AIAA JOURNAL
Vol. 30, No. 12, December 1992

# Finite-Volume Implementation of High-Order Essentially Nonoscillatory Schemes in Two Dimensions

Jay Casper*
*Vigyan, Inc., Hampton, Virginia 23666*

**We continue the study of the finite-volume application of high-order-accurate, essentially nonoscillatory shock-capturing schemes to two-dimensional initial-boundary-value problems. These schemes achieve high-order spatial accuracy, in smooth regions, by a piecewise polynomial approximation of the solution from cell averages. In addition, this spatial operation involves an adaptive stencil algorithm, in order to avoid the oscillatory behavior that is associated with interpolation across steep gradients. High-order Runge-Kutta methods are employed for time integration, thus making these schemes best suited for unsteady problems. Schemes are developed which use fifth- and sixth-order-accurate spatial operators in conjunction with a fourth-order time operator. Under a sufficient time-step restriction, numerical results suggest that these schemes converge according to the higher-order spatial accuracy, for unsteady problems. Second-, third-, and fifth-order algorithms are applied to the Euler equations of gasdynamics and tested on a problem that models a shock-vortex interaction.**

## Introduction

In recent years, a class of numerical shock-capturing schemes has been developed for the purpose of generalizing the first-order method of Godunov[1] to an arbitrary order of accuracy. High-order spatial accuracy is obtained, wherever the solution is smooth, by an essentially nonoscillatory (ENO), piecewise-polynomial approximation 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 spatial operator enables a flux computation that is high order accurate in smooth regions and avoids oscillations near discontinuities.

Many of the ideas for higher-order Godunov-type schemes can be traced to the work of Van Leer,[2] followed by Harten,[3] and Harten and Osher,[4] and culminating in the unified presentation of ENO schemes by Harten et al.[5] High-order ENO schemes were originally developed[5] for one-dimensional hyperbolic equations within a control-volume formulation. The present author's previous work[6] in this area has resulted in a "natural" extension of these schemes to two dimensions, as they apply to rectangular and smoothly varying grids. The extension is natural in the sense that, when restricted to one dimension, the spatial operator involved in these two-dimensional schemes reduces to that of the original 1D schemes. This two-dimensional extension is briefly reviewed in the following two sections.

Multidimensional ENO schemes have also been proposed by Shu and Osher.[7] These schemes differ fundamentally with the original presentation in Ref. 5, in that the control-volume formulation is discarded, in favor of one that involves only point values. Conservation is achieved by a cleverly derived flux function. The principal motivation for this approach is the greater computational efficiency of finite difference algorithms, when accuracy above second order is desired. However, the adequate handling of boundary conditions is more readily apparent within a finite-volume formulation. Furthermore, the application of these "point-ENO" schemes to discontinuous or unstructured meshes appears, at present, impossible, whereas high-order finite-volume algorithms for such geometries have already been developed (e.g., Barth and Frederickson[8]).

The finite-volume ENO schemes presented in our previous work[6] were developed in semidiscrete form, employing high-order Runge-Kutta methods for temporal accuracy. The particular algorithms tested in that research were third and fourth order accurate in space and time. In this paper, because of the anticipated high cost of using a fifth- or higher-order time integration, we examine the use of fifth- and sixth-order-accurate spatial operators with a fourth-order Runge-Kutta time-marching method. Results of numerical experiments are presented.

## High-Order Two-Dimensional ENO Schemes

We review here the design of finite-volume, high-order-accurate, ENO schemes in two dimensions. Since this review is necessarily brief, the interested reader is referred to Ref. 6 for details. We seek to numerically approximate weak solutions of a hyperbolic system of equations:

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

subject to given initial data and boundary conditions.

The function $u = (u^1, u^2, \ldots, u^m)^T$ is a state vector, and the flux functions $f(u)$ and $g(u)$ are vector-valued, differentiable functions of $m$ components. It is well known that a weak solution of our hyperbolic system obeys the integral form of Eq. (1), where the limits of integration can reflect any smoothly bounded domain in the $x$-$y$ plane and any time interval $(t^1, t^2)$. Seeking such a solution, let the set $\{\mathcal{C}_{ij}\}$, where

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

denote a rectangular partition of the $x$-$y$ plane, with $(x_i, y_j)$ denoting the centroid of each rectangle. With a semidiscrete formulation in mind, we note that a weak solution of Eq. (1) must satisfy

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

Presented as Paper 91-0631 at the AIAA 29th Aerospace Sciences Meeting, Reno, NV, Jan. 7-10, 1991; received March 19, 1991; revision received April 10, 1992; accepted for publication April 11, 1992. 

*Research Engineer. Member AIAA.

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

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

is the cell average of $u$ in $\mathcal{C}_{ij}$ at time $t$. The fluxes $\hat{f}$ and $\hat{g}$ are given by

$$\hat{f}_{i+\frac{1}{2},j}(t) = \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f[u(x_{i+\frac{1}{2}},y,t)]\, dy$$
$$\hat{g}_{i,j+\frac{1}{2}}(t) = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g[u(x,y_{j+\frac{1}{2}},t)]\, dx \tag{2c}$$

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

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

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

Given $\{\bar{\bar{u}}^n\}$, cell averages of a piecewise smooth solution $u(x,y,t^n)$ of Eq. (1), we desire a numerical solution operator $E_h$ that will be $r$th-order accurate in the sense of local truncation error; i.e., wherever $u$ is sufficiently smooth,

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

where $h$ is the maximum local mesh spacing on a given partition $\{\mathcal{C}_{ij}\}$, and $\Delta t = \mathcal{O}(h)$. We expect the cumulative error due to $N$ applications of $E_h$, to the initial data $\{\bar{\bar{u}}^0\}$, to be $\mathcal{O}(h^r)$:

$$E_h^N \bar{\bar{u}}^0 - \bar{\bar{u}}^N = \mathcal{O}(h^r) \tag{5}$$

Furthermore, we desire that our numerical scheme avoid the development of spurious $\mathcal{O}(1)$ oscillations near discontinuities in $u$. To achieve this property, as well as those of Eqs. (4) and (5), we require our operator to be *essentially nonoscillatory*. Such an operator satisfies, in the one-dimensional scalar case,

$$TV(E_h \bar{u}) \le TV(u) + \mathcal{O}(h^{1+q}), \qquad q > 0 \tag{6}$$

where $TV$ represents total variation in $x$. For details concerning the initial development of ENO schemes, see Ref. 5 and the references therein.

