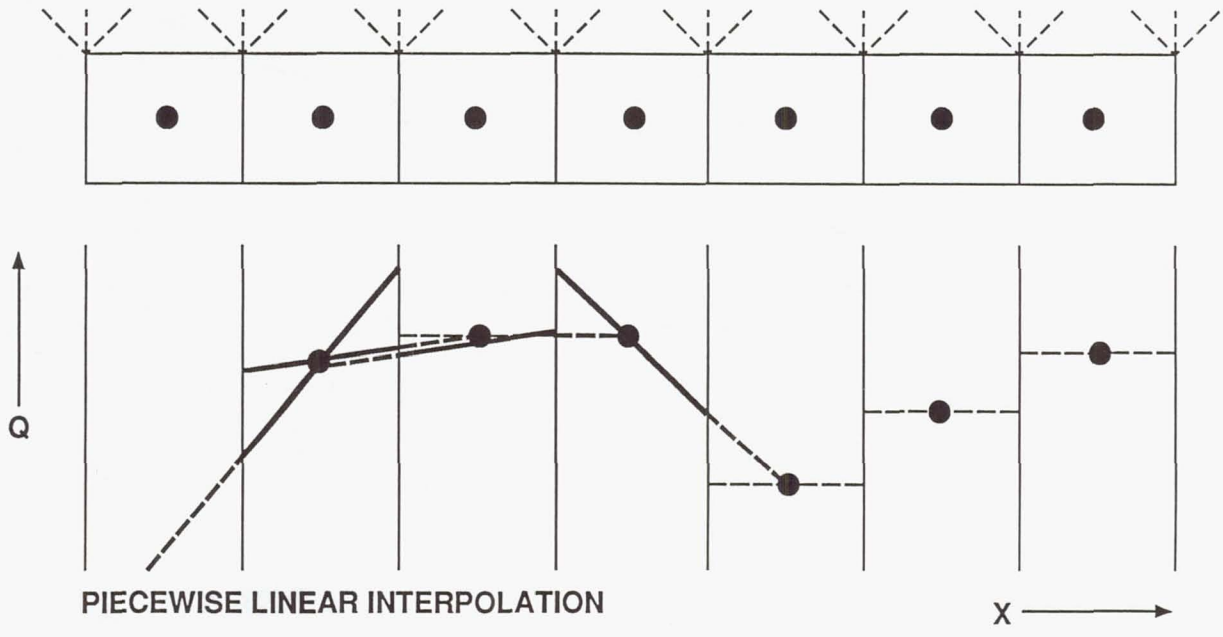


Figure 5. Piecewise constant and linear interpolations

Section 5.0
THE RIEMANN SOLVER

The Riemann problem is an initial value problem with piecewise constant initial data (one set to the left of the origin and the other to the right). It is similar to the "shock tube problem" of gasdynamics where at $t = 0$, a diaphragm separates the left state from the right. In the shock tube, the bursting of the diaphragm brings the left and right states together. In the mathematical statement, we keep the left and right states separated at $t = 0$ and at $t > 0$, we let the two states interact.

"Riemann Solver" is the name given to the procedure that constructs the solution to the Riemann problem. The exact solution for the linear wave equation and the Euler equations mentioned in Section 2.0 are well known. The solution comprises a quantitative knowledge regarding $q(x,t)$ for $-\infty < x < +\infty$ and $t > 0$. It turns out that the solution is self similar in the variable $\theta = x/t$. Therefore, $q(x,t) = q^R(\theta)$, with the superscript R denoting the Riemann problem.

The exact solution to the Riemann problem for the equations in Section 2.0 is made up of piecewise constant states separated by transitions. Each transition is associated with an eigenvalue of the Jacobian matrix. For the 1-d Euler equations, there are three eigenvalues $u - c$, u and $u + c$, where c is the speed of sound ($c = \sqrt{\gamma p/\rho}$). The transitions associated with $u \pm c$ can either be a shock wave or a rarefaction and that associated with u is called a contact discontinuity. Reference 1 provides formulae for the construction of the exact solution. In particular, this provides the extents of the piecewise constant states and the magnitudes of the transitions. From this information, the value of q along the ray $\theta = 0$ may be determined. We denote this by q^R and the corresponding flux as $f^R = f(q^R)$.

Consider now Eq. (3.6) and the steps (a)-(d) of the solution procedure given in Section 4.0. We assume piecewise constant behavior of dependent variables

$$q(x) = \bar{q}_j, \quad x_{j-1/2} < x < x_{j+1/2} \tag{5.1}$$

This results in local Riemann problems which can be solved to construct

$$\hat{f}_{j+1/2} = f^R_{j+1/2} \tag{5.2}$$

These numerical fluxes can be substituted into Eq. (3.6) along with a suitable time-stepping procedure to advance $\bar{q}_j$.

Since for hyperbolic systems it takes finite time for spatially separated locations to influence each other, for a sufficiently small time step, the Riemann problem from $x_{j-1/2}$ will not affect the solution to the Riemann problem at $x_{j+1/2}$ and vice-versa. We call such a time step as Δt_{CFL} with the subscript referring to the Courant-Friedrichs-Lewy stability limit for linear equations.

$$\text{For} \quad \Delta t \leq \Delta t_{CFL}, \quad \bar{f}^n_{j+1/2} = \hat{f}_{j+1/2} \tag{5.3}$$

and therefore the fully discrete form, Eq. (3.5) may also be used to advance the solution $\bar{q}_j$.

We must note that, when there are source terms in 1-d, for higher-order polynomial interpolation, and for multidimensional problems, it is still true that two spatially separated Riemann problems will not influence each other for a sufficiently small interval of time. However, each Riemann problem solution is no more self similar under these circumstances and therefore Eq. (5.3) is not true. In such cases, it is convenient to resort to the method of lines (semi-discrete) approach.

We now illustrate three properties:

1) Discretization methods such as those described in this section (which use a Riemann Solver) are "upwind" schemes.

2) Methods based on piecewise constant interpolation are only first-order accurate.

3) First-order accurate upwind schemes are monotonocity preserving.

When a Riemann flux is used as the cell interface flux, the fully discrete integral form defined in Eq. (3.5) becomes

$$\bar{q}^{n+1}_j = \bar{q}^n_j - \frac{\Delta t}{\Delta x}(\hat{f}^n_{j+1/2} - \hat{f}^n_{j-1/2}) \tag{5.4}$$

where $\hat{f}_{j+1/2} = f^R_{j+1/2}$. Adding and subtracting f_j from the second term on the right hand side (RHS), we can rewrite that term to be

$$\hat{f}_{j+1/2} - \hat{f}_{j-1/2} = (\hat{f}_{j+1/2} - f_j) + (f_j - \hat{f}_{j-1/2}) \tag{5.5}$$

The term $\hat{f}_{j+1/2} - f_j$ includes the effect of all left-moving waves from the right. The term $f_j - \hat{f}_{j-1/2}$ includes the effect of all right-moving waves from the left. Therefore, Eq. (5.4)

describes a method of updating $\bar{q}_j$ that accounts for the appropriate signal propagation effects, and hence describes an "upwind" scheme. Note that this was goal (b) identified at the end of Section 2.

We leave it to the reader to show that when $a > 0$ in Eq. 2.5,

$$\hat{f}_{j+1/2} = a\bar{u}_j, \quad \hat{f}_{j-1/2} = a\bar{u}_{j-1} \tag{5.6}$$

and therefore, we obtain

$$\bar{u}_j^{n+1} = \bar{u}_j^n - a\frac{\Delta t}{\Delta x}(\bar{u}_j^n - \bar{u}_{j-1}^n) \tag{5.7}$$

Once again, we see that the numerical algorithm defined using piecewise constant polynomials and a Riemann Solver results in an "upwind" scheme.

For piecewise constant and piecewise linear polynomial approximations, the value of the cell average is also the pointwise value at the midpoint of each cell. Thus, rewriting Eq. (5.7) as

$$u_j^{n+1} = u_j^n - \nu(u_j^n - u_{j-1}^n) \tag{5.8}$$

where $\nu = a\,\Delta t/\,\Delta x$, we see that

$$u_j^{n+1} = (1-\nu)u_j^n + \nu u_{j-1}^n \tag{5.9}$$

and therefore the method is monotonicity preserving for the linear wave equation as long as $\nu \leq 1$. It is clear that the values u_j^{n+1} will be bounded by the maxima and minima of u_j^n, and if the u_j^n described a monotone profile, then u_j^{n+1} will preserve such monotonicity. A Taylor-series analysis of Eq. (5.8) will also show that the finite-difference scheme is first-order accurate.

As motivation for the following sections, we look at Eq. (5.9) from the following perspective. For the linear wave equation, the solution u_j^{n+1} should be equal to the solution at $t = t^n$ at the foot of the characteristic drawn backwards from x_j, t^{n+1}. The RHS of Eq. (5.9) is equal to that value computed using linear interpolation of the discrete values u_j^n and u_{j-1}^n.

We now describe some generalizations that we can use as framework for describing many schemes including those that are based on "approximate" Riemann Solvers or even

those that are not based on Riemann Solvers at all. Extending the second term on the RHS of Eq. (5.4) even further along the lines shown in Eq. (5.5),

$$\hat{f}_{j+1/2} = \frac{f(q_R)_{j+1/2} + f(q_L)_{j+1/2}}{2} - \frac{(f(q_R)_{j+1/2} - f^R_{j+1/2}) - (f^R_{j+1/2} - f(q_L)_{j+1/2})}{2} \tag{5.10}$$

Note the use of superscript R to denote the Riemann problem solution and the subscripts R and L to denote right and left states. This can be rewritten, after dropping the subscript $j + 1/2$, as

$$\hat{f} = \frac{f(q_R) + f(q_L)}{2} - \frac{(df)^+ - (df)^-}{2} \tag{5.11}$$

It is shown in Ref. 1 how this form can be used to represent methods using Osher's or Roe's approximate Riemann Solvers in addition to that using the exact Riemann Solver described earlier in this section (also known as the Godunov scheme). In fact, Eq. (5.11) can be used to represent even Split-Flux schemes as well as schemes that do not use Riemann Solvers at all. For example,

$$\hat{f} = \frac{f(q_R) + f(q_L)}{2} - \phi \frac{q_R - q_L}{2} \tag{5.12}$$

where ϕ can be a positive constant following the Lax-Friedrichs scheme or computed as the absolute value of the maximum local eigenvalue in the manner of the Rusanov scheme. We have already observed that such simpler approaches become quite useful with higher degree polynomial interpolation.

Section 6.0
THE ROLE OF INTERPOLATION

Considering the semi-discrete form Eq. (3.6) or (3.9), we observe that even if we are able to use an exact time-stepping scheme, we must contend with the inaccuracies of the polynomial approximations used for the spatial discretization. Therefore we now take a detailed look at piecewise polynomial approximations.

The role of interpolation, in the context used, is to provide a way to reconstruct the pointwise behavior from a knowledge of the cell average values of the dependent variables. Considering 1-d for simplicity, we assume that the dependent variables q are described by the polynomials $P_j(x)$ in each cell j.

$$P_j(x) = \sum_{i=0}^{r} b_{ij} x^i \tag{6.1}$$

where b_i are the polynomial coefficients and r is the degree of the interpolating polynomial. Any polynomial that we choose must satisfy the requirement that

$$\int_{x_{j-1/2}}^{x_{j+1/2}} P_j(x)\, dx = \bar{q}_j\, \Delta x_j \tag{6.2}$$

In words, the average of the interpolating polynomial must equal the cell average values of the dependent variables. Similar relations may be easily obtained for multidimensional problems.

For the case of a piecewise constant description, it is easy to see that

$$b_{0j} = \bar{q}_j \tag{6.3}$$

In this case, the polynomial for each cell j is completely determined from the cell average value $\bar{q}_j$ for that cell. We have already seen that this results in a first-order accurate upwind scheme when used with a Riemann Solver.

For polynomial interpolation of degree 1 or higher, we cannot define the polynomial coefficients in each cell by just considering $\bar{q}_j$ of that cell. We can define the coefficients, however, by matching the cell averages at the required number of neighboring cells. This is the method of reconstruction by matching cell averages directly, identified as RM in

Section 4.0. For example, b_{0j} and b_{1j} can be defined uniquely by applying Eq. (6.2) to cells j and $j-1$. We can also define them by matching $\bar{q}_{j+1}$ and $\bar{q}_j$. We therefore have two choices of piecewise linear polynomials as well as the old choice of piecewise constant polynomial. These choices are illustrated in Figure 5. For interpolation polynomials that are not of higher degree than piecewise linear, the cell average values $\bar{q}_j$ are also equal to $q(x_j)$, the mid point value of the dependent variables.

For specific values of $\bar{q}_j = q(x_j)$ shown in Figure 5, on the right hand side of the figure, piecewise constant interpolation is indicated. On the left half of the figure, both choices of piecewise linear interpolation are shown for each cell. It becomes clear that some choices may be better than others if we do not want to introduce new extrema via the interpolation process. In some cells, the choice based on $\bar{q}_j, \bar{q}_{j-1}$ may be preferable to the one based on $\bar{q}_{j+1}, \bar{q}_j$, while in other cells, the reverse may be true. At local maxima or minima, either choice would introduce new extrema, and if that is to be strictly avoided, one must revert to piecewise constant interpolation locally. Methods that seek to strictly avoid introducing new extrema along the lines described above, are known as TVD schemes. The advantages and drawbacks of TVD formulations were described in the introductory section.

The goal is therefore to obtain polynomial interpolation of as high a degree that is compatible with the desired order of accuracy and simultaneously avoid spurious numerical oscillations caused by extrema introduced via polynomial interpolation. Along the lines covered for TVD schemes, the approach is to use "smart" interpolation. However, to achieve uniformly high orders of accuracy, we do not want to hybridize with piecewise constant behavior anywhere. The goal is therefore to construct ENO interpolation by choosing the best polynomial among the alternatives available of equal degree of interpolation. ENO interpolation is covered in detail in the next section.

Section 7.0
ENO INTERPOLATION

While the previously described framework covers both scalar and systems of hyperbolic conservation laws, in the remaining subsections we restrict our attention to scalar equations. The interpolation procedure for systems of equations will be described in a later section.

In this section, we introduce $H_m(x;w)$, a piecewise polynomial function of x that interpolates w at the points $\{x_j\}$, i.e.,

$$H_m(x_j;w) = w(x_j) \tag{7.1a}$$

$$H_m(x;w) \equiv P_{m,j+1/2}(x;w) \text{ for } x_j \le x \le x_{j+1} \tag{7.1b}$$

where $P_{m,j+1/2}$ is a polynomial in x of degree m.

We take $P_{m,j+1/2}$ to be the (unique) (m-th)-degree polynomial that interpolates $w(x)$ at the $(m+1)$ successive points $\{x_i\}, i_m(j) \le i \le i_m(j)+m$, that include x_j and x_j+1, i.e.

$$\begin{aligned} P_{m,j+1/2}(x_i;w) &= w(x_i) \\ &\text{for } i_m(j) \le i \le i_m(j)+m \end{aligned} \tag{7.2a}$$

$$1-m \le i_m(j) - j \le 0 \tag{7.2b}$$

Clearly there are exactly m such polynomials corresponding to the m different choices of $i_m(j)$ subject to Eq. 7.2b. This freedom is used to assign to (x_j, x_{j+1}) a stencil of $m+1$ points (Eq. 7.2) so that $w(x)$ is "smoothest" in $(x_{i_m(j)}, x_{i_m(j)+m})$ in some asymptotic sense.

The information about smoothness of $w(x)$ is extracted from a table of divided differences of w. The latter can be defined recursively by

$$w[x_i] = w(x_i) \tag{7.3a}$$

$$\begin{aligned} &w[x_i, \cdots, x_{i+k}] = \\ &\frac{(w[x_{i+1}, \cdots, x_{i+k}] - w[x_i, \cdots, x_{i+k-1}])}{(x_{i+k} - x_i)} \end{aligned} \tag{7.3b}$$

The algorithm to evaluate $i_m(j)$ is recursive. We start by setting

$$i_1(j) = j \quad , \tag{7.4a}$$

i.e. $P_{1,j+1/2}$ is the first-degree polynomial interpolating w at x_j and x_{j+1}. Let us assume that we have already defined $i_k(j)$, i.e. $P_{k,j+1/2}$ is the (k-th)-degree polynomial interpolating w at $x_{i_k(j)}, \cdots, x_{i_k(j)+k}$. We consider now as candidates for $P_{k+1,j+1/2}$ the two ($(k+1)$-th)-degree polynomials obtained by adding to the above stencil the neighboring point to the left or the one to the right; this corresponds to setting $i_{k+1}(j) = i_k(j) - 1$ or $i_{k+1}(j) = i_k(j)$, respectively. We choose the one that gives a $(k+1)$-th order divided difference that is smaller in absolute value, i.e.

$$i_{k+1}(j) = \begin{cases} i_k(j) - 1 & \text{if } |w[x_{i_k(j)-1}, \cdots, x_{i_k(j)+k}]| \\ & \quad < |w[x_{i_k(j)}, \cdots, x_{i_k(j)+k+1}]| \\ i_k(j) & \text{otherwise.} \end{cases} \tag{7.4b}$$

In two dimensions, we will adopt a tensor-product approach in this report:

$$P(x,y) = P^x(x)P^y(y) \tag{7.5}$$

In this fashion, we can use the one-dimensional "best" polynomials defined above in order to construct both one-dimensional and two-dimensional reconstructions. This forms the subject matter of the next section.

