JOURNAL OF COMPUTATIONAL PHYSICS 122, 244–265 (1995)

Numerical Simulation of Shock–Cylinder Interactions

I. Resolution

WAI SUN DON[1] AND CARL B. QUILLEN[2]

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

Received December 12, 1994; revised April 10, 1995

We apply two different high-order shock capturing schemes to the study of a two-dimensional unsteady inviscid flow. In particular, we study the interaction of a planar shock with a cylindrical volume of a light gas (helium or hydrogen) contained in air. The two schemes used are the Chebyshev collocation method and the ENO finite difference scheme of Osher and Shu, and they are applied to a physical model consisting of the Euler equations with a real gas equation of state and multiple chemical species. The parallel implementation and low-level coding of the ENO scheme on the Thinking Machines CM-5 results in much higher performance than is possible on a standard serial or vector machine. The ENO code is compared with an existing experimental result and agrees well with it. The results of spectral and ENO calculations are then compared with each other at different resolutions for a Mach 2 interaction. The spectral scheme, though highly oscillatory in nature for discontinuous problems (Gibbs), accurately predicts both large and fine scale structures of the interaction between the shock and the light gas cylinder. Good results can be recovered from the spectral results by post-processing the raw numerical data to remove the Gibbs phenomena. These results are compared with the ENO schemes. The comparison is progressively better as the grid refinement and numerical order of the ENO scheme is increased. This demonstrates definitively the applicability and value of high order schemes to flows with shocks and complicated non-linear physics.

1. INTRODUCTION

Air-breathing SCRAM-jet engines present a difficult challenge to the designer. Extremely rapid combustion is a necessity in these engines, where supersonic flow through the reaction chamber sweeps reactants out of the engine within a few milliseconds. In addition, the reaction must take place well away from the walls of the combustion chamber to prevent excessive heat transfer to the engine itself.

In a combustion process with reactants that are not premixed, the combustion rate is limited by the diffusion rate of reactants across the air–fuel interface. The rate can be increased by stretching out the interface by the motion induced by a point vortex; see Marble [14] for details. This not only increases the total length of the interface, but at the same time it steepens the concentration gradient and thus the diffusion rate of reactants by bringing them in proximity to the interface. Marble *et al.* [15] suggest a way to induce this vorticity—to have a shock pass through a jet of fuel nearly perpendicular to its axis (see Fig. 1). While passing through the region of inhomogeneous density the shock will produce vorticity via the $(\nabla\rho \times \nabla p)/\rho^2$ term in the vorticity equation:

$$\frac{\partial\omega}{\partial t} = (\omega\cdot\nabla)V - \omega\nabla\cdot V + \frac{\nabla\rho\times\nabla p}{\rho^2}. \tag{1}$$

If the fuel (usually hydrogen) and air have significantly different densities, then an intense production of vorticity results from the gradient in pressure across the shock in conjunction with the gradient in fluid density across the interface. The vorticity induced by the normal shock will be concentrated along the fuel-air interface and have the largest magnitude where $\nabla\rho$ is perpendicular to ∇p, e.g., at the top and bottom edges of the circular fuel jet [12]. One might expect that the induced vorticity surface would roll up to form a vortex propagating in the same direction as the shock (for a light fuel). The fuel–air interface should roll up along with the vorticity in this flow, and barring unforeseen interactions between the chemical reaction and the generated flow field, mixing and the resulting combustion should be efficient.

1.1. *Previous Work*

The interaction between shock waves and inhomogeneous regions of density has been studied previously experimentally, analytically, and numerically. The starting point for many has been studying the Richtmyer–Meshkov instability of a shock passing through a perturbed planar interface, beginning with the experiments of Meshkov [17]. Experimental work in this area continued with Zaitsev *et al.* [33] and Brouillette and Sturtevant [4]. Oblique plain interfaces attracted the attention of Bonazza *et al.* [3] and Yang *et al.* [31]. Spherical interfaces were first studied by Rudiger and Sommers [20]. Haas and

---

[1] E-mail: wsdon@hydra.cfm.brown.edu.

[2] E-mail: cbq@hydra.cfm.brown.edu.