Employing formulation (3), we discretize the temporal operation by using Runge-Kutta methods developed by Shu and Osher.[7] These methods are of high-order accuracy and TVD, in the sense that the temporal operator itself does not increase the total variation of the solution. For our purposes, such a $p$-stage Runge-Kutta method can be written:

$$\bar{\bar{u}}_{ij}^{(l)} = \sum_{m=0}^{l-1} [\alpha_{lm} \bar{\bar{u}}_{ij}^{(m)} + \beta_{lm} \Delta t (\mathcal{L}u^{(m)})_{ij}],$$
$$l = 1, 2, \ldots, p \tag{7}$$
$$\bar{\bar{u}}_{ij}^{(0)} = \bar{\bar{u}}_{ij}^{n}, \qquad \bar{\bar{u}}_{ij}^{(p)} = \bar{\bar{u}}_{ij}^{n+1}$$

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 $\alpha$ and $\beta$ (see Ref. 7 for details).

For spatial accuracy, it is shown in Ref. 6 that a discrete spatial operator $L$ can be designed that acts upon the $\{\bar{\bar{u}}(t)\}$ and approximates the pointwise operation of $\mathcal{L}$ to high order, i.e.,

$$L\bar{\bar{u}}(t) = \mathcal{L}u(t) + \mathcal{O}(h^r) \tag{8a}$$

wherever $u$ is smooth. Explicitly, $L$ can be written

$$[L\bar{\bar{u}}(t)]_{ij} = -\frac{1}{a_{ij}} \left[ \bar{f}_{i+\frac{1}{2},j}(t) - \bar{f}_{i-\frac{1}{2},j}(t) + \bar{g}_{i,j+\frac{1}{2}}(t) - \bar{g}_{i,j-\frac{1}{2}}(t) \right] \tag{8b}$$

where $\bar{f}$ and $\bar{g}$ are correspondingly high-order numerical approximations to the fluxes in Eq. (2c) and are determined as follows.

Noting that the fluxes in Eq. (2c) require pointwise information of the solution at time $t$, our first step is the high-order approximation of $u(x,y,t)$ from the cell averages $\{\bar{\bar{u}}(t)\}$. To this end, let $\mathcal{P}_{ij}(x,y)$ denote a polynomial of degree $r-1$, which approximates the solution at time $t$, within each cell $\mathcal{C}_{ij}$. The integration in Eq. (2c) is then approximated by quadrature in the following manner. The polynomials $\{\mathcal{P}_{ij}\}$ are evaluated at appropriately chosen quadrature points along each cell interface. The differing representations of $\mathcal{P}_{ij}(x,y)$, from cell to cell, will create discontinuities at the cell boundaries, giving rise to a set of local Riemann problems. Let $f^{Rm}[u_1,u_2]$ denote the flux associated with the solution to the Riemann problem whose initial states are $u_1$ and $u_2$. The discrete flux associated with each of these initial-value problems then contributes to the high-order quadrature. The numerical flux computation can then be summarized as follows.

Polynomial approximation in each cell:

$$\mathcal{P}_{ij}(x,y) \approx u(x,y,t) \qquad \text{for } x,y \in \mathcal{C}_{ij} \tag{9a}$$

Riemann fluxes at quadrature points:

$$\tilde{f}_{i+\frac{1}{2}}^k \equiv f^{Rm}[\mathcal{P}_{ij}(x_{i+\frac{1}{2}},y_k), \quad \mathcal{P}_{i+1,j}(x_{i+\frac{1}{2}},y_k)]$$
$$\tilde{g}_{j+\frac{1}{2}}^k \equiv g^{Rm}[\mathcal{P}_{ij}(x_k,y_{j+\frac{1}{2}}), \quad \mathcal{P}_{i,j+1}(x_k,y_{j+\frac{1}{2}})] \tag{9b}$$

where $x_k \in (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$ and $y_k \in (y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}})$ for

$$k = 1,2, \ldots, K$$

Integration by quadrature:

$$\bar{f}_{i+\frac{1}{2},j}(t) = \frac{\Delta_j y}{2} \sum_{k=1}^{K} c_k \tilde{f}_{i+\frac{1}{2}}^k$$
$$\bar{g}_{i,j+\frac{1}{2}}(t) = \frac{\Delta_i x}{2} \sum_{k=1}^{K} c_k \tilde{g}_{j+\frac{1}{2}}^k \tag{9c}$$

The $K$-point quadratures [Eq. (9c)] are the required numerical fluxes $\bar{f}$, $\bar{g}$ in Eq. (8b). Our two-dimensional high-order ENO scheme is then completed by the substitution of Eq. (8b) for $\mathcal{L}$ in the $p$-stage Runge-Kutta method [Eq. (7)]. It is shown in Ref. 6 that, if the error made at the approximation stage (9a) is $\mathcal{O}(h^r)$ and globally Lipschitz continuous, then the numerical solution operator $E_h$, defined by Eqs. (7–9), is formally $r$th-order accurate in the sense of local truncation error, as given by Eq. (4). Furthermore, the desired ENO property (6) of $E_h$ is dependent on this polynomial approximation. Clearly, then, this pointwise "reconstruction" of the solution from its cell averages is the defining step for this class of schemes. The particular reconstruction operator we utilize in this work is a two-dimensional extension of the "reconstruction by primitive" in Ref. 5 and, for completeness, is briefly reviewed in the following section.