Procedure 1 is a FORTRAN subprogram to determine the best 1-d stencil. It is invoked repeatedly, in Procedure 2, to compute the best stencil in each direction. We will see in the next section how such stencils may be used in 2-d reconstructions.

```
      subroutine ninter(ii,du,ni,n)
      dimension u(50),ii(50),du(4,50)
c     ****************************
c     nonoscillatory interpolation
c     ****************************
      do 30 m=2,ni
      do 35 i=1,n-1
   35 du(m,i)=du(m-1,i+1)-du(m-1,i)
      du(m,n)=du(m-1,  1)-du(m-1,n)
      do 40 i=1,n
      i0=ii(i)
      ip=i0
      if(ip.le.0)ip=ip+n
      im=i0-1
      if(im.le.0)im=im+n
      ii(i)=i0+imn(du(m,im),du(m,ip))
   40 continue
   30 continue
      return
      end
      function imn(x,y)
      imn=0
      if(abs(x).le.abs(y))imn=-1
      return
      end
```

Procedure 1. Nonoscillatory interpolation

```
      subroutine stencil(u,ii,jj,nx,ny)
      dimension u(50,50),ii(50,50),jj(50,50)
      dimension du(4,50),kk(50)
      do 10 i=1,nx
      do 5 j=1,ny
      du(1,j)=u(i,j)
    5 kk(j)=j
      call ninter(kk,du,4,ny)
      do 7 j=1,ny
    7 jj(i,j)=j-kk(j)
   10 continue
      do 20 j=1,ny
      do 25 i=1,nx
      du(1,i)=u(i,j)
   25 kk(i)=i
      call ninter(kk,du,4,nx)
      do 27 i=1,nx
   27 ii(i,j)=i-kk(i)
   20 continue
      return
      end
```

Procedure 2. Best stencils in each direction

Section 8.0
RECONSTRUCTION BY PRIMITIVE FUNCTION

Given cell averages $\bar{w}$ of a piecewise smooth function w

$$\bar{w}_j = \frac{1}{h_j} \int_{x_{j-1/2}}^{x_{j+1/2}} w(\xi)\, d\xi \quad , \quad h_j = x_{j+1/2} - x_{j-1/2} \tag{8.1}$$

we can immediately evaluate the point-values of the primitive function $W(x)$

$$W(x) = \int_{x_0}^{x} w(y)\, dy \tag{8.2a}$$

by

$$W(x_{j+1/2}) = \sum_{i=i_0}^{j} \bar{w}_j \quad . \tag{8.2b}$$

Since $w(x) \equiv \frac{d}{dx} W(x)$, we apply interpolation to the pointwise values (Eq. 8.2b) of the primitive function $W(x)$ and then obtain a pointwise reconstruction by defining

$$R(x; \bar{w}) = \frac{d}{dx} H_r(x; W) \quad . \tag{8.3}$$

While there are several ways to reconstruct pointwise behavior in 2-d, we present one specific method here based on the primitive function and a tensor-product approach.

The two-dimensional cell average $\bar{w}$ of a piecewise smooth function w is defined to be

$$\bar{w}_{jk} = \frac{1}{A_{jk}} \int_{y_{k-1/2}}^{y_{k+1/2}} \int_{x_{j-1/2}}^{x_{j+1/2}} w(\xi, \eta)\, d\xi d\eta \quad , \tag{8.4a}$$

where

$$A_{jk} = (x_{j+1/2} - x_{j-1/2})(y_{k+1/2} - y_{k-1/2}) \tag{8.4b}$$

In the above, we have assumed a Cartesian grid. We can immediately evaluate the point values of the primitive function $W(x, y)$.

$$W(x, y) = \int_{y_0}^{y} \int_{x_0}^{x} w(\xi, \eta)\, d\xi d\eta \tag{8.5a}$$

by

$$W(x_{j+1/2}, y_{k+1/2}) = \sum_{ik=k_0}^{k} \sum_{ij=j_0}^{j} \bar{w}_{jk} \quad . \tag{8.5b}$$

Since $w(x,y) \equiv \frac{\partial^2}{\partial x \partial y} W(x,y)$, we apply interpolation to the pointwise values (Eq. 8.5b) of the primitive function $W(x,y)$ and then obtain a pointwise reconstruction by defining

$$R(x,y;\bar{w}) = \frac{\partial^2}{\partial x \partial y} H_r(x,y;W) \quad . \tag{8.6}$$

We note that the above procedures do not require uniformity of the mesh but only that the mesh is Cartesian and not curvilinear. The non-oscillatory nature of the reconstruction follows primarily from the nonoscillatory nature of the interpolation. We also note that in the above description, w represents the dependent variable or other quantity being interpolated.

Procedure 3 presents a FORTRAN subprogram to develop a table of coefficients to be used to efficiently compute two-dimensional reconstruction at the center of a cell. Procedure 4 presents a subprogram to perform 2-d reconstructions using the stencils generated in Procedures 1 and 2. Such reconstruction is necessary, for example, to compare numerical results for pointwise values at the center of the cell with analytic solutions. Similar reconstruction is required to evaluate q_L and q_R at Gaussian quadrature points along cell boundaries.

```
      subroutine rctble
      common/rc/dlprc(4,4)
      al=0.5
      do 20 k=1,4
      zk=k+al-1.
      do 20 j=1,4
      pr=1.
      sm=0.
      do 15 m=0,4
      if(m.eq.j)goto15
      anw=zk-m
      pr=pr*anw/float(j-m)
      sm=sm+1./anw
   15 continue
   20 dlprc(j,k)=pr*sm
      write(8,*)'dlprc'
      do 30 j=1,4
      write(8,*)j,(dlprc(j,k),k=1,4)
   30 continue
      return
      end
```

Procedure 3. Constructing table of reconstruction coefficients

```
      subroutine rcnst(u,uu,nx,ny)
      common/rc/dlprc(4,4)
      common/d/a,b,almdx,almdy
      dimension u(50,50),uu(50,50)
      dimension wgs(50,50)
      dimension ii(50,50),jj(50,50)
c
      call stencil(u,ii,jj,nx,ny)
c
      do 10 i=1,nx
      do 10 j=1,ny
      kj=jj(i,j)+1
      jbg=j-kj
      do 10 l=1,2
      sm=0.
      smu=0.
      do 5 jlg=1,4
      jv=jbg+jlg
      if(jv.gt.ny)jv=jv-ny
      if(jv.lt.  1)jv=jv+ny
      smu=smu+u(i,jv)
    5 sm=sm+smu*dlprc(jlg,kj)
   10 wgs(i,j)=sm
      do 20 i=1,nx
      do 20 j=1,ny
      ki=ii(i,j)+1
      ibg=i-ki
      sm=0.
      smu=0.
      do 15 ilg=1,4
      iv=ibg+ilg
      if(iv.gt.nx)iv=iv-nx
      if(iv.lt.  1)iv=iv+nx
      smu=smu+wgs(iv,j)
   15 sm=sm+smu*dlprc(ilg,ki)
      uu(i,j)=sm
   20 continue
      return
      end
```

Procedure 4. Two-dimensional tensor-product reconstruction

Section 9.0
SYSTEMS OF EQUATIONS

It is easy to construct ENO formulations for systems of equations such as the Euler equations. Often it suffices to treat each dependent variable in the fashion described earlier. For best results in situations involving moving discontinuities, it preferable to treat each wave field using the scalar ENO formulation. A few more implementation details are given here for convenience.

First we consider 1-d problems. For systems of equations, we choose one stencil for each wave field in each cell. Let R be the matrix of right-eigenvectors (the columns of R are right-eigenvectors r^p). Let L be the matrix of left-eigenvectors (the rows of L are the left-eigenvectors l^p). For convenience in what follows, let $RL = I$, where I is the identity matrix. The individual left and right eigenvectors can always be suitably normalized to achieve this identity. In order to decide which stencil to use for wave-field p, we compare the dot products of l^p with the two choices for the divided differences of the vector of dependent variables. For each wave field, the comparison is therefore of two scalar quantities.

For the one-dimensional Euler equations, there will be three stencils defined in each cell, one for each local characteristic variable defined by $L_j \cdot \bar{w}_j$. Using these three stencils, it is possible to reconstruct three sets of vector valued states to the left and right of each cell interface. Denote the p-th set defined at the left face of cell j by $w^{p+}_{j-1/2}$ and denote the p-th set defined at the right face of cell j by $w^{p-}_{j+1/2}$. Let $w^{+}_{j-1/2}$ and $w^{-}_{j+1/2}$ be the values to be used in the Riemann Solver. We construct these from the individual sets corresponding to each wave field using

$$
\begin{aligned}
w^{+}_{j-1/2} &= \sum_{p=1}^{3} \left(l^p_j \cdot w^{p+}_{j-1/2} \right) r^p_j \\
w^{-}_{j+1/2} &= \sum_{p=1}^{3} \left(l^p_j \cdot w^{p-}_{j+1/2} \right) r^p_j
\end{aligned}
\tag{9.1}
$$

For the 2-d Euler equations, we can apply the same procedure direction by direction since we are using a tensor-product approach in this report.

Section 10.0
SOLUTION PROCEDURE

For most solutions presented in this report, the fourth-order accurate Runge-Kutta method was used to advance the solution in time using the semi-discrete approaches presented in previous sections. Some solutions for the 1-d scalar wave equation were obtained by other methods such as reconstruction by deconvolution, fully-discrete formulations, etc., but are presented here for completeness.

For 2-d problems, the cell boundary integrals of Eq. (3.9) were evaluated using Gaussian quadrature. A two point quadrature was used for each of the four sides for the fourth-order accurate results. Tables of interpolation coefficients were computed, similar to the presentation in Subroutine Rctble of Procedure 3 given in an earlier section to simplify the evaluation of first derivatives along the x and y directions at Gaussian quadrature points on cell boundaries. Another computational efficiency was achieved by computing the best stencil only once for each time step and using these for all stages of the Runge-Kutta scheme.

For the examples to be presented, only the fixed and periodic boundary conditions were required. The FORTRAN procedures given earlier assume periodic behavior but can be easily modified for non-periodic cases. The procedures also assumed that no grid stretching was used in either coordinate direction. This specialization was useful for all the results presented but can be generalized so that the method can be applied to Cartesian grids with direction-by-direction stretching .

Section 11.0
ONE-DIMENSIONAL EXAMPLES

We now present several one-dimensional examples to illustrate various aspects of ENO schemes.

Polynomial Interpolation

Figure 6 illustrates the advantages of ENO polynomial interpolation. The highly oscillatory curve in Figure 6a was obtained by standard 6-th degree piecewise polynomial interpolation — in every cell, the same relative stencil was used independent of the data. The circle symbols are the discrete values sampled to obtain the piecewise polynomial interpolation. The other curve passing through the circles is the analytically defined function based on which the discrete values were obtained. In contrast, Figure 6b shows the piecewise polynomial interpolation obtained using the 1-d ENO reconstruction method described earlier in this report.

Wave Equation — Sine wave

In the next example, the linear wave equation is solved on a very coarse grid with only 6 intervals. A periodic boundary condition is used. Figures 7a-d compare the numerical solution obtained with various orders of accuracy with the analytic solution after the sine wave has moved one cycle through the grid. In these figures, the circular symbol denotes the cell average value of the dependent variable. The analytic solution is an exact sine wave. The piecewise polynomial interpolation is also shown in each cell. The left and right values at each cell interface are connected by a vertical line. The improvement in accuracy with higher-order ENO formulations is clearly seen. Figure 8 portrays a composite of such individual results.

Polynomial Interpolation

The previous example used smooth initial data and therefore was not really a good illustration of the ENO properties of the scheme. Now, we present results for the linear wave equation but with the initial data corresponding to the first example. Figures 9a-d show how increasingly higher-order ENO formulations improve accuracy without introducing spurious numerical oscillations.

Damping Random Oscillations

In the next example, we start with sine wave initial data but add random perturbations to it (with maximum magnitude 0.1). Figure 10a shows the basic sine wave profile along with the initial data, with the square symbols denoting the cell center values. Figure 10b shows the numerical solution obtained with a 4-th order accurate ENO formulation after 1 cycle through the mesh assuming periodic boundary conditions. Notice how the highly oscillatory behavior of the initial data has been damped out but the smooth mean profile is propagated without error.

Shock Tube — Sod's Problem

The next example uses the 1-d Euler equations with initial data proposed by Sod. Figures 11a-c show a comparison of the numerical solution (symbols) with the exact solution (solid line) for density. Figures 12a-c show a comparison of the velocity profiles. Note the improvement in accuracy from 1st to 2nd order accuracy and the marginal improvement beyond 2nd order — in this case, because of the relative lack of higher gradients except near the transition points. In this example, characteristic variable interpolation was used.