0021-9991/96 $12.00

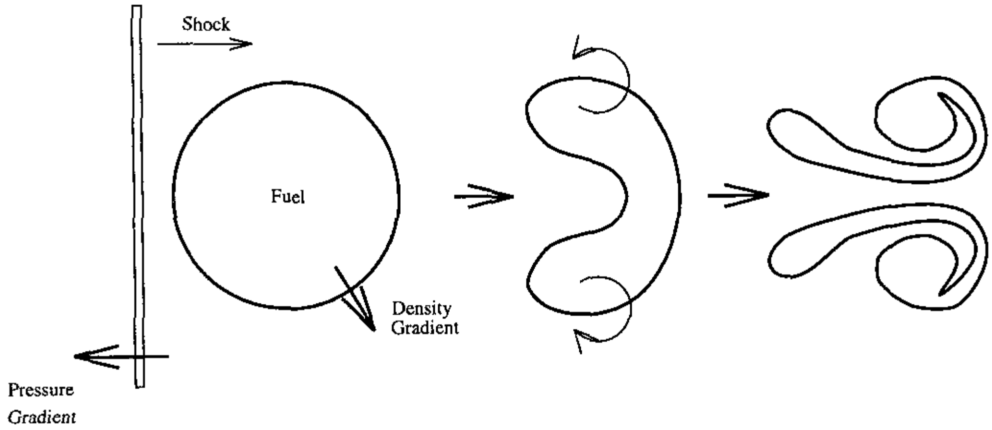


FIG. 1. Roll-up in a shock–cylinder interaction due to induced vorticity.

Sturtevant [11] studied both cylindrical and spherical interfaces experimentally, and analyzed the results in terms of linear geometric optics as well as the Richtmyer–Meshkov instability. These experimental results inspired the numerical work of Picone and Boris [18] and further experimental work of Jacobs [12] concentrating on the shock–cylinder interaction.

More recently a number of researchers have continued these initial attempts at numerical modeling. Among these, Yang *et al.* [32] applied the FCT computation methodology to the conservation equations in order to analyze a shock–cylinder interaction, and derived scaling laws for the amount of circulation produced by the interaction, the time scale of the interaction, and vortex spacing. Ton [29] used a detailed physical real gas model including detailed models of viscosity and chemistry and examined the contribution of these to the production of circulation and the degree of mixing and combustion. Samtaney and Zabusky [21] studied oblique, sinusoidal and cylindrical slow–fast interfaces using the shock polar analysis technique of Henderson [10] to model the interaction and make comparisons with a second order numerical scheme. Recently Quirk and Karni [19] have updated the work of Picone and Boris by carrying out an adaptive high resolution numerical simulation of Haas and Sturtevant's experiment using adaptive mesh refinement (AMR) and a finite difference scheme based on primitive instead of conservative variables. Numerical calculations have begun in three dimensions as well. Waitz [30] has simulated a realistic combustor design using a 3-D code at fairly coarse resolution, and has made a comparison with experimental data. Bell [2] has carried out a fully three-dimensional shock sphere calculation using an adaptive code.

1.2. *High-Order Schemes*

The motivation for the current research has been the recent development of high order shock capturing numerical schemes for the computation of flows with shock waves [7, 5, 23], etc. The fact that these schemes have less artificial dissipation than lower order schemes makes them attractive for computation of an unsteady transient flow, especially one where a solution is required over several tens of thousands of time steps. The presence of large amounts of artificial dissipation in lower order methods suppresses fluid instabilities and results in artificially smooth simulations. Lagging phase errors are another affliction of these numerical methods. Some computational schemes applied previously also suffer from the anisotropic nature of the resulting truncation error, producing odd artifacts that are clearly visible in the computed results [32].

In this investigation, we apply two quite different high order methods. One approach is based upon Chebyshev collocation (pseudospectral) methods [6, 9]. A weak exponential smoothing of the fluxes and the solution is used to stabilize the discretized PDE [7]. With the recent development of sophisticated reconstruction techniques for the removal of Gibbs phenomena [1], the computed raw data can be post-processed to produce a highly accurate solution away from the discontinuity. While there is little rigorous theoretical support for this methodology for a system of nonlinear PDEs, Maday and Tadmor [26] and Tadmor [27, 28] have proved that spectral methods for nonlinear scalar conservation laws using the spectral super-vanishing viscosity formulation do converge spectrally. It is shown in an upcoming paper of Don and Gottlieb that the spectral super-vanishing viscosity method is essentially the same as filtering with a specially designed low pass filter similar to the exponential filter described in Section 3.1. Building on previous successes [5, 7] in simulating shock flows, we push the limits of spectral methods by applying them to reactive flows with real equations of state.