A few comments should be made concerning implementation and stability. For this purpose, the spatial and temporal orders of accuracy, for which $E_h$ is designed, are denoted $r_s$ and $r_t$, respectively. For $r_s = 1$ or 2, the flux quadratures (9c) are computed by the midpoint rule, making these two cases no different from conventional first- or second-order finite-vol-

ume schemes. For $r_s \geq 3$, we use a $K$-point Gaussian quadrature, where $K = 2$ for $r_s = 3$ or 4, $K = 3$ for $r_s = 5$ or 6, etc. Regarding stability, the size of a time step is restricted by the limits established by Shu and Osher[7] for the high-order Runge-Kutta methods we will use. For the discrete scalar case our stability criterion can be expressed as

$$\Delta t = (\text{CFL}) \min_{ij} \left[ \frac{a_{ij}}{|f'(u_{ij})|\Delta_j y + |g'(u_{ij})|\Delta_i x} \right] \tag{10}$$

and is easily extended to a hyperbolic system. As reported in Ref. 7, the stability limits on the values of the previously given CFL coefficient for $r_t = 2$, 3, and 4, are 1.0, 1.0, and 0.87, respectively.

## Two-Dimensional Reconstruction

We review here the method by which we obtain high-order-accurate pointwise information of the solution $u(x,y,t)$ from its cell averages $\{\bar{\bar{u}}(t)\}$. For this purpose let $R^2$ denote a two-dimensional spatial operator that reconstructs the cell averages of the solution at time $t$, and yields a global, piecewise polynomial $R^2[x,y;\bar{\bar{u}}(t)]$ of degree $r-1$. Relating this approximation to the local polynomials $\{\mathcal{P}_{ij}\}$ of the previous section and assuming sufficient smoothness in $u(x,y,t)$, we write

$$R^2[x,y;\bar{\bar{u}}(t)] = \bigcup_{ij} \mathcal{P}_{ij}(x,y) = u(x,y,t) + \mathcal{O}(h^r)$$

It might seem plausible to extend any one-dimensional spatial operator to two dimensions by simply *summing* two one-dimensional operators, which is most commonly done in conventional finite-volume algorithms. However, this "directional-splitting" approach will inherently limit a finite-volume scheme to second order accuracy, when applied to the solution of a nonlinear equation. Nonetheless, we can describe the implementation of $R^2$ as a product of two one-dimensional, "reconstruction-by-primitive" operators in Ref. 5. The purpose served by interpolation of a primitive function is that of "conservation," in the sense that the cell-averaging operator defined by the right-hand side of Eq. (2b) is the left-hand inverse of $R^2$, i.e.,

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

For the purpose of clarity, this procedure is discussed within the framework of a scalar function defined on a rectangular computational mesh. It should be noted, however, that this operator has been successfully extended to hyperbolic systems of conservation laws and to general curvilinear coordinates. Because this review is necessarily brief, the interested reader is referred to Ref. 6 for a more detailed description.

Assume we are given a discrete set of cell averages $\{\bar{\bar{w}}_{ij}\}$ of a piecewise smooth function $w(x,y)$, which can be written in the following from:

$$\bar{\bar{w}}_{ij} = \frac{1}{\Delta_i x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{1}{\Delta_j y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} w(x,y)\, dy\, dx \equiv \mathcal{Q}_x \mathcal{Q}_y w(x,y) \tag{11a}$$

The implementation of $R^2$ is based on the observation that the cell-average quantity in Eq. (11a) is produced by the product of two one-dimensional averaging operators acting on the function $w(x,y)$. Therefore, it makes sense to extract pointwise information from cell averages by the use of a product of differential operators. The first operator reduces the cell averages to line averages, which are, in turn, reduced to point values by the second operator.

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

$$\bar{W}_j(x) = \int_{x_0}^{x} \frac{1}{\Delta_j y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} w(\xi,y)\, dy\, d\xi \tag{11b}$$

which clearly implies

$$\frac{d}{dx} \bar{W}_j(x) = \frac{1}{\Delta_j y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} w(x,y)\, dy = \mathcal{Q}_y w(x,y) \tag{11c}$$

Thus, the derivative of the primitive $\bar{W}_j(x)$ is a function in $x$, whose values are line averages in $y$, thereby representing the first of the two integral reductions we seek. Though, in general, we do not know the specific form of $\bar{W}_j(x)$, we can approximate it by polynomial interpolation, if we know its values at a sufficient number of points in $x$. However, the only values we know, initially, are the cell averages (11a). We therefore seek a relationship between pointwise values of $\bar{W}_j(x)$ and the cell averages $\{\bar{\bar{w}}_{ij}\}$, and we see immediately from the definitions (11a) and (11b) that

$$\Delta_{ix} \bar{\bar{w}}_{ij} = W_j(x_{i+\frac{1}{2}}) - W_j(x_{i-\frac{1}{2}})$$

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

$$\bar{W}_j(x_{i+\frac{1}{2}}) = \sum_{k=i_0}^{i} \Delta_{kx} \bar{\bar{w}}_{kj} \tag{11d}$$

Let $Q_{ij}^r(x;\bar{W}_j)$ denote an $r$th-degree polynomial in $x$, within cell $\mathcal{C}_{ij}$, which interpolates $r+1$ of the values in (11d). We require that two of these interpolated values be located at the interfaces $\{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\}$ of $\mathcal{C}_{ij}$. We are therefore free to choose the other $r-1$ points, and we do so subject to a condition of smoothness, as measured by discrete derivatives of $\bar{W}_j$, on a contiguous stencil in $x$. We annex a point to the left or right of a lower-order interpolation stencil, according to the magnitude of the next higher-order difference, which can be calculated due the inclusion of that point. The "smoother direction" is then determined by the smaller difference. Henceforth, we will refer to this hierarchical algorithm as an ENO interpolation (see Ref. 5 for details).