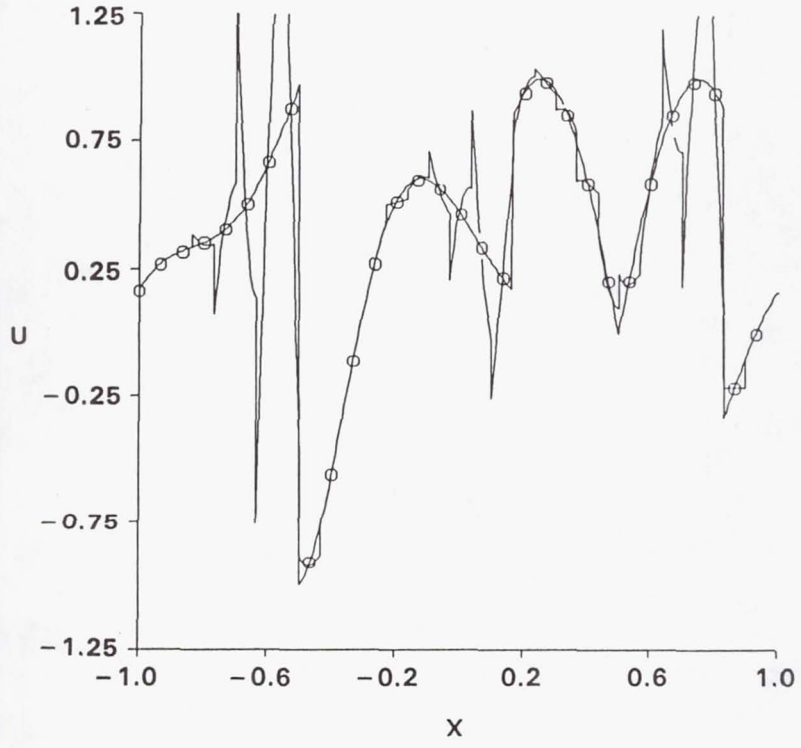


Figure 6a. Piecewise polynomial interpolation

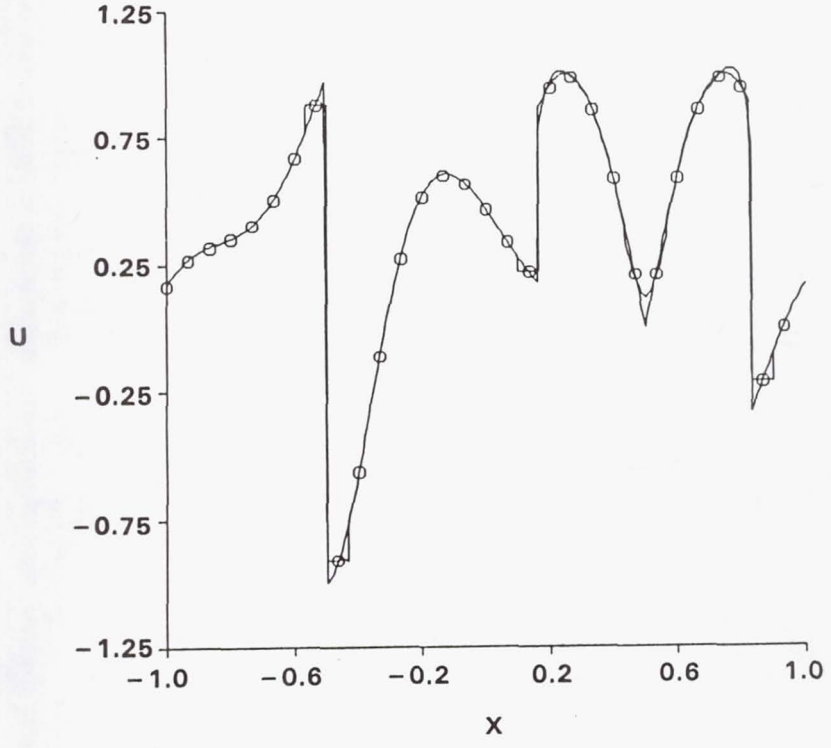


Figure 6b. Sixth-order non-oscillatory polynomial interpolation

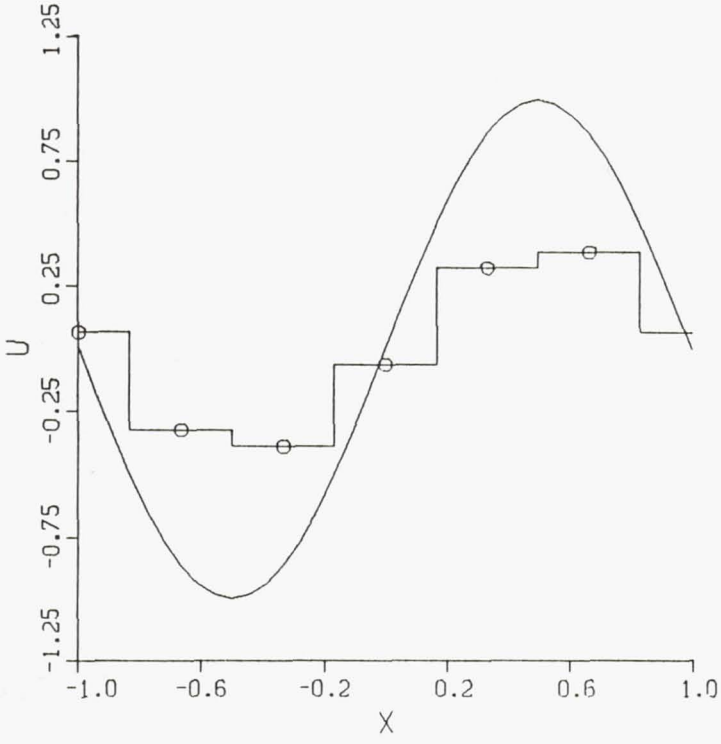


Figure 7a. First-order accuracy

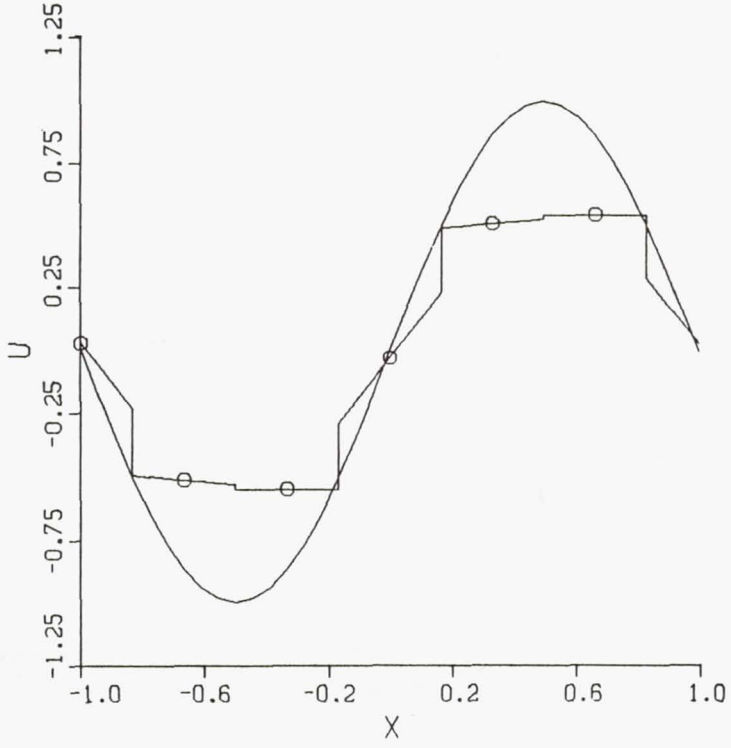


Figure 7b. TVD second-order accuracy

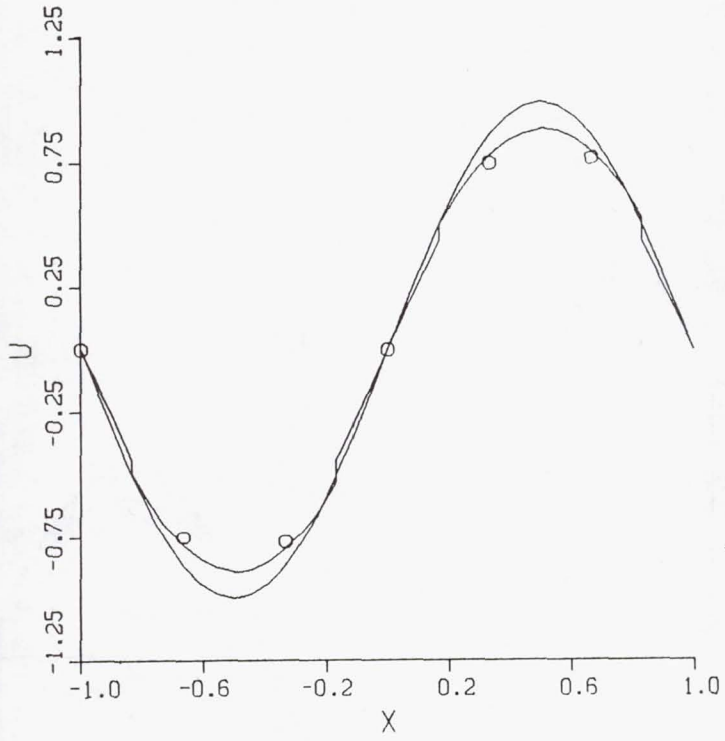


Figure 7c. ENO third-order accuracy

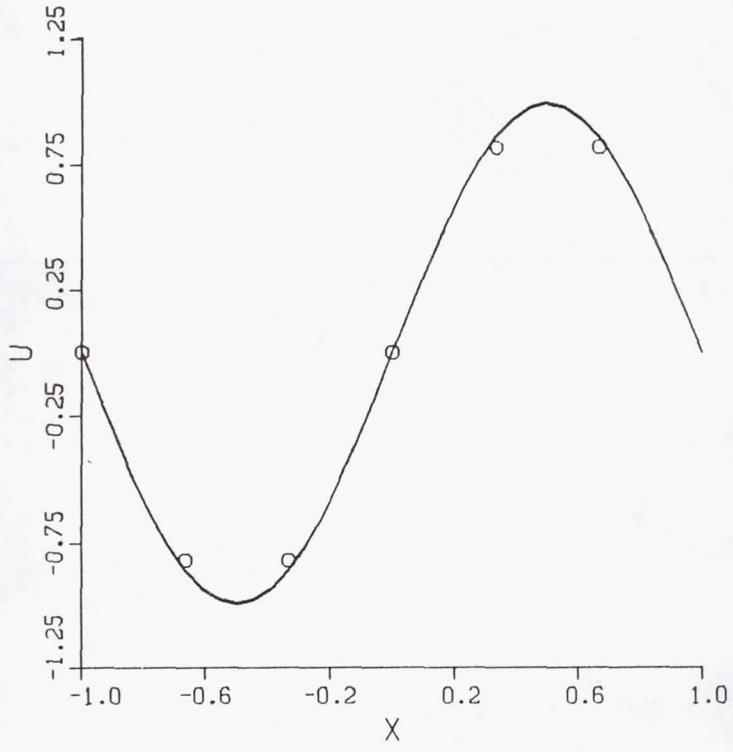


Figure 7d. ENO sixth-order accuracy

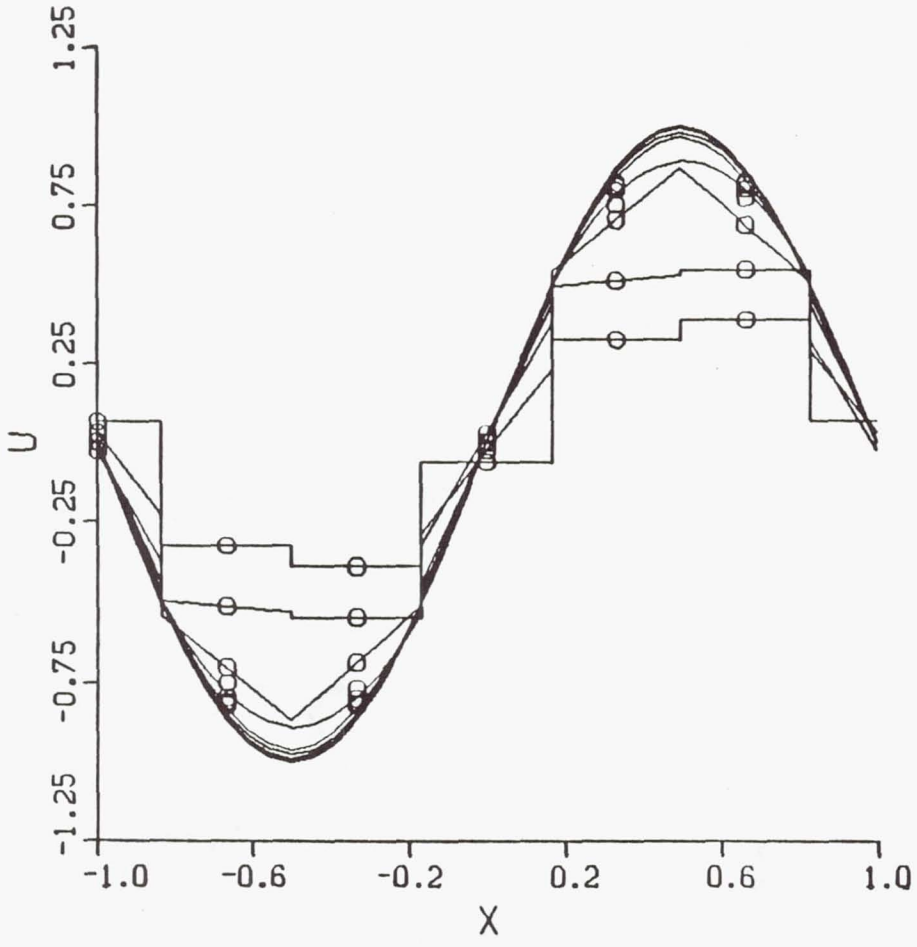


Figure 8. Comparative study of accuracy

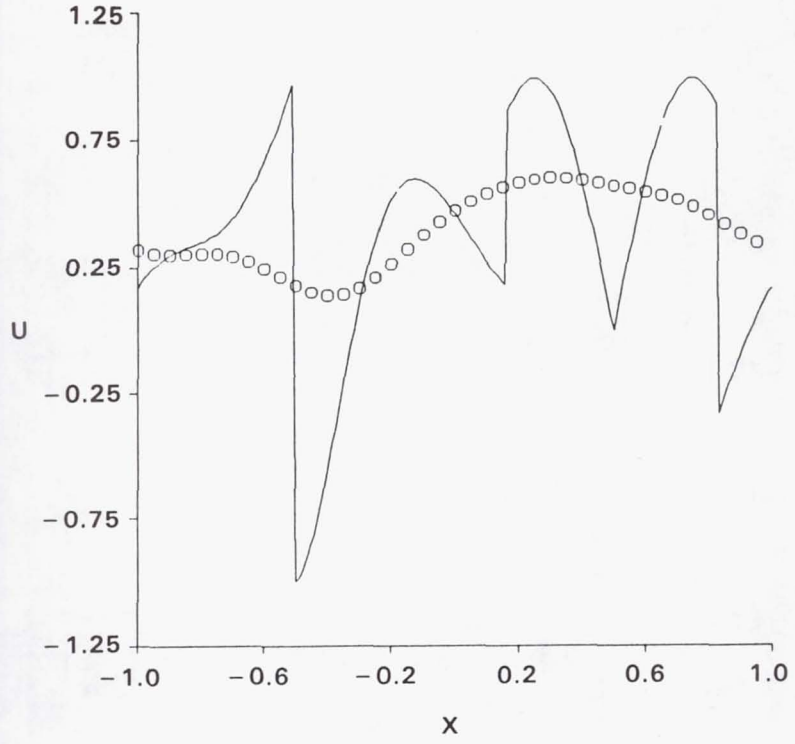


Figure 9a. First-order accuracy

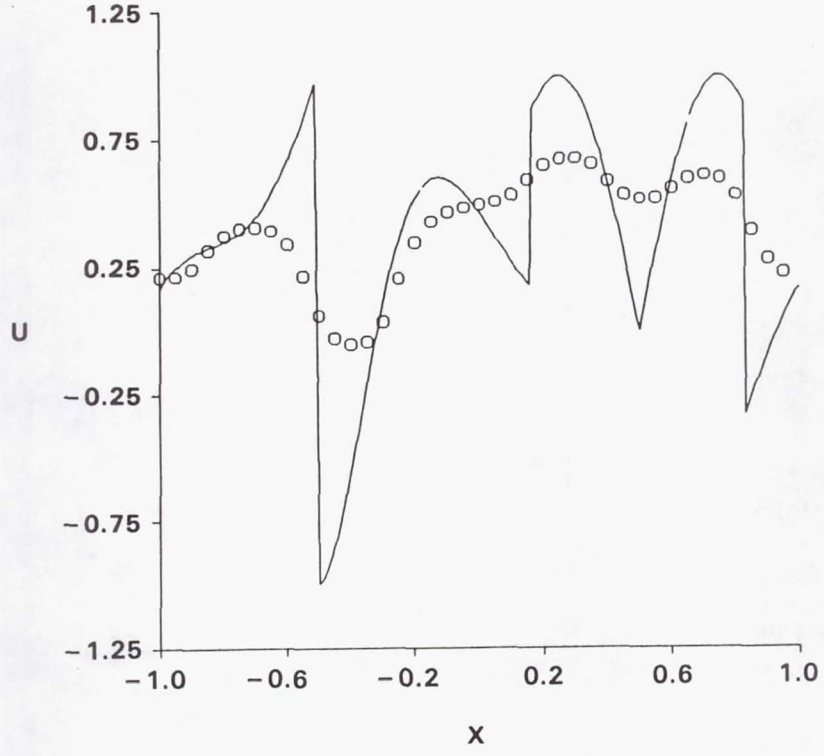


Figure 9b. TVD second-order accuracy

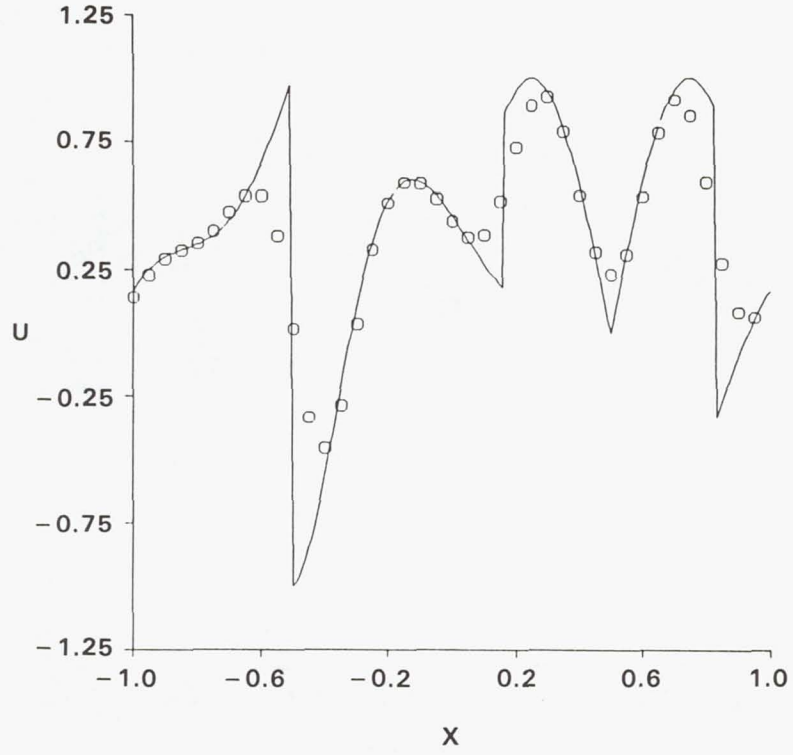


Figure 9c. ENO third-order accuracy

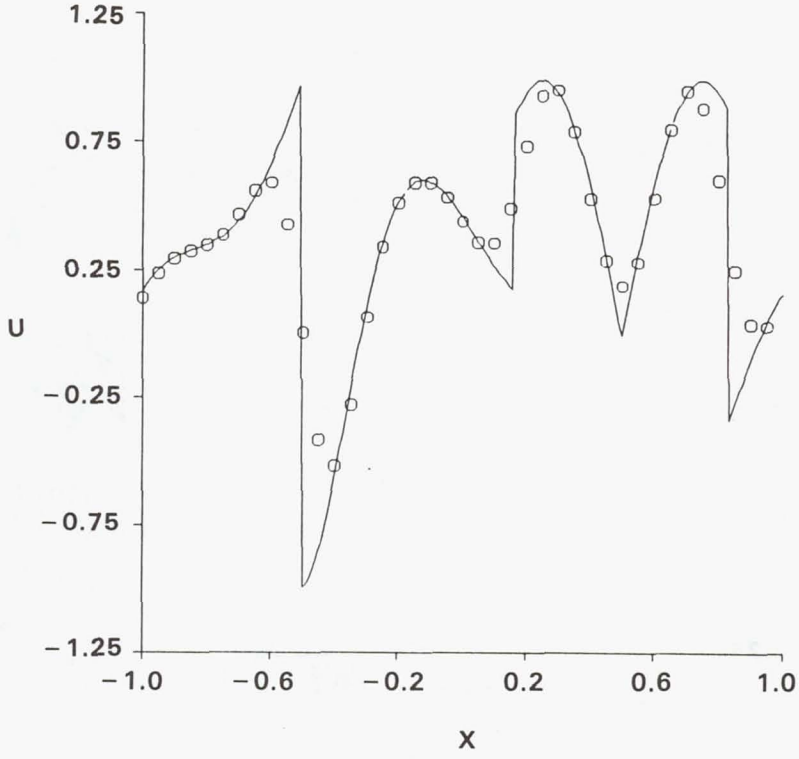


Figure 9d. ENO fourth-order accuracy

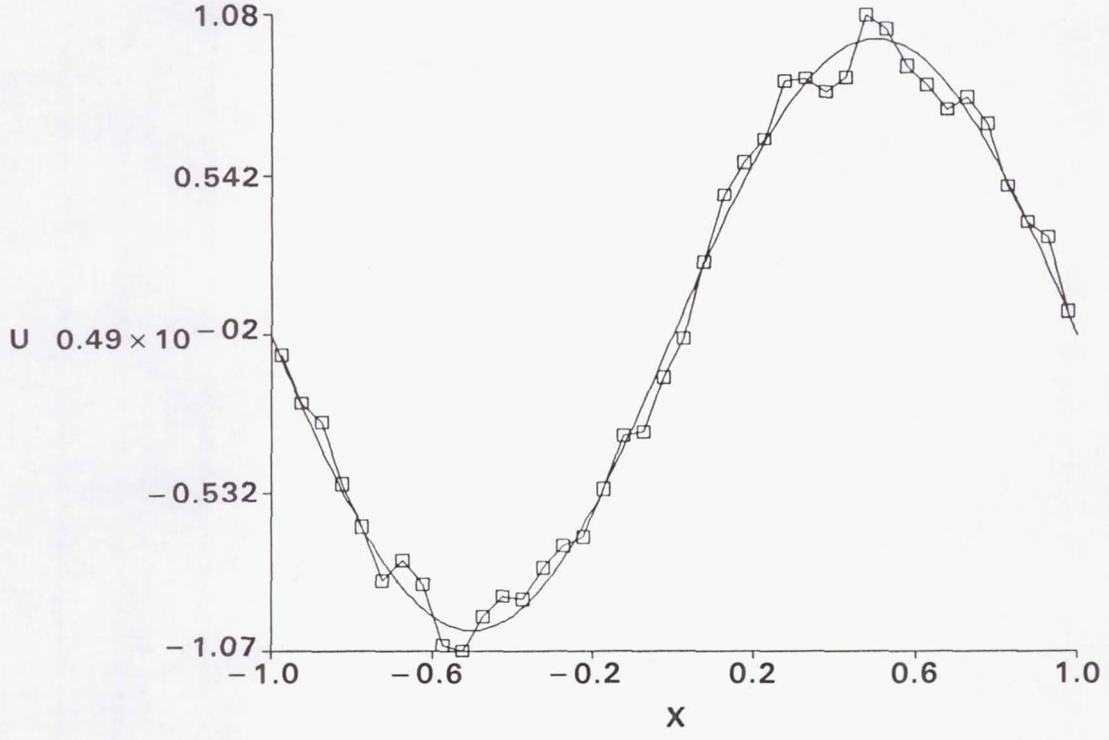


Figure 10a. Initial data — sine wave with random noise

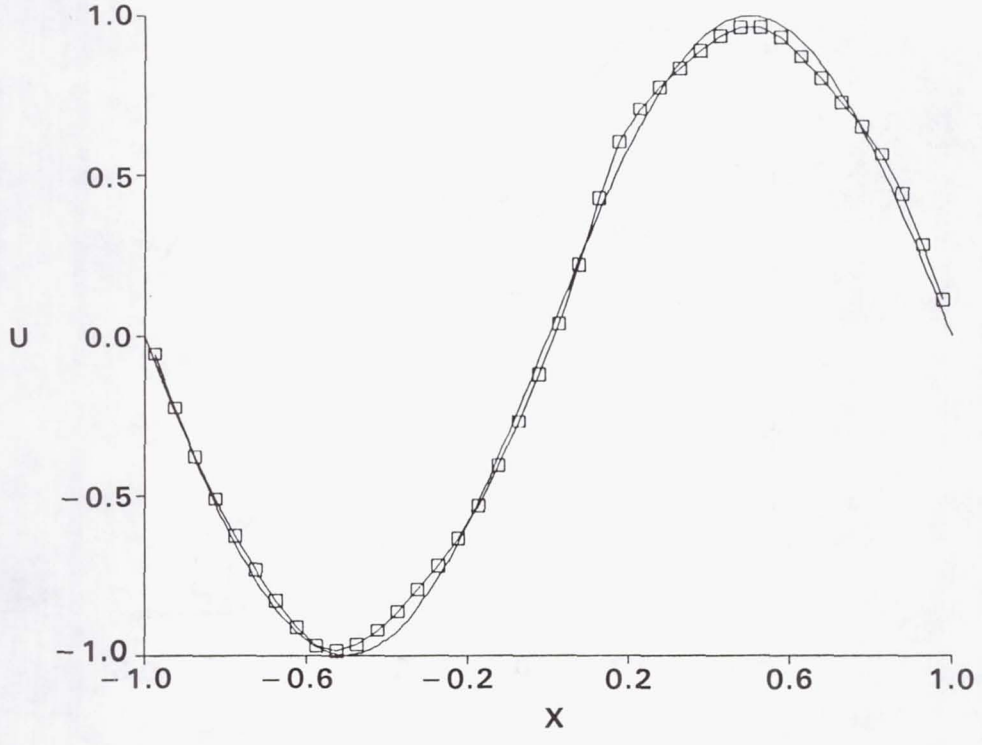


Figure 10b. Results after one cycle

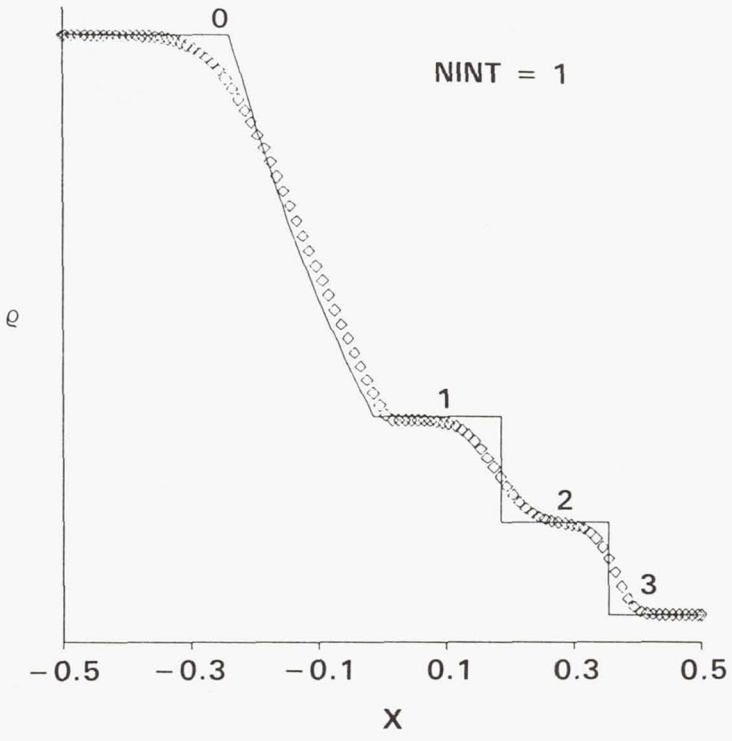


Figure 11a. Sod problem, density, first-order scheme

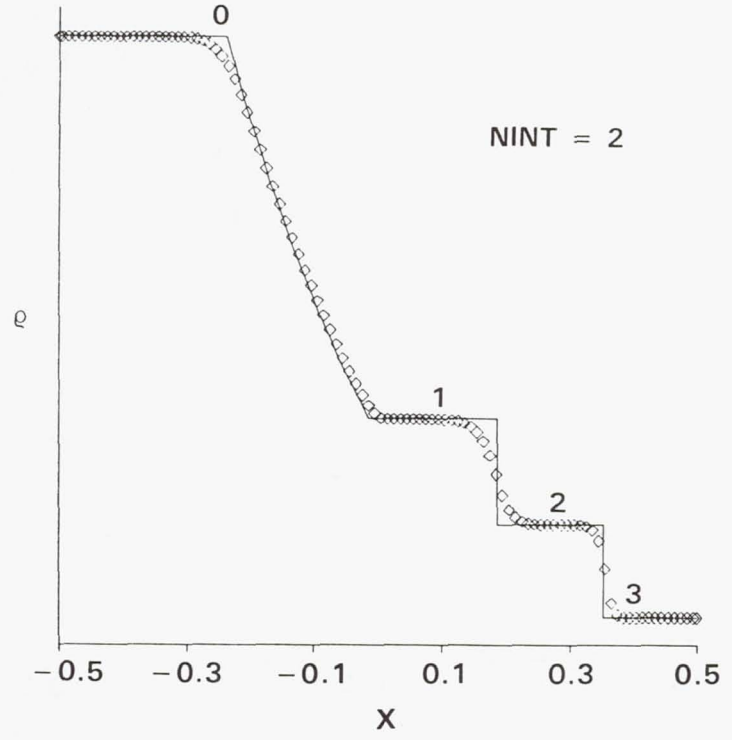


Figure 11b. Sod problem, density, ENO second-order scheme

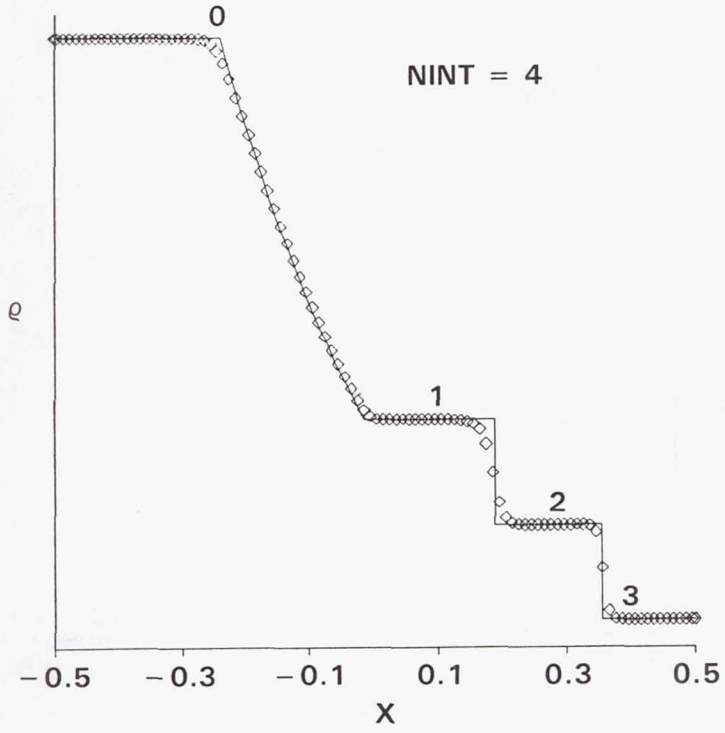


Figure 11c. Sod problem, density, ENO fourth-order scheme

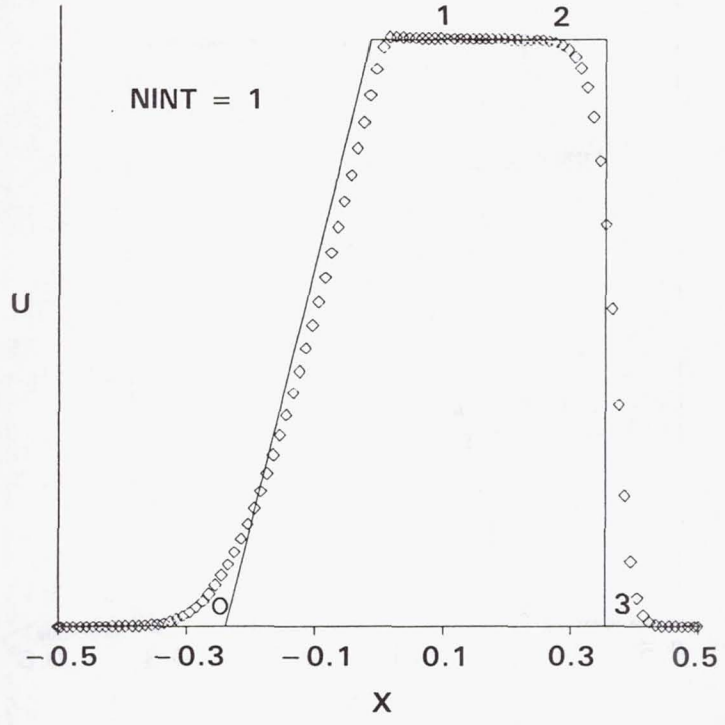


Figure 12a. Sod problem, velocity, first-order scheme

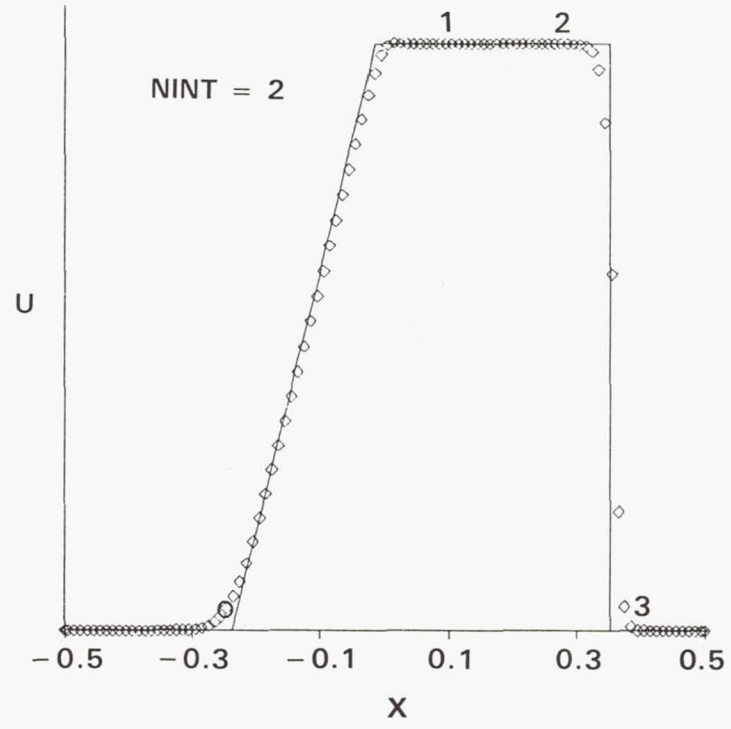


Figure 12b. Sod problem, velocity, ENO second-order scheme

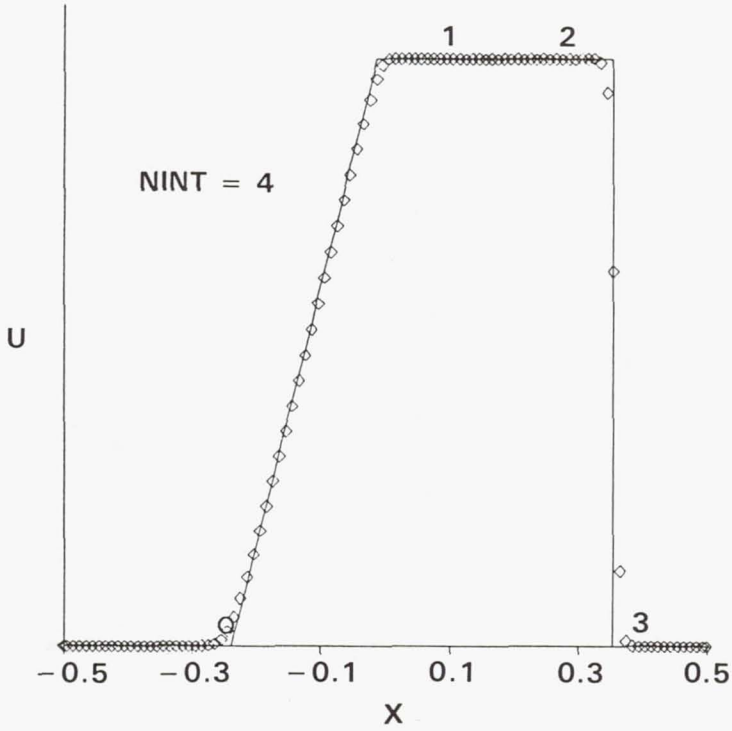


Figure 12c. Sod problem, velocity, ENO fourth-order scheme

Section 12.0
TWO-DIMENSIONAL EXAMPLES

We now present examples in two spatial dimensions.

Rotated Square Hat Problem

The two-dimensional scalar wave equation was solved on a uniform grid (20 x 20) with the wave speeds equal to unity in both directions. The exact solution corresponds to a translating initial data along the diagonal from lower left to upper right corner of the grid. Periodic boundary conditions were assumed in all four directions. The initial profile was a square hat, rotated by 45°; inside the square region, the dependent variables were assigned a value of unity and outside the region it was set to zero. Contour plots are shown for 10 snapshots at time-intervals of 10 steps in Figure 13. This example demonstrates the appropriateness of the 2-d interpolation procedure used.

Turbulence Amplification in Shock-Wave Interactions