In any scientific computation, it is important to be able to validate results from one numerical code with results from others, preferably those with different numerical methodology. ENO [23, 24] has demonstrated great promise in being a uniformly high order family of methods for shocked problems with essentially non-oscillatory behavior. One of the main stumbling blocks for ENO methods has been the fact that a truly efficient code on standard serial or vector machines has proven to be quite elusive. Here we will address this problem by exploiting the high-speed computational ability of a parallel computer—the Thinking Machines CM-5. The CM-5 architecture allows an optimal implementation of the ENO scheme for fine meshes and permits a high resolution comparison with the spectral scheme.

In Section 2, the physical problem and its governing equations are given. In Section 3, the numerical aspects of this problem are described. Section 3.1 discusses in detail the nonstandard numerical techniques being used in the spectral algorithm. A general discussion of the ENO code is given in Section 3.2 and this is extended in the appendix since the ENO scheme implemented is a standard one. The ENO code is compared for a two dimensional shock–helium jet interaction with both the experiment of Haas and Sturtevant and numerically with the results of Quirk and Karni [19] in Section 4. In Section 5, both the spectral and ENO algorithm are used to simulate a Mach 2 shock–hydrogen circular jet interaction. The time evolution of the flow and the resolution of various schemes are discussed in detail. Some general comments of this study and conclusion are given in Sections 6 and 7, respectively.

2. THE PHYSICAL PROBLEM

In this study, a two-dimensional planar shock (O_2, N_2) in air moving downstream is collided with a cylinder of light gas; see Fig. 1. The light gas is either helium (He) or hydrogen (H_2).

The physics of the problem are described by the two-dimensional Euler equations in conservative form, which we write as

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = 0, \tag{2}$$

where the state vector $\mathbf{U}$ is

$$\mathbf{U} = (\rho\mathbf{f}, \rho, \rho u, \rho v, \rho E)^{\mathrm{T}}. \tag{3}$$

The fluid density is ρ, $\mathbf{u} = (u, v)^{\mathrm{T}}$ is the fluid velocity, E is the specific total energy, and $\mathbf{f}$ is the vector of mass fractions $(f_1, \ldots, f_{N_s-1})$ for $N_s - 1$ chemical species. There are N_s species in the problem. The mass fraction of the last species f_{N_s} may be derived from that of the first $N_s - 1$ by applying the principle of conservation of mass:

$$\sum_{i=1}^{N_s} f_i = 1. \tag{4}$$

In terms of these variables and the pressure p, the Euler fluxes are given by

$$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho f_1 u \\ \vdots \\ \rho f_{N_s-1} u \\ \rho u \\ \rho u u + p \\ \rho u v \\ (\rho E + p) u \end{pmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{pmatrix} \rho f_1 v \\ \vdots \\ \rho f_{N_s-1} v \\ \rho v \\ \rho v u \\ \rho v v + p \\ (\rho E + p) v \end{pmatrix}. \tag{5}$$

2.1. *Thermodynamic Model*

Because the motivating interest in the simulation is reactive flow where gas temperatures are high, the perfect gas equation of state is a reasonable approximation for the pressure:

$$p = \rho \overline{R} T.$$

$\overline{R}$ is a mixture gas constant computed by weighting the perfect gas constant R by the appropriate average of the species molecular weights M_i:

$$\overline{R} = R \sum_{i=1}^{N_s} f_i / M_i.$$

The specific total internal energy is given by

$$E = \int_0^T \overline{C}_p(T)\, dT - \frac{p}{\rho} + \frac{1}{2}(u^2 + v^2) + \sum_{i=1}^{N_s} f_i h_i^0,$$

where h_i^0 is the reference enthalpy of species i and $\overline{C}_p$ is a mixture specific heat at constant pressure. It is computed from the individual species specific heats C_{p_i} by

$$\overline{C}_p = \sum_{i=1}^{N_s} C_{p_i} f_i / M_i.$$

Each C_{p_i} is approximated by a fourth-order polynomial fit in temperature. The coefficients for this approximation were obtained from [16].