The relation (11c), then, suggests that we approximate $\mathcal{Q}_y w(x,y)$ by defining

$$\bar{v}_{ij}(x) \equiv \frac{d}{dx} Q_{ij}^r(x;\bar{W}_j), \qquad x_{i-\frac{1}{2}} \leq x \leq x_{i+\frac{1}{2}} \tag{11e}$$

Then $\bar{v}_{ij}(x)$ is a polynomial in $x$, of degree $r-1$, which approximates $\mathcal{Q}_y w(x,y)$ within the cell $\mathcal{C}_{ij}$. We can then define $R_x$, the first of two reconstruction operators, by

$$R_x(x;\bar{\bar{w}}) \equiv \frac{d}{dx} Q_{ij}^r(x;\bar{W}_j), \quad \text{all} \quad i,j = \bigcup_{ij} \bar{v}_{ij}(x) \tag{11f}$$

Thus, the operator $R_x$ yields a set of polynomials $\{\bar{v}_{ij}(x)\}$, each of which approximates $\mathcal{Q}_y w(x,y)$ to $\mathcal{O}(\Delta x^r)$ in smooth regions. We may then consider $\bar{v}_{ij}(x)$ as the line average of piecewise smooth function $v(x,y)$, i.e.,

$$\bar{v}_{ij}(x) = \mathcal{Q}_y v(x,y) \approx \mathcal{Q}_y w(x,y)$$

We now reduce these approximate line averages to point values, with a differential operation in $y$. Analogous to the method (11), we define another primitive function $W(x,y)$, which for $x$ fixed, is associated with $v(x,y)$ by

$$W(x,y) = \int_{y_0}^{y} v(x,y)\, dy \tag{12a}$$

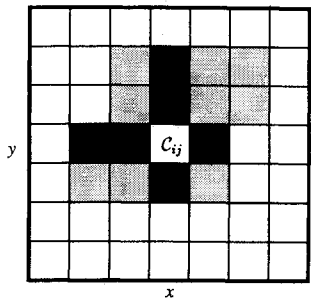


**Fig. 1 Possible reconstruction stencils for the cell $\mathcal{C}_{ij}$, as determined by $R^2$ and $R_+$ with $n_s = 4$.**

and whose pointwise values at cell interfaces are given by

$$W(x, y_{j+\frac{1}{2}}) = \sum_{k=j_0}^{j} \Delta_k y \; \bar{v}_{ik}(x) \tag{12b}$$

Now, for a fixed $x$ within a cell $\mathcal{C}_{ij}$, we fit $r+1$ of the point values in Eq. (12b) with an $r$th-degree polynomial $Q_{ij}^r(x;W)$, by ENO interpolation. We can then obtain a pointwise approximation, in $y$, to $v(x,y)$ by defining

$$\mathcal{P}_{ij}(x,y) \equiv \frac{\mathrm{d}}{\mathrm{d}y} Q_{ij}^r(y;W) \qquad \text{for} \quad x,y \in \mathcal{C}_{ij} \tag{12c}$$

and a second operator, $R_y$:

$$R_y[(y;\bar{v}(x)] \equiv \frac{\mathrm{d}}{\mathrm{d}x} Q_{ij}^r(y;W) \qquad \text{all} \quad i,j \tag{12d}$$

$$= \bigcup_{ij} \mathcal{P}_{ij}(x,y)$$

The values obtained in Eq. (12d) are the pointwise approximations to $w(x,y)$ that are desired from its cell averages. Clearly, from Eqs. (11f) and (12d), this operator is implemented as a product, whose form and truncation error, for smooth $w(x, y)$, can be written as

$$R^2(x,y;\bar{\bar{w}}) \equiv R_y[y;R_x(x;\bar{\bar{w}})] = w(x,y) + \mathcal{O}(h^r) \tag{13a}$$

Furthermore, the ENO interpolation enables this procedure to avoid $\mathcal{O}(1)$ oscillations near discontinuities. We also note that this procedure does not require uniformity of the mesh. Thus, on a rectangular gird, the reconstruction (13a) retains its formal properties, even though $\Delta_i x$ and $\Delta_j y$ may be random functions of $i$ and $j$.

We note here that the "directional-splitting" approach to the two-dimensional reconstruction problem would yield a summation operator

$$R_+(x,y;\bar{\bar{w}}) \equiv R_x(x;\bar{\bar{w}}) + R_y(y;\bar{\bar{w}}) = w(x,y) + \mathcal{O}(h^2) \tag{13b}$$

whose implementation in Eqs. (7-9) yields a second-order scheme, regardless of the one-dimensional truncation errors of $R_x$ or $R_y$.

Let $n_s$ denote the number of cells in the one-dimensional interpolation set for $R_x$ or $R_y$. Then there are $n_s^2$ cells required for the product (13a) and $2n_s - 1$ for the second-order summation (13b). Examples of both types of reconstruction stencils, for $n_s = 4$, are pictured in Fig. 1. The entire shaded area represents a fourth-order stencil that might result from the product (13a), and the darker shading represents a stencil determined by the summation (13b). Clearly, there is more computational cost involved in the truly two-dimensional implementation of the scheme (7-9). However, we expect better accuracy, on a given mesh, than with a directionally split version. We defer this issue to the end of the paper.

## Spatial vs Temporal Accuracy

In the second-, third-, and fourth-order algorithms tested in Ref. 6, we have employed the scheme (7-9) with $r_s = r_t$. However, in extending these schemes beyond fourth-order accuracy, we find it currently impractical to retain this strict correspondence between the spatial and temporal operators. For example, the fifth-order Runge-Kutta method suggested in Ref. 7 requires six stages, whereas the stability limit on the CFL number is only 7/30. One is then motivated to investigate the possible use of higher-order spatial operators with a lower-order Runge-Kutta method. Mathematically, the formal order of accuracy of such a scheme will be determined by the lower-order temporal truncation error. However, we do consider, the possibility of "suppressing" this lower-order error by reducing the time step. We have explored this possibility by performing grid refinement tests, using $r_s$, $r_t$, and the CFL coefficient as parameters. The solutions of two-dimensional scalar hyperbolic equations are the subject of this grid refinement study.