A preliminary investigation was carried out to evaluate the applicability of the ENO formulation presented here to the problem of turbulence amplification in shock-wave interactions. The basic physical phenomena were described in Ref. 16 along with numerical results using a shock fitting procedure. The goal here was to perform an assessment of the feasibility of using shock-capturing schemes instead. The problem selected was the refreaction of a vorticity wave striking a Mach 8 shock at a 30 degree angle of incidence as illustrated in Fig. 2 of Ref. 16 (except that the incident wave is a vorticity wave rather than an acoustic one).

For the calculations presented in this report, upstream and downstream conditions (with respect to the shock wave) were chosen such that the shock wave is stationary. The flow velocity on the upstream side (right hand side) of the shock wave is from right to left with a Mach number of 8. The flow on the downstream side is subsonic. A vorticity perturbation is added to the supersonic side. The perturbation velocity components are denoted in Figure 14 as u' and v'. The geometry parameters are defined such that an entire period of the perturbation velocity profile fits the region between y_{max} and y_{min}. The upstream flow is held constant for the duration of the computations. The downstream flow feels the effect of the vorticity perturbations and over a period of time, these effects propagate to the left of the shock wave. A uniform grid in both the coordinate directions

was assumed. Periodic boundary conditions were imposed connecting the upper and lower boundaries. The flow field was held fixed at the unperturbed subsonic conditions at the left boundary. The computational grid used 101 points in the x direction and 61 points in the y direction. The upstream values of pressure and density were used as reference conditions to nondimensionalize all appropriate quantities. In order to specify the magnitude of the vorticity perturbation, u_{ref} (see Fig. 14) was chosen to be the upstream velocity's magnitude and the intensity selector ϵ was chosen to be 0.001.

Three sets of results are now presented in Figs. 15-17. Each set comprizes a) pressure contours in the range 74.46 to 74.56, b) vorticity contours, and c) vorticity profiles along the horizontal direction for every horizontal grid line. Figs. 15a-c display results at a value of nondimensional time, τ, of 0.094. Figs. 16a-c are for $\tau = 0.153$ and Figs. 17a-c are for $\tau = 0.2$. The smoothness of the contours in the contour plots demonstrates the utility of the ENO shock capturing scheme for this problem. The vorticity contours lie between the maximum and minimum values displayed in the figures presenting vorticity profiles. The shock wave location is at $x = 0.0$. The profiles show the amplification of vorticity occurring downstream of the shock wave.