3. NUMERICAL IMPLEMENTATION

In two dimensions, a planar shock in air interacts with a hydrogen (helium) circular cylinder (Fig. 1) inside a shock tube. The temperature of the hydrogen and air in the undisturbed region ahead of the shock is set to 1000 K with a pressure of 1 atm. The radius of the hydrogen (helium) cylinder r_0 is 2 cm

(2.5 cm), and reflecting boundaries parallel to the x-axis are placed 6.5 cm (4.45 cm) above and below the axis of the cylinder. The hydrogen (helium) cylinder has an initial diffuse boundary with the air. All elements of the conservative variables **U** and temperature T are modified by a factor

$$s(r) = \exp(-\alpha(r/r_0)^{\delta}), \tag{6}$$

where r is the radial distance from the center of the hydrogen (helium) jet and $\alpha = -\ln \varepsilon$ with ε being the machine zero. δ is chosen to be 16 (200).

Both numerical implementations employ the third order TVD Runge–Kutta scheme of Shu and Osher [24] to solve the system of ODEs produced by the spatial differencing. It has the form of

$$\begin{aligned}\vec{U}^1 &= \vec{U}^n + \Delta t L(\vec{U}^n)\\ \vec{U}^2 &= \frac{1}{4}(3\vec{U}^n + \vec{U}^1 + \Delta t L(\vec{U}^1))\\ \vec{U}^{n+1} &= \frac{1}{3}(\vec{U}^n + 2\vec{U}^2 + 2\,\Delta t L(\vec{U}^2)),\end{aligned}$$

where L is the spatial operator for the fluxes, $\vec{U}^n$, $\vec{U}^{n+1}$, and $\vec{U}^1$ are the two-dimensional data arrays of the conservative variables **U** at the nth and n + 1th time steps, respectively, and $\vec{U}^1$ and $\vec{U}^2$ are these arrays at the intermediate Runge–Kutta stages. Note that this method may be rewritten in such a way that only two levels of storage are required for the Runge–Kutta scheme, namely, $\vec{U}^n$ and $\vec{U}^1$.

3.1. *Spectral Shock Capturing Scheme*

The spectral shock capturing scheme (*spectral code*) used here is essentially the same as the one described by Don [7]. We describe in detail only the additional numerical techniques that we apply here.

1. During the course of the evolution of the flow field, instability occurs for various reasons. The most obvious form of instability in spectral shock capturing methods is the nonlinear instability due to the Gibbs phenomena. This form of instability can be well taken care of by weakly filtering the fluxes **F**, **G** and the solution **U** at each time step. The filter used in this study is the exponential filter described in [7]. For the kth coefficient of the Chebyshev polynomial expansion of a function, the exponential filter has the form of

$$\sigma_k = e^{-\alpha|k/N|^{\beta}}, \quad 0 \le |k| \le N, \tag{7}$$

where N is the number of Chebyshev collocation points (polynomials). The parameter α is chosen so that $\sigma_N = \varepsilon$, where ε is the machine zero, i.e., $\alpha = \ln \varepsilon$. β is the order of exponential filter σ_k. The smaller β is, the more smoothing is applied to the function. $\beta = 12$ is used for the smoothing of **F** and **G**, and $\beta = 10$ is used for **U**. The temperature T is found to require a slightly heavier smoothing with $\beta = 8$.

The other form of instability that can occur is due to spatial scales developing in the flow field that are too small to be properly resolved. For example, the shock compresses the hydrogen cylinder as it passes through. Increasingly fewer mesh points are available to resolve features as the volume of the cylinder decreases. But at the same time, vorticity induced by the shock interaction is creating features with very small length scales within the cylinder. Hence, large oscillations and/or spikes tend to appear at this location, and the result is large errors and oscillations in the fluid density. The situation progresses until a negative fluid density occurs within the flow, and at this point the calculation will fail due to a floating point exception.

Increasing the number of collocation points obviously will only slightly delay, and not solve the problem. There is very little hope of performing a direct numerical simulation at these high Mach numbers. Adaptive mesh refinement could be used to relocate grid points inside the cylinder to certain extent. It is not clear that whether one should track the hydrogen cylinder only or all other conservative variables as well. The algorithm could get extremely complicated and expensive. In any case, it is not clear that it would be successful for spectral methods.

The other way to alleviate this kind of instability is to increase the amount of filtering on both fluxes and the solution, say by using $\beta = 3$ adaptively when negative densities occurs. Once the instability has subsided, the amount of filtering applied can be diminished by increasing β. This does help to stabilize the solution, but at a great expense in accuracy. One additional side effect is that the heavy filtering tends to destroy the fine scale structures. Once they are gone, accurate representation of the evolution and interaction of fine scale structures with large scale structures becomes impossible. The resulting solution is no better than any low order/low resolution finite difference scheme with large artificial dissipation.