Our first test involves the solution of the two-dimensional linear advection equation:

$$u_t + u_x + u_y = 0, \qquad t > 0 \tag{14a}$$

with initial data

$$u(x,y,0) = \tfrac{1}{2} \cos \pi(x+y) + \tfrac{1}{2} \tag{14b}$$

The solution of this initial-value problem (IVP) is two-periodic in $x$ and $y$ for all time $t$. By restricting our computational domain to $\{-1 \le x,y \le 1\}$, we thereby make the boundaries two-periodic also, effectively removing them from consideration. We note here that, even though we are solving a linear equation, the high-order scheme (7-9) applied to Eq. (14a) is still nonlinear, due to the adaptive-stencil algorithm of the reconstruction operator.

Since the solution is smooth for all time, we apply our scheme for one period in time, i.e., to $t = 2.0$. 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, using the scheme (7-9) and the values $r_s = 5$ and 6, $r_t = 3$ and 4, and CFL = 0.75, 0.5, and 0.25. The algorithms achieved by various combinations of $r_s$, and $r_t$ will be denoted by ENO-2D-$r_s$-$r_t$ in our results.

For $r_t = 3$, the numerical results have suggested that the temporal truncation error is dominant for the tested values of $r_s$, CFL, and mesh size. However, in the case $\{r_s = 5, r_t = 4, \text{CFL} = 0.5\}$, our study suggests the desired fifth-order convergence. (The results were not significantly improved by lowering the CFL to 0.25.) These solution errors, along with their corresponding computational orders of accuracy "$r_c$," are

**Table 1 Solution error for IVP (14)**

| | ENO-2D-5-4 CFL = 0.5 | | | |
|---|---|---|---|---|
| Grid | $\|e\|_\infty$ | $r_c$ | $\|e\|_1$ | $r_c$ |
| $8 \times 12$ | 2.332 $E$-2 | | 1.463 $E$-2 | |
| $61 \times 24$ | 7.967 $E$-4 | 4.87 | 4.865 $E$-4 | 4.91 |
| $32 \times 48$ | 2.501 $E$-5 | 4.99 | 1.557 $E$-5 | 4.97 |
| $64 \times 96$ | 7.781 $E$-7 | 5.01 | 4.933 $E$-7 | 4.98 |
| $128 \times 192$ | 2.520 $E$-8 | 4.95 | 1.602 $E$-8 | 4.94 |

**Table 2 Solution error for IVP (14)**

| | ENO-2D-6-4 CFL = 0.25 | | | |
|---|---|---|---|---|
| Grid | $\|e\|_\infty$ | $r_c$ | $\|e\|_1$ | $r_c$ |
| $8 \times 12$ | 1.028 $E$-2 | | 5.728 $E$-3 | |
| $16 \times 24$ | 3.741 $E$-4 | 4.78 | 1.656 $E$-4 | 5.11 |
| $32 \times 48$ | 1.093 $E$-5 | 5.10 | 2.995 $E$-6 | 5.79 |
| $64 \times 96$ | 3.164 $E$-7 | 5.11 | 5.568 $E$-8 | 5.75 |
| $128 \times 192$ | 9.112 $E$-9 | 5.12 | 9.775 $E$-10 | 5.83 |

**Table 3 Solution error for IVP (15)**

| Grid | ENO-2D-6-4 CFL = 0.5 $\|e\|_\infty$ | $r_c$ | $\|e\|_1$ | $r_c$ |
|---|---|---|---|---|
| $8 \times 12$ | 2.091 $E$-2 | | 5.360 $E$-3 | |
| $16 \times 24$ | 2.998 $E$-3 | 2.80 | 5.060 $E$-4 | 3.41 |
| $32 \times 48$ | 1.949 $E$-4 | 3.94 | 2.633 $E$-5 | 4.26 |
| $64 \times 96$ | 1.001 $E$-5 | 4.28 | 1.136 $E$-6 | 4.53 |
| $128 \times 192$ | 3.890 $E$-7 | 4.69 | 4.558 $E$-8 | 4.64 |
| $256 \times 384$ | 1.773 $E$-8 | 4.46 | 1.851 $E$-9 | 4.62 |

**Table 4 Solution error for IVP (15)**

| Grid | ENO-2D-5-4 CFL = 0.25 $\|e\|_\infty$ | $r_c$ | $\|e\|_1$ | $r_c$ |
|---|---|---|---|---|
| $8 \times 12$ | 1.744 $E$-2 | | 4.173 $E$-3 | |
| $16 \times 24$ | 1.918 $E$-3 | 3.19 | 3.328 $E$-4 | 3.65 |
| $32 \times 48$ | 8.417 $E$-5 | 4.51 | 1.320 $E$-5 | 4.66 |
| $64 \times 96$ | 3.242 $E$-6 | 4.70 | 3.225 $E$-7 | 5.36 |
| $128 \times 192$ | 8.073 $E$-8 | 5.33 | 6.455 $E$-9 | 5.64 |
| $256 \times 384$ | 1.734 $E$-9 | 5.54 | 1.228 $E$-10 | 5.72 |

presented in Table 1. This value $r_c$ is calculated by assuming a linear accumulation of error as given by Eq. (5), and the error $\|e\|$ is calculated with respect to the $L_\infty$ and $L_1$ norms. The exact and computational solutions are compared by cell-centered pointwise output, which is achieved by a final reconstruction of the approximate cell averages at $t = 2.0$. Sixth-order convergence is suggested by the results of the case $\{r_s = 6, r_t = 4, \text{CFL} = 0.25\}$; these results are shown in Table 2.