The results shown in Fig. 17c are redrawn in Fig. 17d in a condensed form. To create Fig. 17d, the maximum absolute value of vorticity along each vertical grid line at each x location was computed and plotted as a function of x. Based on the harmonic behavior in the y direction displayed in Fig. 17c, this approach is a reasonable alternative to computing and plotting the Fourier coefficients at each x location. If this plot is compared with Fig. 4 of Ref. 16, the good agreement between the two results becomes evident. The two results, however, use different scaling. The region of non-zero vorticity in both results is clearly seen to extend 0.5 units along the axial direction downstream of the shock position. The shape of the numerical profile on the downstream side of the shock wave is almost identical. The present shock-captured results also show vorticity through the numerical shock layer and on the upstream side of the shock wave. The values of vorticity inside the numerical shock layer cannot be usefully interpreted.

In the above computations, the ENO interpolation procedure was applied to the characteristic variables and not the conservation variables. The next two sets of figures illustrate the need for this implementation. These results were obtained using initial conditions (upstream and downstream) chosen so that the shock wave is moving to the right with $M_s = 8.0$ (Mach number computed using shock velocity and upstream speed of sound). The perturbations were set to zero. Without the presence of any two-dimensional effects,

the same axial profiles were obtained at every vertical location (solution independent of y). Figs. 18a, 19a and 20a portray density, pressure and u-velocity profiles when conservative variable interpolation is used and Figs. 18b, 19b and 20b illustrate the numerical behavior when characteristic variable interpolation is used. The marked low frequency oscillations shown in Figs. 18a, 19a, 20a are not unexpected to researchers familiar with ENO formulations. The use of characteristic variables greatly reduces, but not eliminates, such undesirable behavior. It is yet to be determined whether such numerical effects may be obstacles to the application of shock-capturing ENO schemes, such as the formulation presented in this report, to high resolution applications such as turbulence amplification in shock-wave interactions when the shocks are moving with respect to the computational grid.

In summary, it has been shown that an appropriately applied ENO shock-capturing scheme can be used to study shock-wave turbulence interactions. Some issues related to capturing high Mach number moving shock wave have also been raised. This researcher is confident that recent advances such as "subcell resolution" techniques (Ref. 14) and evolving improvements to and variations of ENO schemes will resolve such problems. It must also be repeated here that the above study represents only preliminary work in applying a specific type of ENO formulation to shock-wave turbulence interaction problems and is therefore not definitive.