For a nonlinear example, we consider the two-dimensional inviscid Burgers equation

$$u_t + (\tfrac{1}{2}u^2)_x + (\tfrac{1}{2}u^2)_y = 0, \qquad t > 0 \tag{15a}$$

again with initial data

$$u(x,y,0) = \tfrac{1}{2} \cos \pi(x+y) + \tfrac{1}{2} \tag{15b}$$

We solve this initial-value problem on the same domain as the previous example and again apply periodic boundary conditions. In this case, because of the nonlinearity of Eq. (15a), gradients immediately begin to steepen for $t > 0$, until a shock eventually forms at time $t = 1/\pi$. We therefore solve this IVP to $t = 0.15$, when the solution remains smooth. Tables 3 and 4 illustrate the accumulated errors for the the same parameters as in the previous example.

Because of the fewer number of time steps allowed for this nonlinear case, we perform one additional grid refinement. Again, the results suggest fifth- and sixth-order convergence. The solution was then computed to and past the point of shock formation, with no visible oscillatory behavior near the discontinuity. For example, Fig. 2 depicts the pointwise distribution of $u$ along three $y = \text{const}$ lines, at $t = 0.45$, when the shock is present.

## Shock Vortex Interaction

We now turn to an application of these schemes to a problem that models an interaction of a vortex with a shock wave. The two-dimensional shock-vortex interaction problem serves as a model for the generation of sound waves in a high-speed flow. For this application we will include a fifth-order solution, which we compute in a manner based on the observations of the previous section. Such high-order accuracy in an ENO-based scheme could be of great benefit to the computational study of compressible flows that exhibit acoustic phenomena, particularly in the presence of shocks.

Early numerical studies[9,10] of the shock-vortex interaction focused primarily on shock-fitting techniques. Consequently, these applications were limited to weak interactions, in which a shock is only slightly deformed. However, there are cases of practical importance, in which a vortex will cause a shock to deform to the point of bifurcation. In such cases, a shock-fitting method is difficult, if not impossible, to apply. More recently, in Ref. 11, this previous inability to predict strong interactions was overcome by the use of a shock-capturing method. We attempt to improve upon these second-order results with our high-order ENO schemes.

The initialization of this problem occurs on the domain $\{0 \le x \le 1\} \times 0 \le y \le 1\}$, with a stationary normal shock at $x = 0.5$ and a Mach 1.1 flow at the inlet, $x = 0$. A vortex is superimposed upstream of the shock and allowed to propagate downstream. Neglecting viscous effects near the walls, we take the two-dimensional Euler equations of gasdynamics to be the governing equations of this flow, having normalized all flow variables with respect to the Mach 1.1 inflow. Our fluid is assumed to be ideal air, with a ratio of specific heats $\gamma = 1.4$.

The solid-core vortex we implement is a slight modification of the hybrid model in Ref. 11. We model vortical flow as a mass rotating at a constant tangential velocity within a core of specified radius, and decaying exponentially outside the core:

$$V_\theta(r) = \begin{cases} Vc, & r \le R_c \\ V_c \exp\,[-(r-R_c)^2/\Omega], & r > R_c \end{cases} \tag{16}$$

where $V_\theta$ is the tangential velocity, $V_c$ the core velocity, $r$ the radial distance from the vortex center, $R_c$ the core radius, and $\Omega$ a decay parameter. [Though the vortex (16) is somewhat unphysical, it is used for the purpose of comparison with results in Ref. 11.] This vortex is initially located halfway between the inlet and the shock, with the parameters $V_c = 0.3$, $R_c = 0.05$, and a clockwise orientation. The decay parameter $\Omega$ is determined by requiring that $V_\theta$ initially decay to the value $10^{-10}$ at the channel walls. Such a velocity profile is illustrated in Fig. 3.

We solve the Euler equations on a $150 \times 100$ Cartesian grid, which is uniform in $y$ and refined in $x$ about the shock location, using a Roberts transformation.[12] The scheme (7-9) is applied for the cases $\{r_s = r_t = 2\}$, $\{r_s = r_t = 3\}$, and $\{r_s = 5, r_t = 4\}$, all at a CFL of 0.5. The required Riemann fluxes (9b) are approximated by Roe's method.[13]

Figures 4a–4c show a time history of our third-order calculation, each plot representing 30 equally spaced pressure con-