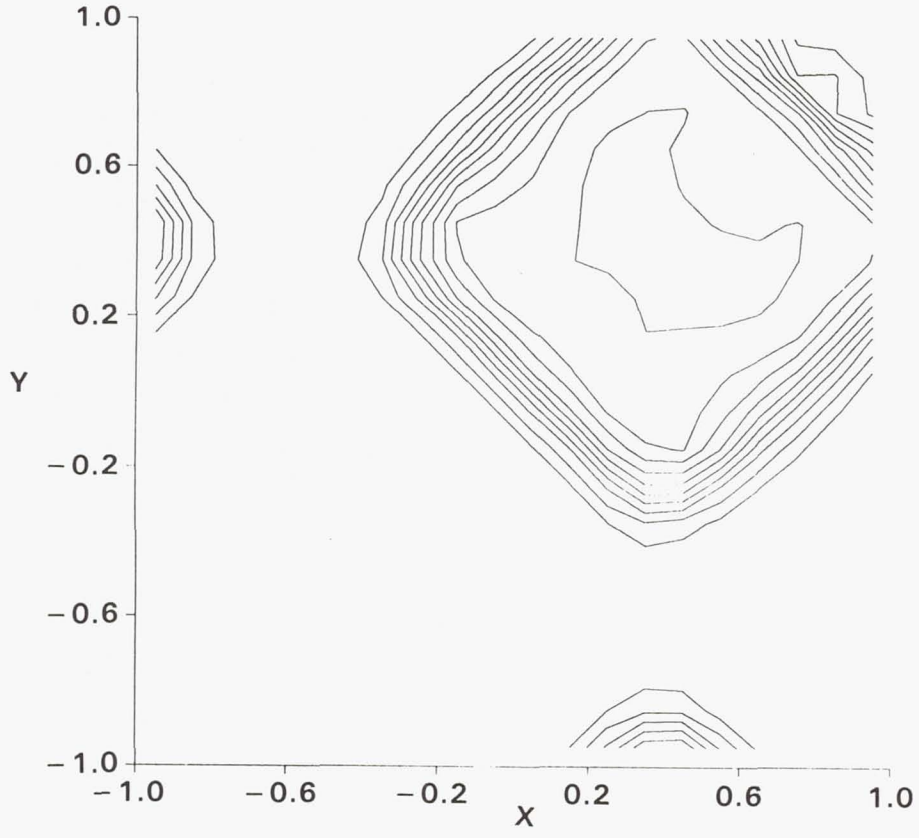


Figure 13a. Square hat, 10 time steps

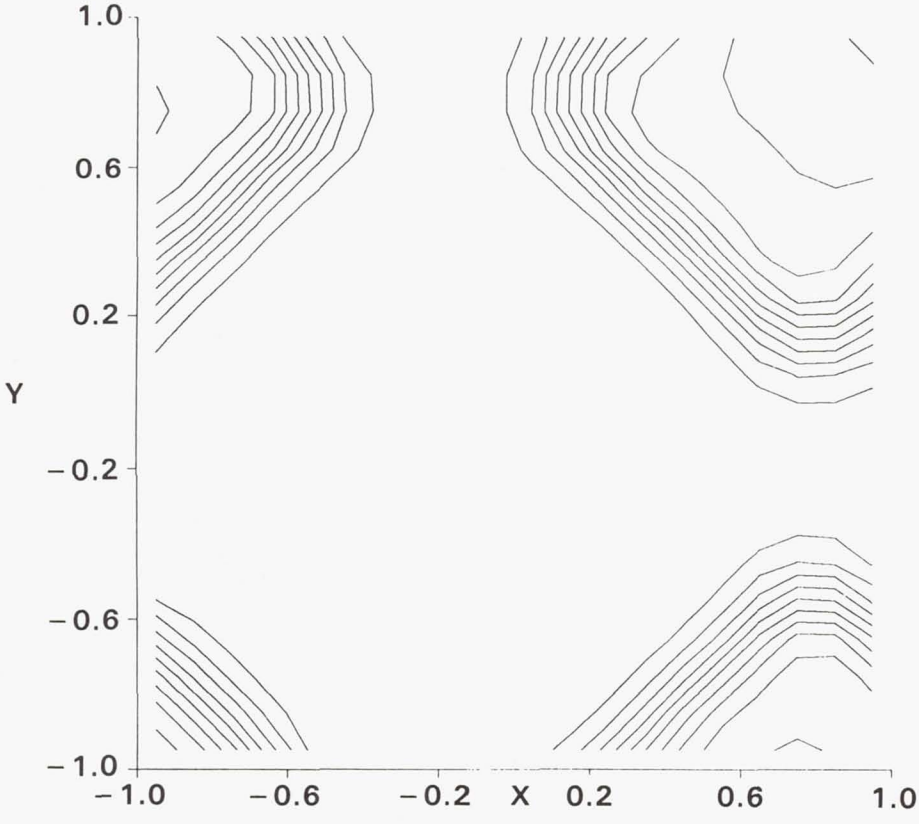


Figure 13b. Square hat, 20 time steps

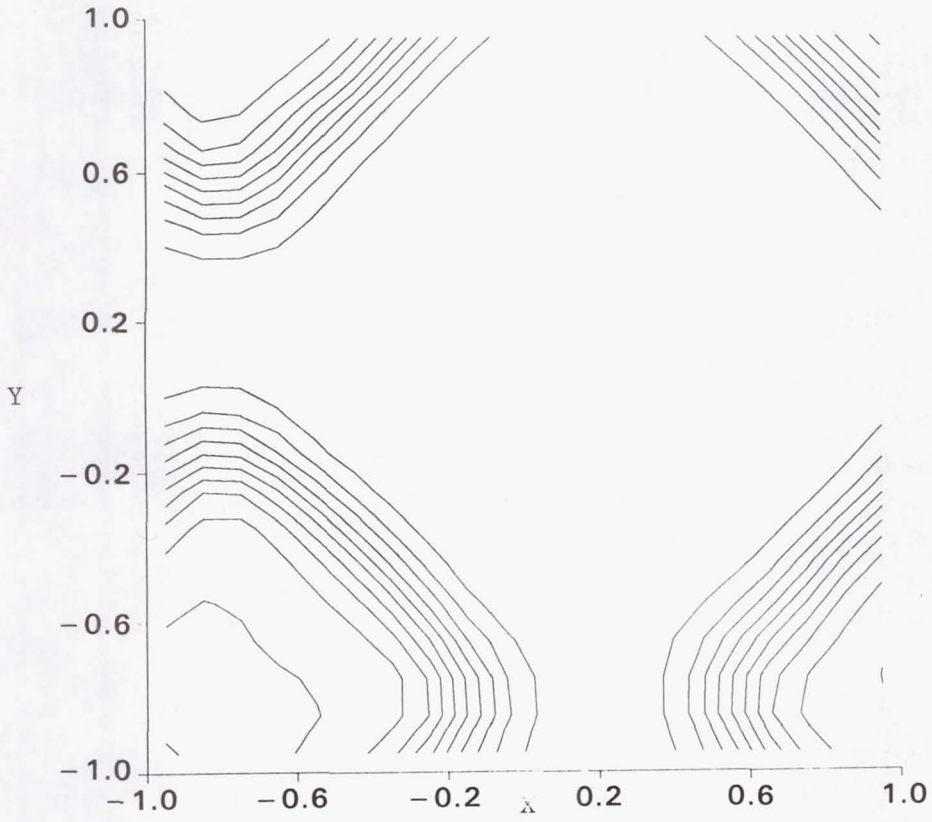


Figure 13c. Square hat, 30 time steps

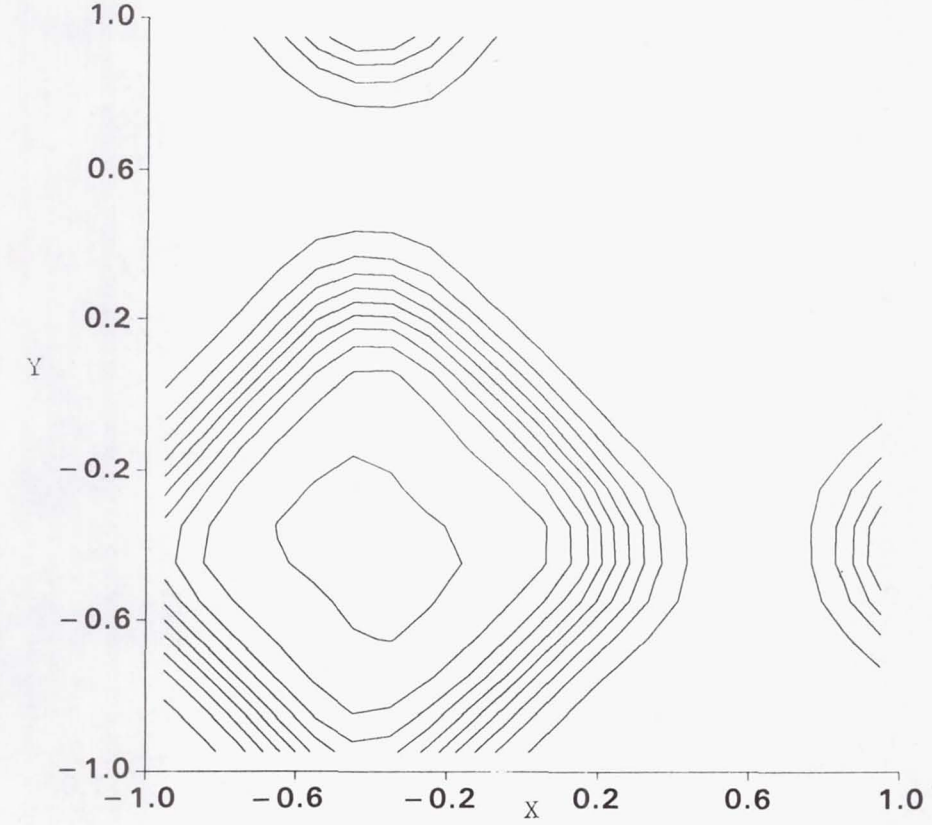


Figure 13d. Square hat, 40 time steps

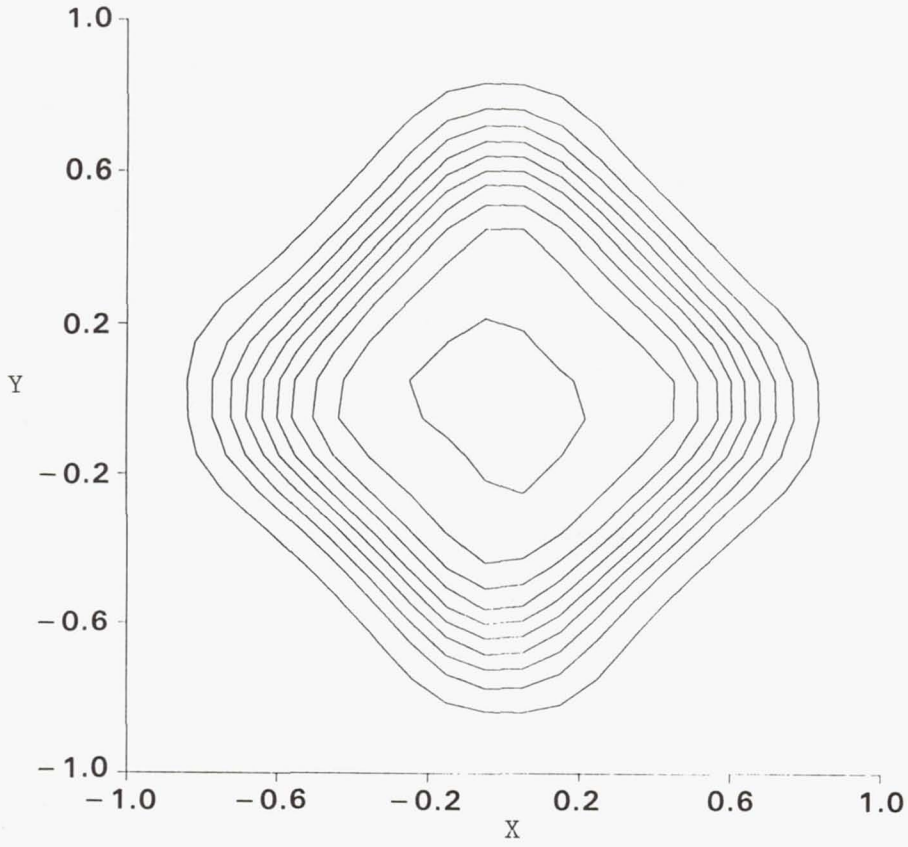


Figure 13e. Square hat, 50 time steps

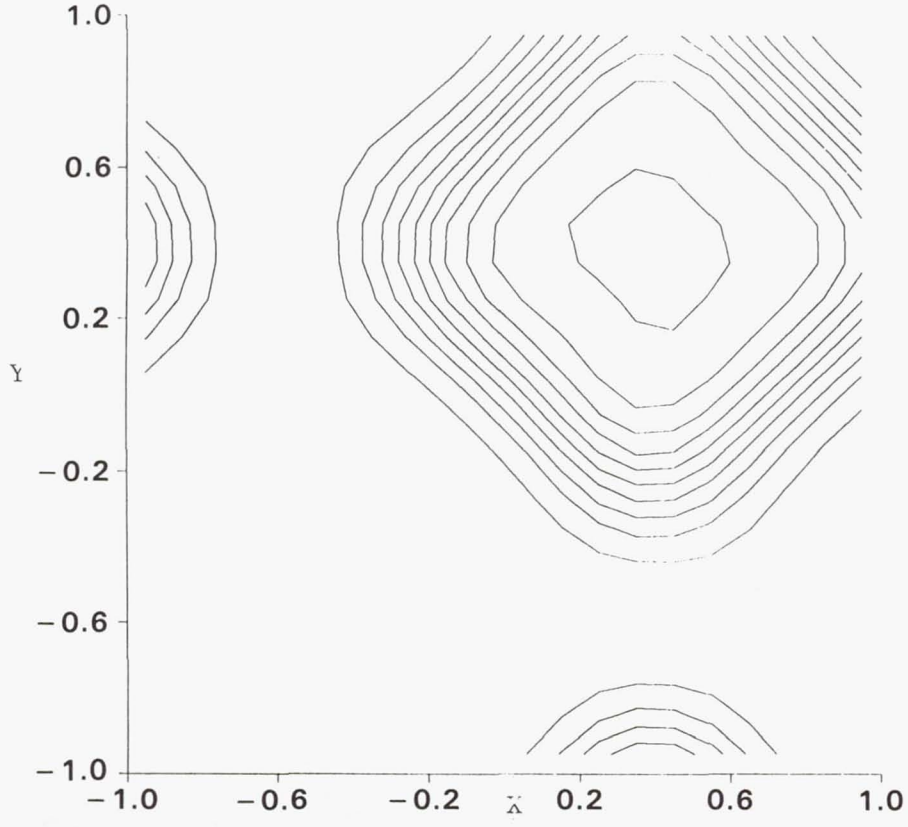


Figure 13f. Square hat, 60 time steps

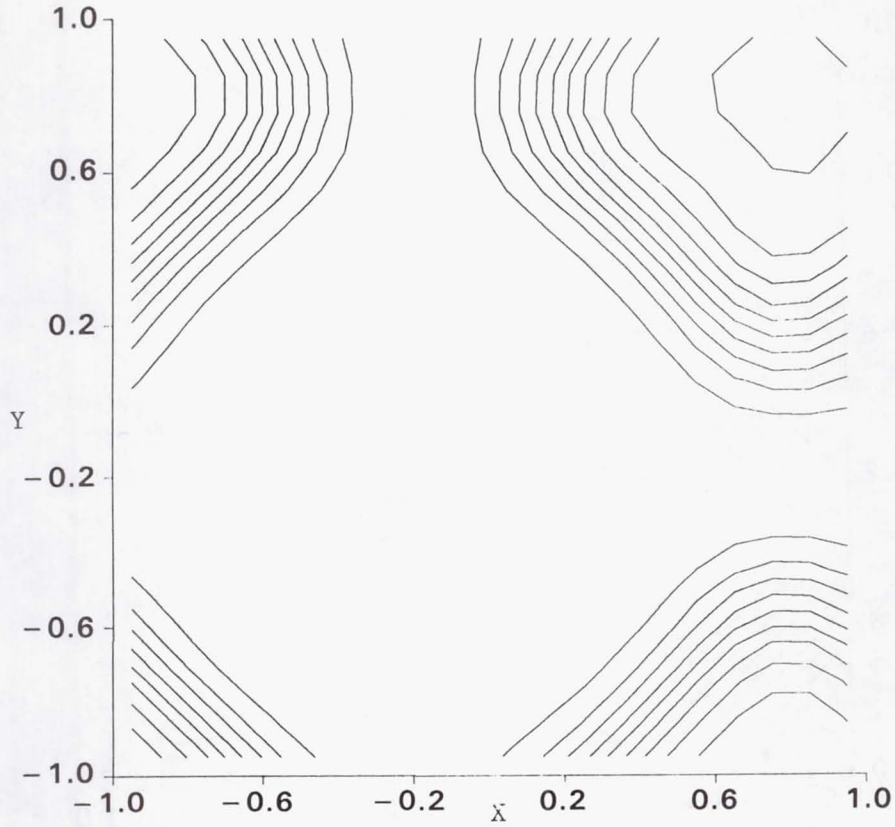


Figure 13g. Square hat, 70 time steps

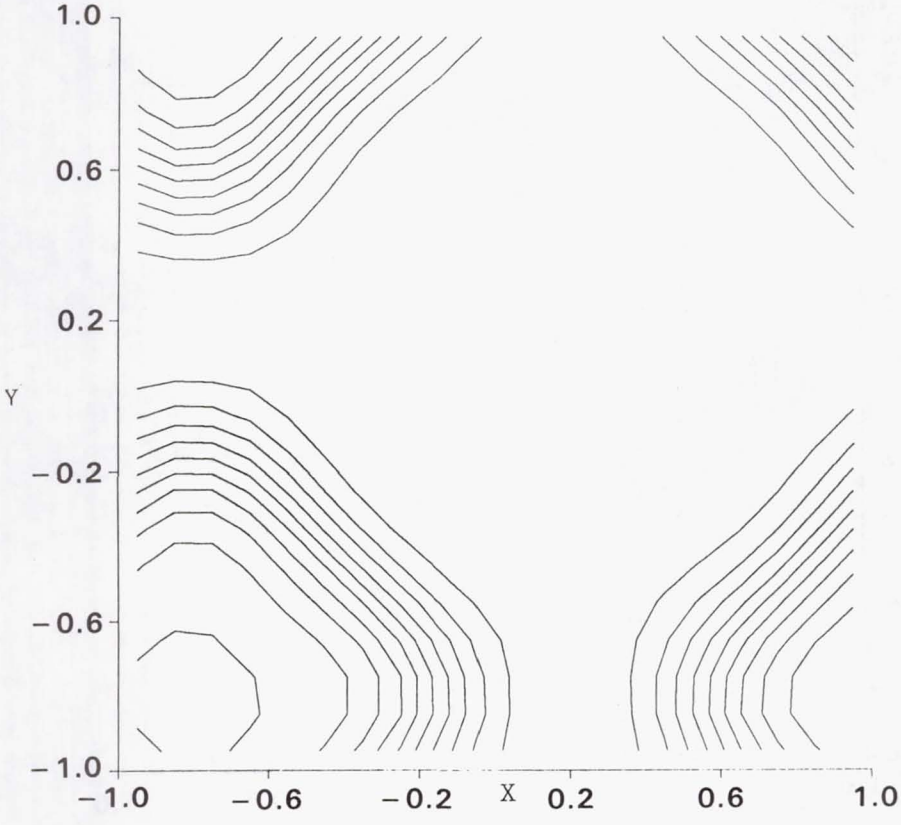


Figure 13h. Square hat, 80 time steps

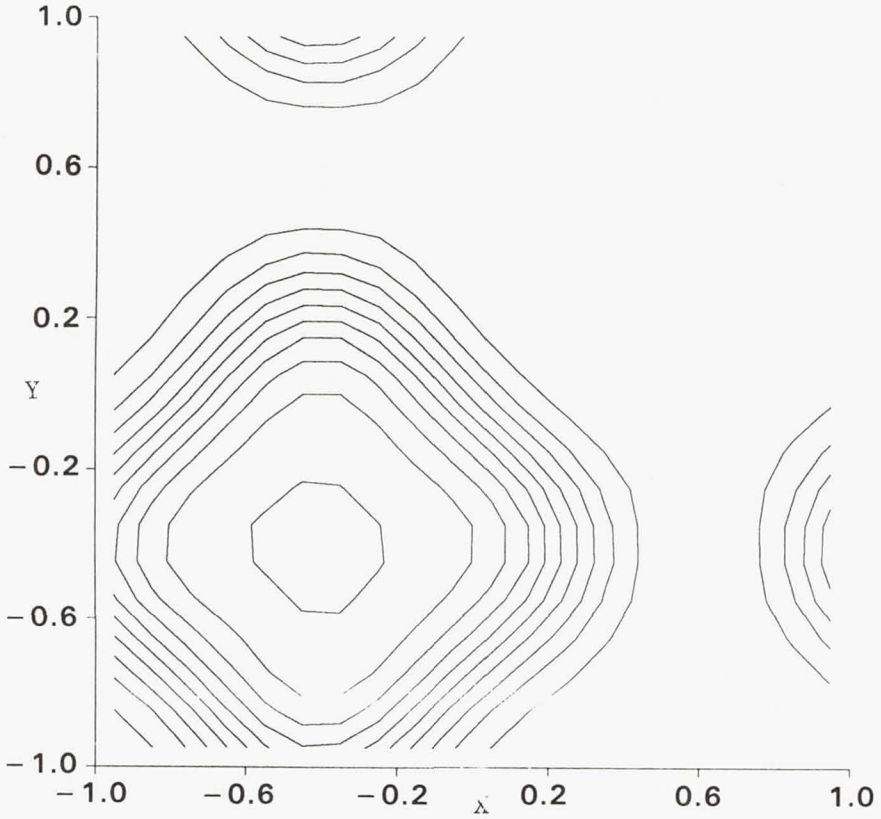


Figure 13i. Square hat, 90 time steps

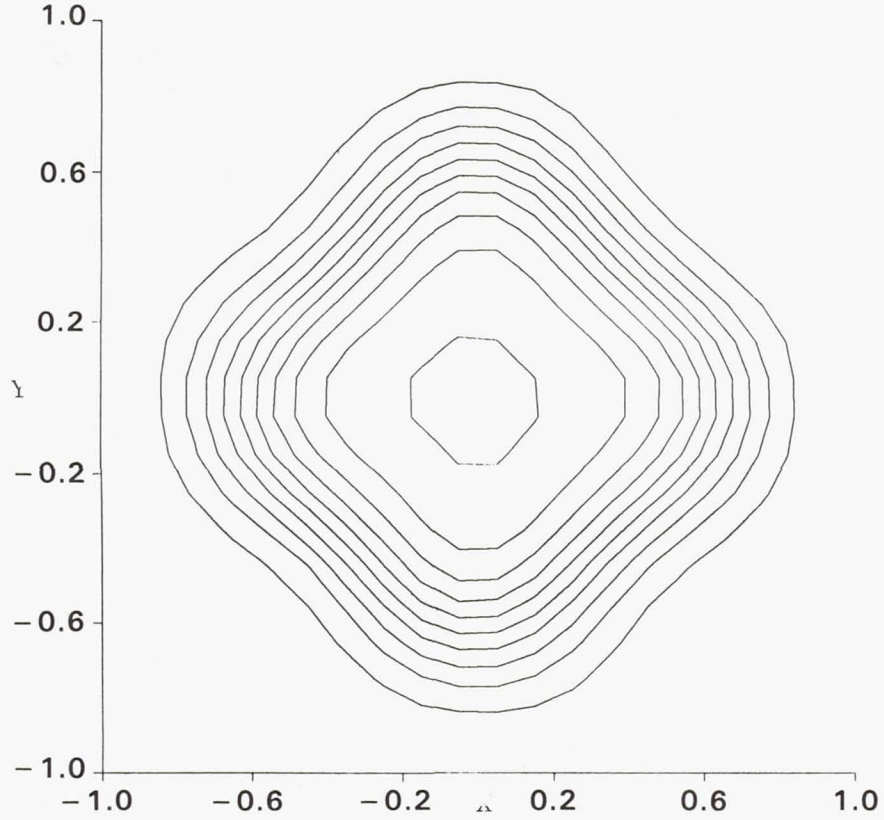
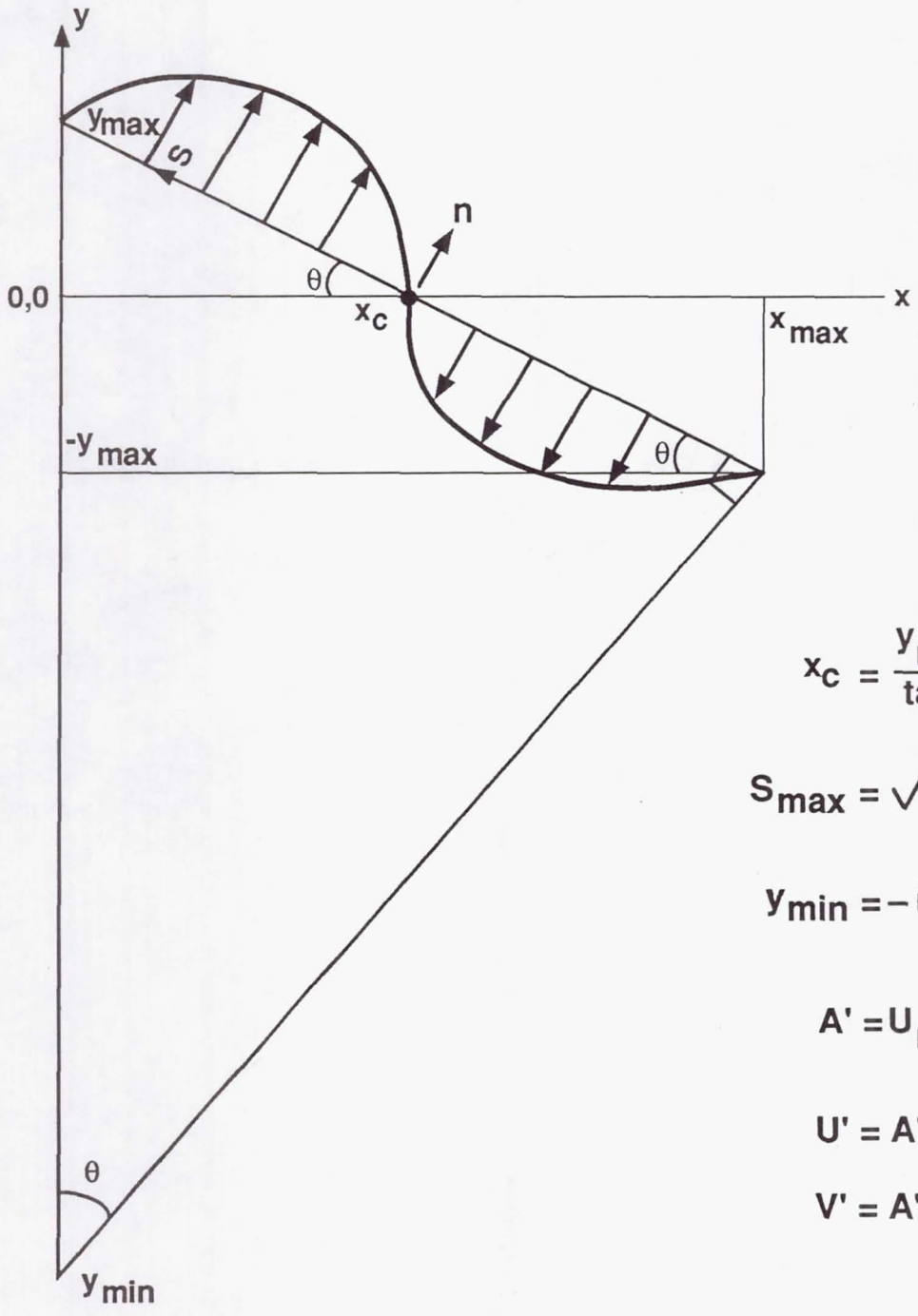


Figure 13j. Square hat, 100 time steps



$$x_c = \frac{y_{max}}{\tan\theta}$$

$$S_{max} = \sqrt{x_c^2 + y_{max}^2}$$

$$y_{min} = -\left(y_{max} + 2\,\frac{x_c}{\tan\theta}\right)$$

$$A' = U_{ref}\,\varepsilon\,\sin\left(\frac{S}{S_{max}}\right)$$

$$U' = A'\sin\theta$$

$$V' = A'\cos\theta$$

Figure 14. Specification of upstream boundary conditions

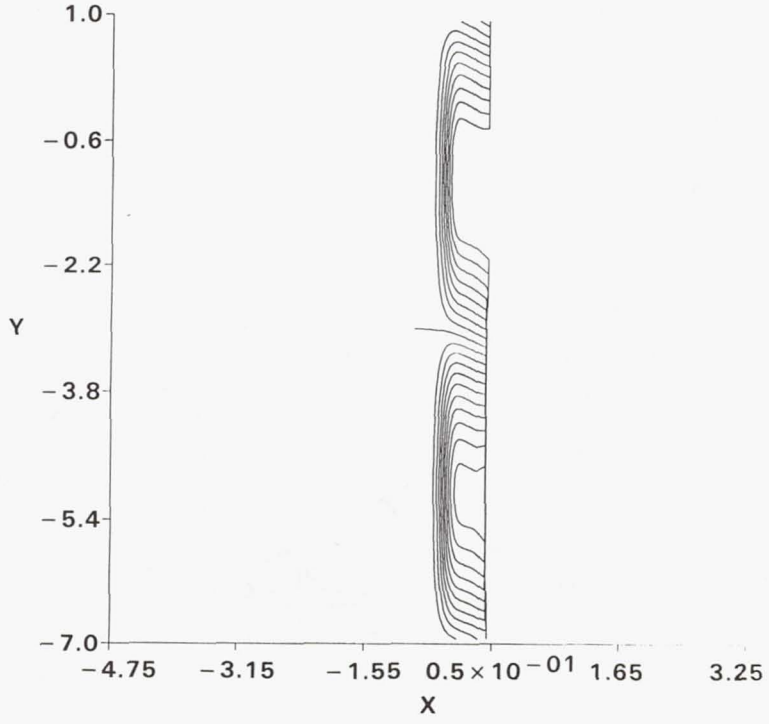


Figure 15a. Pressure contours at $\tau = 0.094$

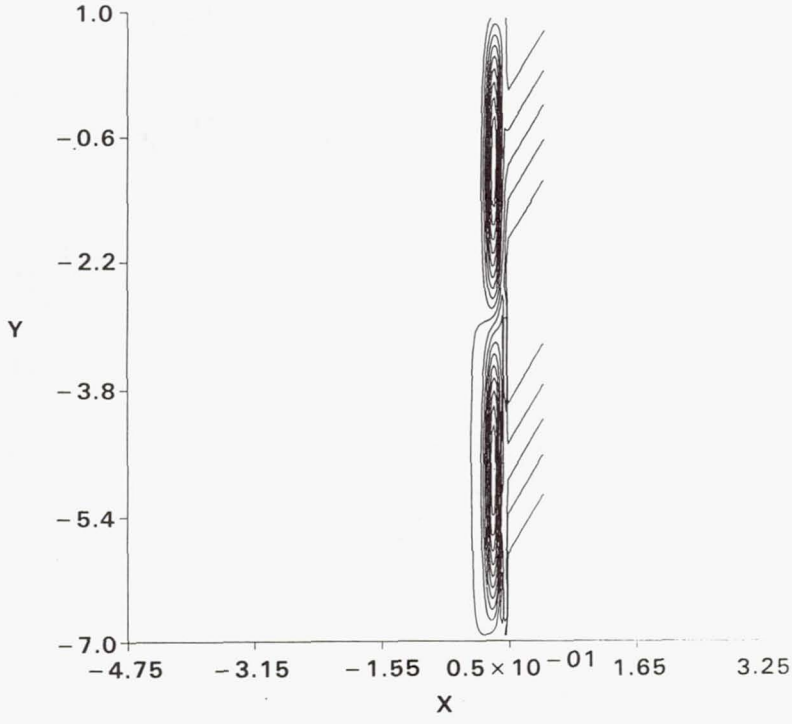


Figure 15b. Vorticity contours at $\tau = 0.094$

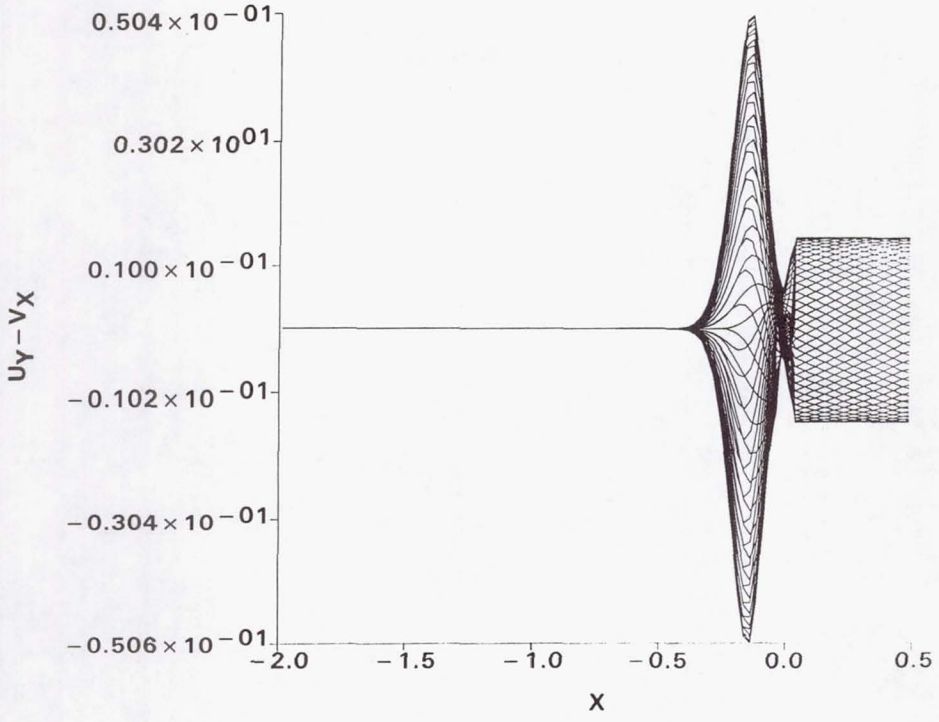


Figure 15c. Vorticity profiles at $\tau = 0.094$

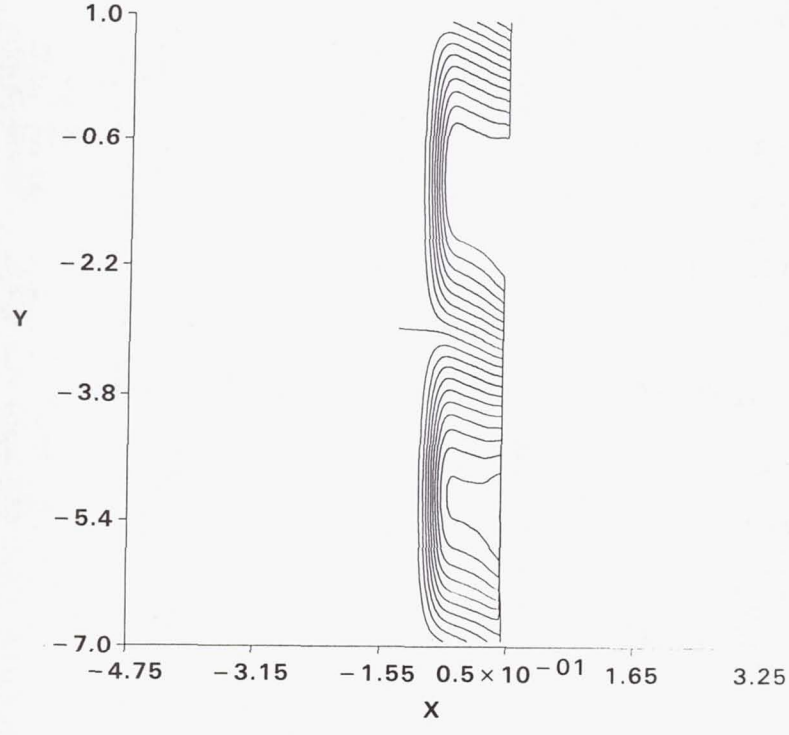


Figure 16a. Pressure contours at $\tau = 0.153$

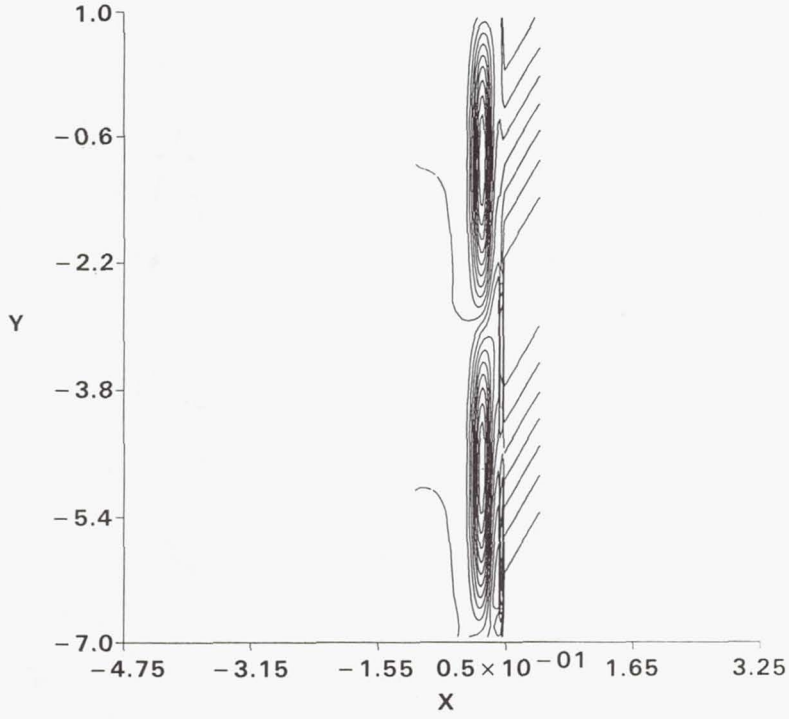


Figure 16b. Vorticity contours at $\tau = 0.153$

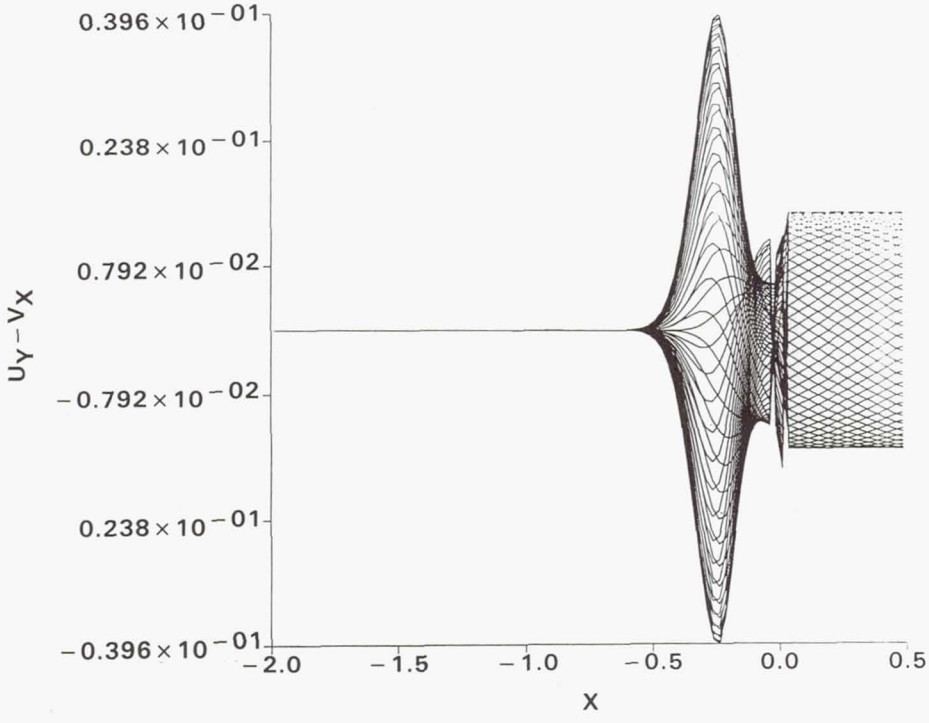


Figure 16c. Vorticity profiles at $\tau = 0.153$

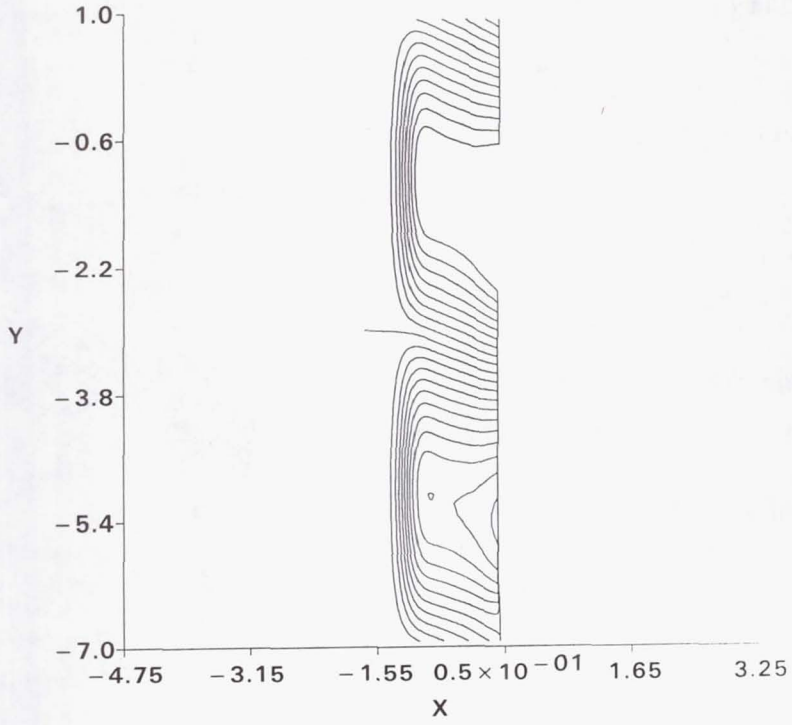


Figure 17a. Pressure contours at $\tau = 0.2$

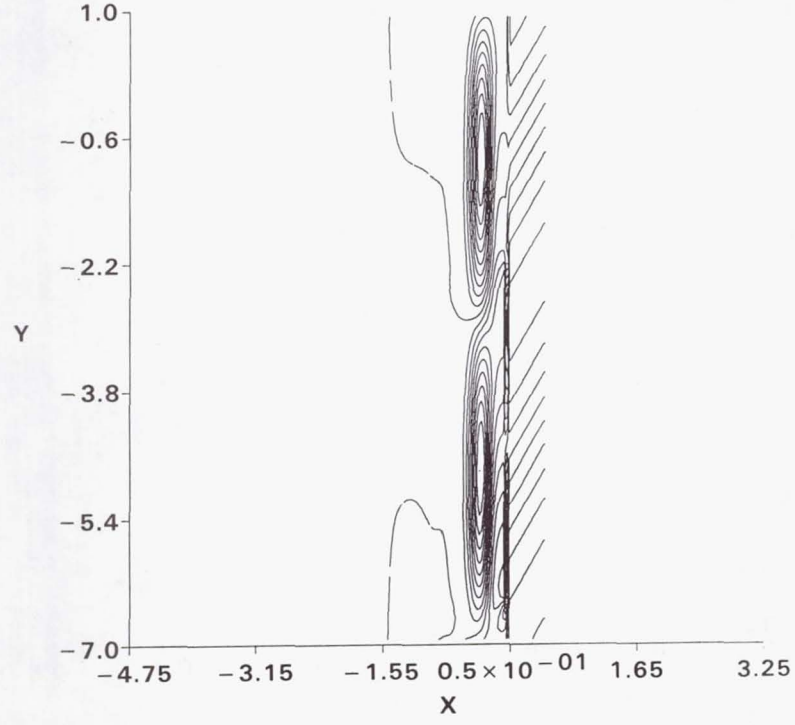


Figure 17b. Vorticity contours at $\tau = 0.2$

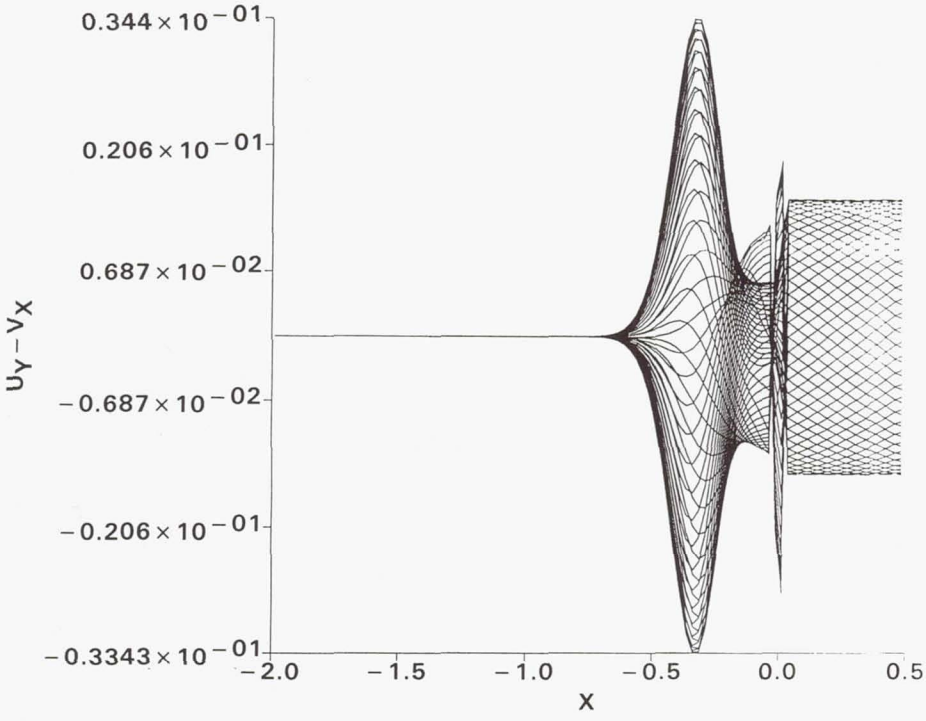


Figure 17c. Vorticity profiles at $\tau = 0.2$

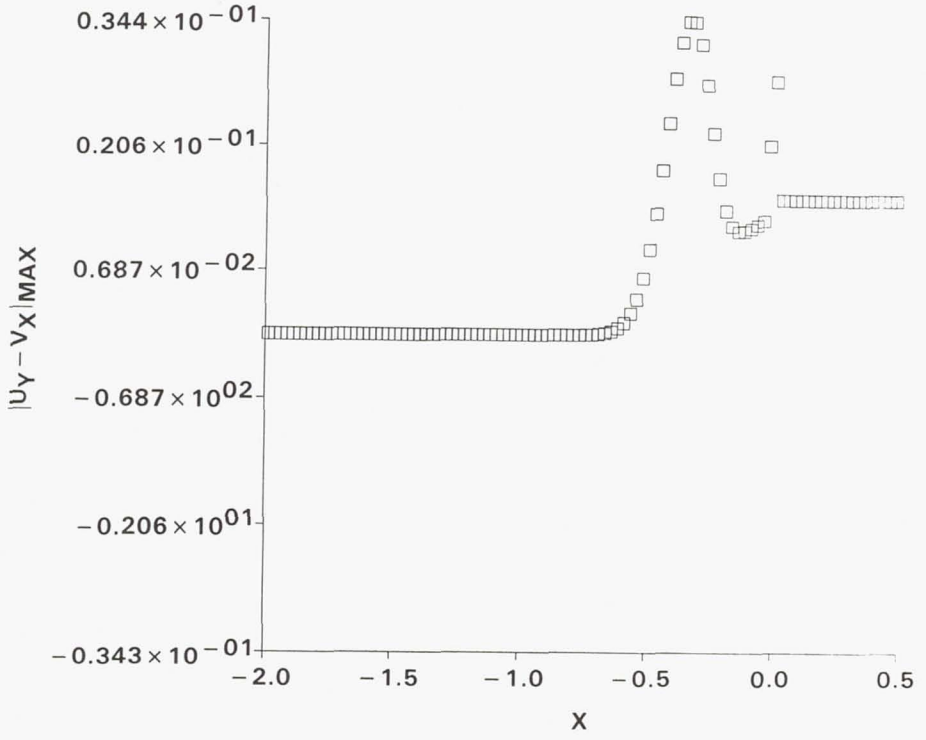


Figure 17d. Maximum absolute vorticity profile at $\tau = 0.2$

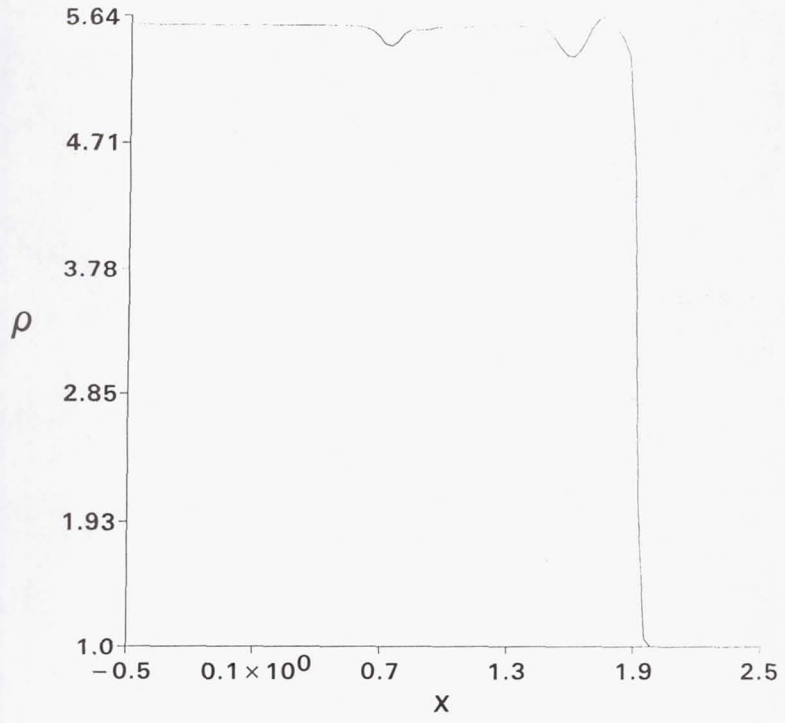


Figure 18a. Density profile with conservative variable interpolation

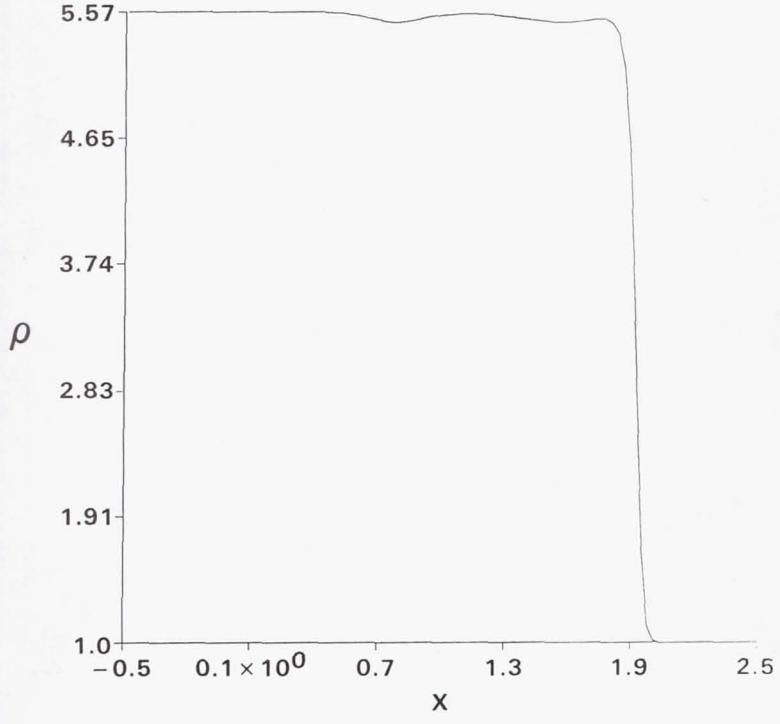


Figure 18a. Density profile with characteristic variable interpolation

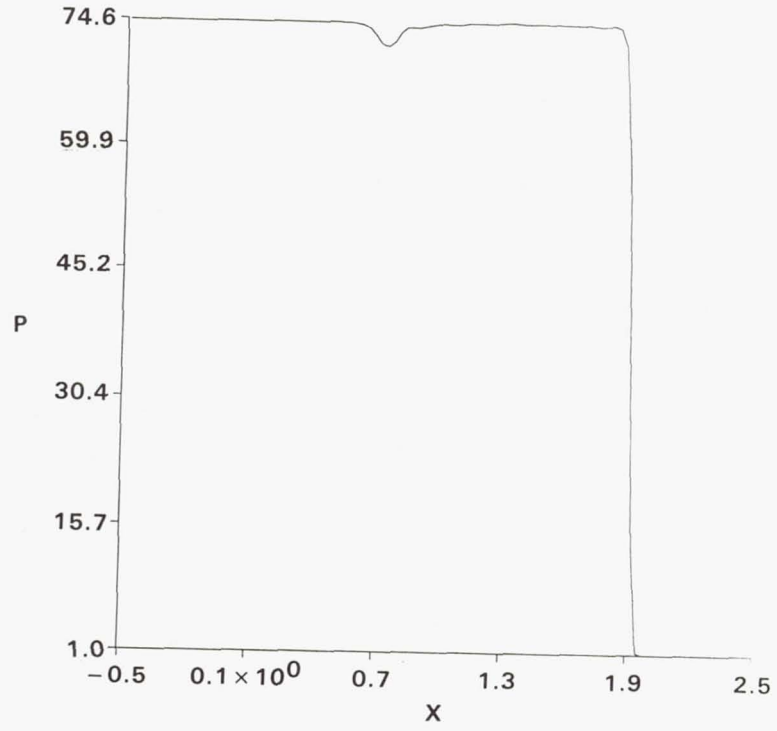


Figure 19a. Pressure profile with conservative variable interpolation

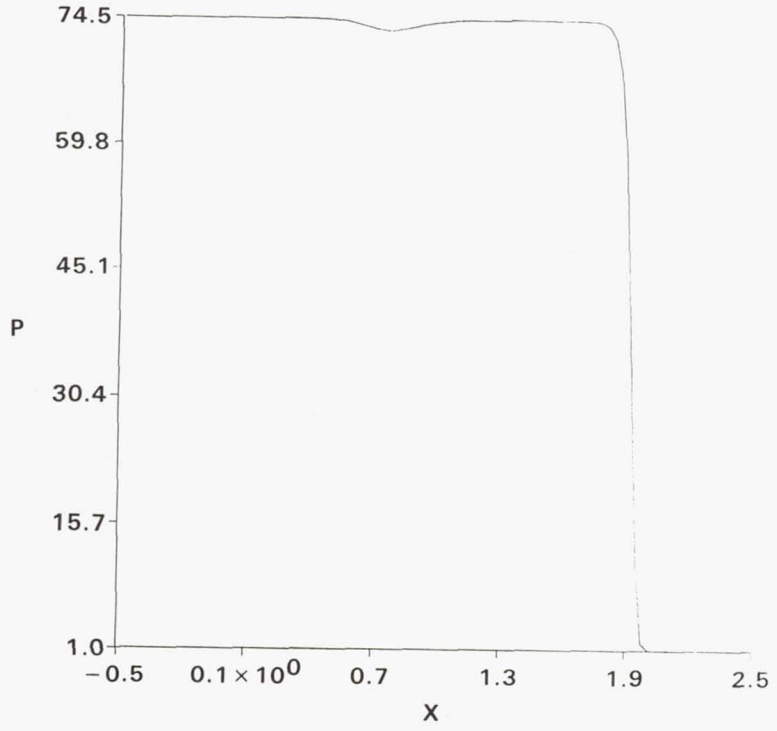


Figure 19a. Pressure profile with characteristic variable interpolation

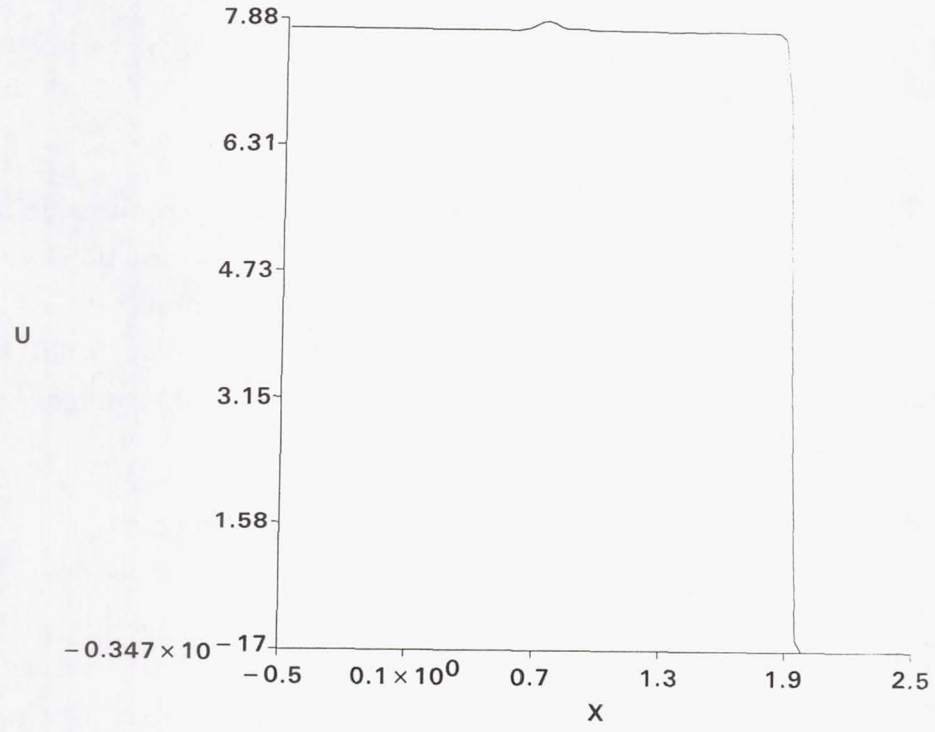


Figure 20a. Velocity profile with conservative variable interpolation

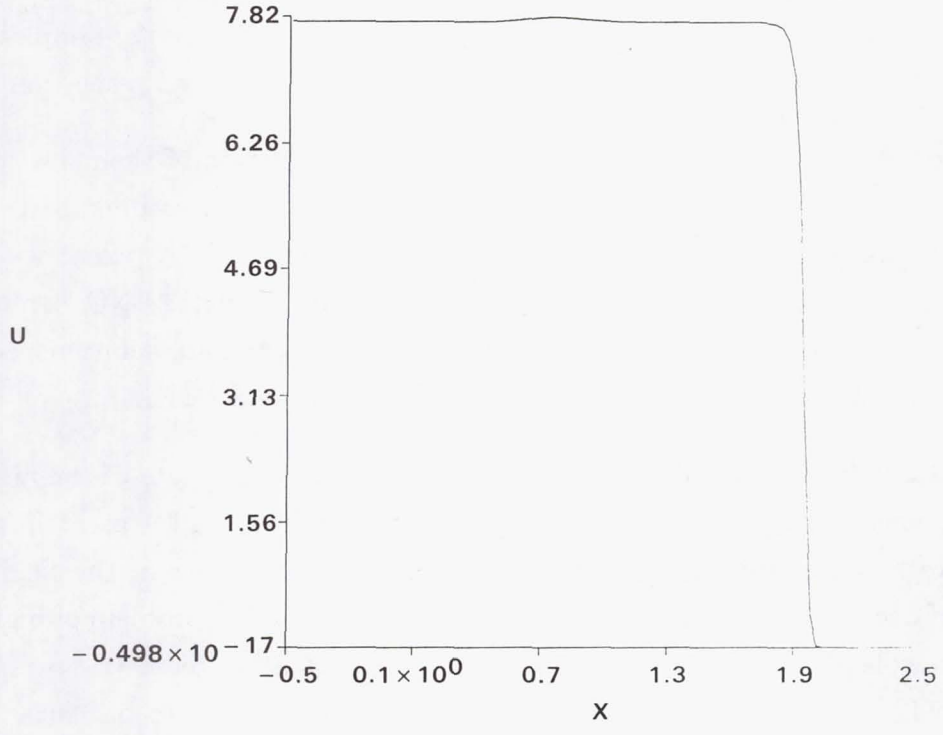


Figure 20a. Velocity profile with characteristic variable interpolation

Section 13.0
CONCLUDING REMARKS

The general framework of ENO formulations for systems of hyperbolic equations have been presented along with a specific implementation based on reconstruction by primitive function, Runge-Kutta time integration and a tensor-product approach to multidimensional problems. Some FORTRAN procedures have been provided to facilitate implementation of ENO schemes by other interested researchers. Several examples have been included to illustrate interesting properties of the new algorithms.

The general framework is very rich with potential. ENO schemes can be constructed based on the integral form or the differential form of hyperbolic equations. We have presented the former in this report. Reconstruction can be based on RD, RP or RM. We have presented RP here. Multidimensional interpolation possibilities are endless. We have presented here a specific method based on a tensor-product approach that is very economical and can be used for all problems that need only Cartesian meshes. We have presented here the semi-discrete formulation and have used a method-of-lines approach by using the fourth-order accurate Runge-Kutta method for advancing the discretized equation in time. Other fully discrete approaches are possible. While the basic theme presented in this report is "smart interpolation," we have only covered the simplest types of implementations. We have only alluded to advances such as "subcell resolution."

If we try to set ENO schemes in perspective, the present state of their development is similar to where TVD schemes were in 1982. The basic TVD algorithms existed then. CFD codes that exploited them came soon thereafter. We are similarly poised with ENO schemes now. Time will tell what the real impact of very highly accurate ENO formulations will be: whether it will be more efficient computations, more accurate solutions, or being able to compute fluid physics problems that were hitherto beyond reach.

While several individuals contribute to the advancement of the state-of-the-art in any field, this researcher would like to identify here a few who have had a great impact on the material presented in this report. Bram van Leer was a pioneer in the field of upwind schemes of accuracy greater than first order. He looked at the problem of high-order schemes and the role of interpolation from a very elegant intuitive geometric perspective. Ami Harten developed the mathematical framework of TVD schemes to put such ideas in an analytic framework. Phil Roe and Stan Osher contributed much by providing simpler approximate solutions to the Riemann problem. Most of the work of this researcher in

recent years has been influenced directly or indirectly by Harten and Osher. Harten extended the TVD ideas to develop the ENO framework and together with Osher and several colleagues, including this researcher, has played a key role in developing what promises to be the algorithmic framework of the next five years for a wide range of CFD problems.

Section 14.0