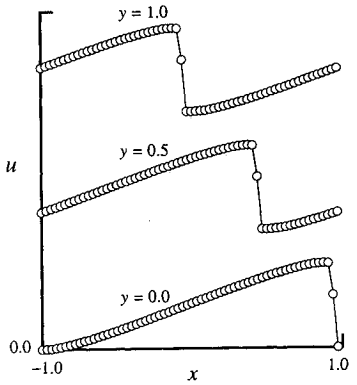


**Fig. 2 The solution of IVP (15) by ENO-2D-5-4, at $t = 0.45$, along three grid lines of a $64 \times 96$ mesh.**

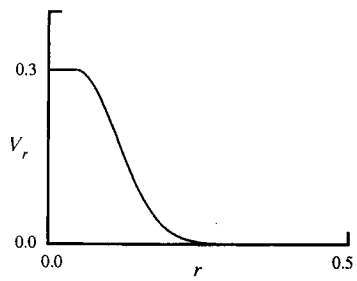


**Fig. 3 Tangential velocity profile of the solid-core vortex defined by Eq. (16).**

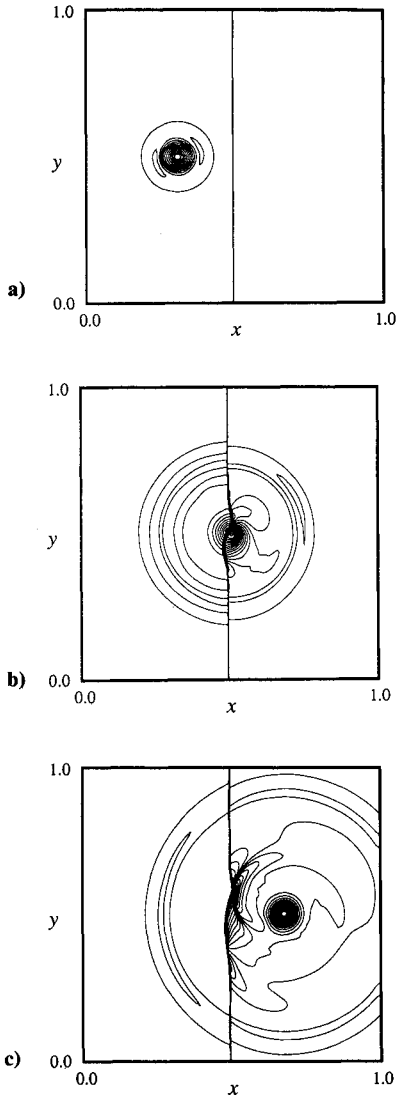


**Fig. 4 a) Pressure, $t = 0.05$, ENO-2D-3-3; b) pressure, $t = 0.20$, ENO-2D-3-3; and c) pressure, $t = 0.35$, ENO-2D-3-3.**

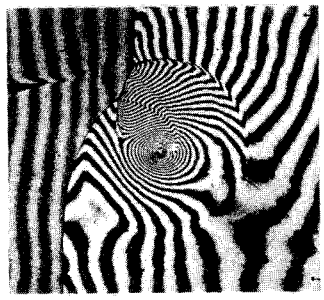

**Fig. 5 Mach-Zender interferogram.[14]**

tours, with the entire pressure range being $0.81 \leq P \leq 1.37$. During the interaction with the vortex core (Fig. 4b), the clockwise rotation of the vortex causes the upper portion of the shock to deflect downstream as compression occurs, whereas the lower portion reacts to an expansion by moving upstream. As the vortex core passes through the shock, we notice that bifurcation occurs, causing the shock to branch into two triple points, connected by a single Mach stem. In Fig. 4c the acoustic pattern is visible, emanating from this bifurcation region. Figure 5 is a Mach-Zehnder interferogram[14] that depicts a shock-vortex interaction resulting in a shock bifurcation similar to that predicted by our computation and is presented for qualitative comparison only.

Figures 6a–6c represent an order-of-accuracy comparison, on a smaller region of the solution domain, at $t = 0.40$, after the vortex has moved through the shock. To more fully view the detailed structure of the acoustic front predicted by these calculations, we plot a narrower range of pressure than is present in the solution. (The shock and vortex core are not entirely visible, as a result.) The range for all three plots is $1.19 \leq P \leq 1.37$, with an increment of 0.002. Viewing the cases $r_s = 2$, 3, and 5, we see that, as the order of accuracy increases, there is clearly a more sharply defined acoustic wave front. Not only is this front more narrowly captured, but also its crest is sharper.

The numerical study of such interactions may lead to an enhanced capability of predicting flows that are of interest to the aeroacoustic community, such as the shock noise in supersonic jets. Studies are currently under way for the purpose of investigating the use of high-order shock-capturing schemes in these types of predictions.

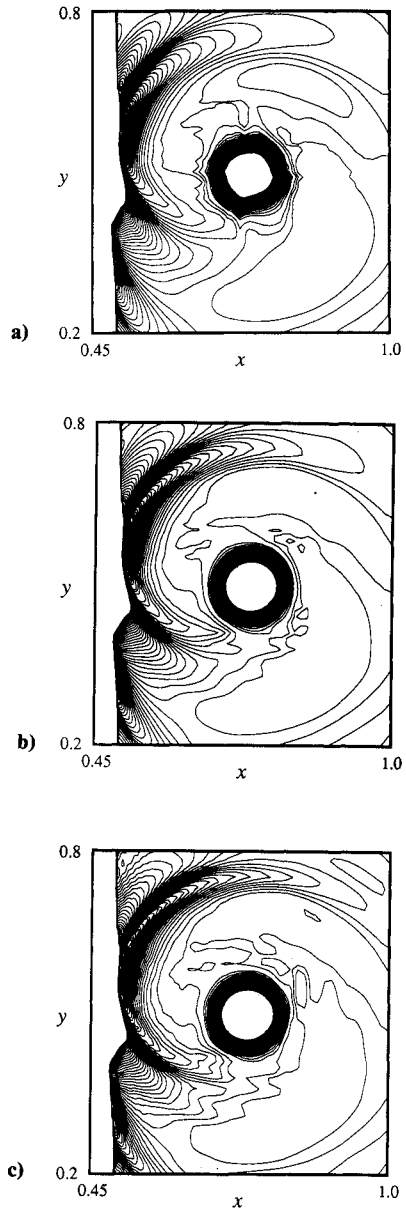


**Fig. 6 a) Pressure, $t = 0.40$, ENO-DS-2-2; b) pressure, $t = 0.40$, ENO-2D-3-3; and c) pressure, $t = 0.40$, ENO-2D-5-4.**

**Table 5 Solution error for IVP (15)**

| Grid | ENO-DS-5-4 CFL = 0.5 $\|e\|_\infty$ | $r_c$ | $\|e\|_1$ | $r_c$ |
|---|---|---|---|---|
| $8 \times 12$ | 2.245 $E$-2 | | 7.589 $E$-3 | |
| $16 \times 24$ | 2.758 $E$-3 | 3.03 | 1.210 $E$-3 | 2.65 |
| $32 \times 48$ | 4.085 $E$-4 | 2.75 | 2.137 $E$-4 | 2.50 |
| $64 \times 96$ | 9.157 $E$-5 | 2.16 | 4.991 $E$-5 | 2.10 |
| $128 \times 192$ | 2.241 $E$-5 | 2.03 | 1.245 $E$-5 | 2.00 |
| $256 \times 384$ | 5.598 $E$-6 | 2.00 | 3.111 $E$-6 | 2.00 |

## Cost Relative to Directional Splitting

We briefly discuss here the additional cost and benefits of the high-order implementation of the scheme (7-9), relative to a conventional directionally split version. Not only is the reconstruction (13a) more costly than (13b), in addition, we require the calculation of more than one Riemann flux along a cell face, whereas only one per cell face is required in the directionally split scheme. However, in the truly two-dimensional case we have $r_s = n_s$, whereas directional splitting yields $r_s \leq 2$ for any value of $n_s$. The two versions therefore invite comparison as to accuracy and cost, with respect to a common value of $n_s$. For a given stencil length $n_s$, we will use "ENO-DS" to denote the directionally split analogy to the high-order ENO-2D scheme, with Eq. (13b) for reconstruction and $K = 1$ in Eqs. (9b) and (9c).

For cost comparison we have measured the CPU times required for a residual calculation on a $120 \times 80$ grid, using both algorithms with $n_s = 3$, 4, 5, and 6. The computer code used in this test is a two-dimensional Euler code, and the CPU times were measured using PERFTRACE on a Cray YMP. For $n_s = 3$ or 4 the current version of the ENO-2D code required 45% more CPU time for a residual calculation than did the ENO-DS code. For $n_s = 5$ there was 96% more time required for the high-order algorithm and 92% more for $n_s = 6$.

The question then becomes whether or not this higher cost is offset by better accuracy. We consider this issue by example. In particular, we look to the results of a grid refinement study, for the case of $n_s = 5$ and $r_t = 4$. Within these parameters, the previously mentioned CPU cost comparison tells us that a high-order two-dimensional Euler solution is nearly twice as expensive as the one computed with the corresponding ENO-DS algorithm, on a given mesh. To get an idea of the relative accuracy we can expect from these schemes, we use the scheme ENO-DS to solve the nonlinear IVP (15), with $n_s = 5$, $r_t = 4$, and CFL = 0.5, in order to compare with the ENO-2D results in Table 3. The corresponding errors using the directionally split scheme are shown in Table 5.

The first thing we note in Table 5 is the expected second-order convergence. In a grid-by-grid error comparison, the error using ENO-2D has outdistanced the ENO-DS error by an order of magnitude, by the time the mesh is refined to $64 \times 96$. The mesh has to be refined to $128 \times 192$ in order to achieve a similar accuracy with ENO-DS, increasing the cost of the directionally split solution by a factor of eight. This added cost by far outweighs the factor of two required by ENO-2D on the coarser mesh. Although this example alone cannot be considered conclusive, it does suggest that, for two-dimensional inviscid calculations, high-order methods can be used economically. Research in the area of cost efficiency is ongoing.

## Concluding Remarks

Previously developed high-order, finite-volume ENO schemes have been further explored. Schemes have been developed which use fifth- and sixth-order-accurate spatial operators in conjunction with a fourth-order time discretization. A grid refinement study has suggested that, under a sufficient time-step restriction, these schemes converge according to the higher-order spatial accuracy, for unsteady problems. The "fifth-order" version of these schemes was successfully tested on the solution of the Euler equations, modeling a shock-vortex interaction. The results of this solution suggest that there is a role for high-order-accurate ENO methods in the prediction of aerodynamic flows that exhibit acoustic phenomena in the presence of shocks.

## Acknowledgment

This work is supported by the Theoretical Flow Physics Branch at the NASA Langley Research Center, Hampton, Virginia, under contract NAS1-18585.

## References

[1]Godunov, S. K., "A Finite-Difference Method for the Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics," *Matthematicheskii Sbornik,* Vol. 47, No. 3, 1959, pp. 271-290.

[2]Van Leer, B., "Towards the Ultimate Conservative Difference Scheme V. A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics,* Vol. 32, No. 1, 1979, pp. 101-136.

[3]Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics,* Vol. 49, No. 3, 1983, pp. 357-393.

[4]Harten, A., and Osher, S., "Uniformly High-Order Accurate Non-Oscillatory Schemes," IMRC Technical Summary Rept. 2823, Univ. of Wisconsin, Madison, WI, May 1985.

[5]Harten, A., Engquist, B., Osher, S., and Chakravarthy, S., "Uniformly High Order Accurate Essentially Non-Oscillatory Schemes III," *Journal of Computational Physics,* Vol. 71, No. 2, 1987, pp. 231-323.

[6]Casper, J., and Atkins, H., "A Finite-Volume High-Order ENO Scheme for Two-Dimensional Hyperbolic Systems," *Journal of Computational Physics* (to be published).

[7]Shu, C., and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes," *Journal of Computational Physics,* Vol. 77, No. 2, 1988, pp. 439-471.

[8]Barth, T. J., and Frederickson, P. O., "High-Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," AIAA Paper 90-0013, Jan. 1990.

[9]Pao, S. P., and Salas, M. D., "A Numerical Study of Two-Dimensional Shock-Vortex Interaction," AIAA Paper 81-1205, June 1981.

[10]Kopriva, D. A., Zang, T. A., Salas, M. D., and Hussaini, M., "Pseudospectral Solution of Two-Dimensional Gas Dynamic Problems," *Proceedings of the 5th GAMM Conference on Numerical Methods in Fluid Mechanics,* edited by M. Pandolfi and R. Piva, Vieweg, Braunschweig, 1983, pp. 185-192.

[11]Meadows, K., Kumar, A., and Hussaini, M., "A Computational Study on the Interaction Between a Vortex and a Shock Wave," AIAA Paper 89-1043, April 1989.

[12]Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechnics and Heat Transfer,* McGraw-Hill, New York, 1984.

[13]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics,* Vol. 43, No. 2, 1981, pp 357-372.

[14]Naumann, A., and Hermanns, E., "On the Interaction Between a Shock Wave and a Vortex Field," AGARD-CP-131, 1973.