REFERENCES

[1] S. R. Chakravarthy and S. Osher, "Computing With High-Resolution Upwind Schemes for Hyperbolic Equations," *Lectures in Applied Mathematics*, Volume 22, 1985, pp. 57-86.

[2] Sukumar R. Chakravarthy, "Development of Upwind Schemes for the Euler Equations," NASA Contractor Report 4043, January 1987.

[3] S. R. Chakravarthy and K.-Y. Szema, "Euler Solver for Three-Dimensional Supersonic Flows with Subsonic Pockets," *Journal of Aircraft*, Volume 24, Number 2, February 1987, pp. 73-83.

[4] K.-Y. Szema, S.R. Chakravarthy, W.T. Riba, J. Byerly, and H.S. Dresser, "Multi-Zone Euler Marching Technique for Flow Over Single and Multi-body Configurations," AIAA Paper No. 87-0592, January 1987.

[5] Kuo-Yen Szema and Sukumar Chakravarthy, "A User Guide for the EMTAC-MZ CFD Code", NASA Contractor Report 4283, 1990.

[6] A. Harten, "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 49, 1983, pp. 357-393.

[7] S. R. Chakravarthy and S. Osher, "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws," AIAA Paper 85-0363, January 1985.

[8] S. R. Chakravarthy, "The Versatility and Reliability of Euler Solvers Based on High-Accuracy TVD Formulations," AIAA Paper No. 86-0243, January 1986.

[9] S.R Chakravarthy, K.-Y. Szema, and J.W. Haney, "Unified Nose-to-Tail Computational Method for Hypersonic Vehicle Applications," AIAA Paper No. 88-2564, June 1988.

[10] A. Harten, "On High-Order Accurate Interpolation for Non-Oscillatory Shock-Capturing Schemes," MRC Technical Summary Report No. 2829, June 1985, University of Wisconsin, Mathematics Research Center.

[11] A. Harten and S. Osher, "Uniformly High-Order Accurate Non-Oscillatory Schemes, I," MRC Technical Summary Report No. 2823, May 1985.

[12] A. Harten, S. Osher, B. Engquist, and S. R. Chakravarthy, "Some Results on Uniformly High-Order Accurate Essentially Non-Oscillatory Schemes," *Journal of Applied Numerical Mathematics*, Vol. 2, 1986, pp. 347-377.

[13] S. R. Chakravarthy, A. Harten, and S. Osher, "Essentially Non-Oscillatory Shock-Capturing Schemes of Arbitrarily-High Accuracy," AIAA Paper 86-0339, January 1986.

[14] A. Harten, "ENO Schemes with Subcell Resolution," ICASE Report 87-56, NASA Langley Research Center, August 1987.

[15] G. A. Sod, "A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 27, 1978, pp. 1-31.

[16] T .A. Zang, M.Y. Hussaini, and D.M. Bushnell, "Numerical Computations of Turbulence Amplification in Shock-Wave Interactions," *AIAA Journal*, Vol. 22, No. 1, January 1984, pp. 13-21.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-4285		

4. Title and Subtitle	5. Report Date
Some Aspects of Essentially Nonoscillatory (ENO) Formulations for the Euler Equations	May 1990
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Sukumar R. Chakravarthy	
	10. Work Unit No. 505-60-01-02

9. Performing Organization Name and Address	11. Contract or Grant No.
Rockwell International Science Center P. O. Box 1085 Thousand Oaks, CA 91360	NAS1-17492
	13. Type of Report and Period Covered: Contractor Report

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
NASA Langley Research Center Hampton, VA 23665-5225	

15. Supplementary Notes

Langley Technical Monitor: David H. Rudy
Final Report (Part 3)

16. Abstract

This report describes an essentially nonoscillatory (ENO) formulation for hyperbolic systems of conservation laws. ENO approaches are based on "smart interpolation" to avoid spurious numerical oscillations. ENO schemes are a superset of Total Variation Diminishing (TVD) schemes. In the recent past, TVD formulations were used to construct shock-capturing finite-difference methods. At extremum points of the solution, TVD schemes automatically reduce to being first-order accurate discretizations locally while away from extrema, they can be constructed to be of higher-order accuracy. The new framework helps construct essentially non-oscillatory finite-difference methods without recourse to local reductions of accuracy to first order. Thus arbitrarily high orders of accuracy can be obtained. The basic general ideas of the new approach can be specialized in several ways and this report describes one specific implementation based on a) the integral form of the conservation laws, b) reconstruction based on the primitive functions, c) extension to multiple dimensions in a tensor product fashion, d) Runge-Kutta time integration. The resulting method is fourth-order accurate in time and space, and is applicable to uniform Cartesian grids. The construction of such schemes for scalar equations and systems in one and two space dimensions is described along with several examples which illustrate interesting aspects of the new approach.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Upwind Schemes Total-Variation-Diminishing Schemes Euler equations	Unclassified - Unlimited Subject Category 34

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	72	A04

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, Virginia 22161-2171

NASA-Langley, 1990