To get a handle of this situation, we realize that the lack of resolution within the hydrogen cylinder is a local phenomena along the fuel–air interface. A global smoothing is inappropriate. Knowing that the hydrogen mass fraction should be restricted to the interval $0 \le f_{\mathrm{H}_2} \le 1$, we sound an alarm if $f_{\mathrm{H}_2} > 1 + \varepsilon$ indicating an overshoot. Typically $\varepsilon = 0.01$. We can then determine the location of those collocation points (x_o, y_o) where $f_{\mathrm{H}_2}(x_o, y_o) > 1 - \varepsilon$ with $\varepsilon = 0.01$. At those collocation points (x_o, y_o) the nearest nine grid points are averaged to obtain a smoother value. It should be noted that the nine point averaging must be performed on the primitive variables (f_i, ρ, u, v, T) instead of the conservative variables (ρf_i, ρ, ρu, ρv, ρE), $i = 1, \ldots, N_s - 1$.

If we denote $\mathcal{F}$ as the nine points smoothing operator, then it is clear that

$$\frac{\mathcal{F}(\rho u)}{\mathcal{F}(\rho)} \neq \mathcal{F}(u) \tag{8}$$

for functions which are not smooth. For example, across a contact discontinuity, ρu is continuous and $\mathcal{F}$ would have no effect on the conservative quantity ρu. It follows that $u(\rho/\mathcal{F}(\rho) > u$ at an overshoot and $u(\rho/\mathcal{F}(\rho) < u$ at an undershoot. In other words, this procedure would serve to amplify the oscillatory behavior of u rather than suppressing it. On the other hand, it is true that the smoothing of the primitive variables would result in a minor loss of conservation at selected points along the air–hydrogen interface. This is an acceptable price to pay, when the alternative is an uncontrollable instability due to lack of resolution.

Once the smoothing is done, all relevant conservative quantities $\mathbf{U}$ and variables, for example T, C_p, are recomputed at these selected grid points (x_o, y_o) only. This procedure seems to work fine even for more complicated combustion problems.

2. Since the problem is symmetric about the $y = 0$ axis, only the upper half ($y \geq 0$) of the physical domain needs to be computed. This is a trivial matter for finite difference schemes because they are local methods. A simple reflective boundary condition applied at the center line $y = 0$ is sufficient.

Spectral methods, on the other hand, are global in nature. However, it is wise to take advantage of the symmetry in the flow. A significant saving in terms of memory storage and CPU usage can be realized. There are two ways to achieve this objective. The first is to discretize the PDE in the upper half of the domain $0 \leq y \leq y_{\max}$ only by placing the Chebyshev collocation points there. The second way is to use the full domain, $-y_{\max} \leq y \leq y_{\max}$, but to store only the upper half of data ($y \geq 0$) in the computation. In both cases, the partial differentiation of the flux in x, $\partial F/\partial x$, is computed only for the upper half of the domain ($y \geq 0$), realizing a 50% saving in CPU time.

The first method can offer a very substantial saving in terms of CPU usage. The partial differentiation of the flux in y, $\partial G/\partial y$, can be computed directly from the upper half of the data. However, there is one complication, namely the boundary condition at $y = 0$. A Neumann boundary condition is required there and it is not clear how to do so accurately and stably for a collocation method in a general nonlinear system of PDE. Hence we will have to favor the latter method provided some additional algorithmic manipulations are taken.

The partial differentiation of the flux in y, $\partial G/\partial y$, must be computed by extending the flux $\mathbf{G}$ anti-symmetrically to the negative y, i.e., $\mathbf{G}(x, y) = -\mathbf{G}(x, y)$, $y \geq 0$. The only exception is that the flux term $\rho v v + p$ is extended symmetrically. When smoothing the solution $\mathbf{U}$, however, the opposite is true. Basically, one has to distinguish whether the function being differentiated or smoothed is symmetric or anti-symmetric about the $y = 0$ axis and extend the data accordingly. Then the full differentiation in y may be performed. Only the upper half of data are stored for later use realizing a 50% saving in memory storage.

Since the differentiation and solution smoothing routines together form the main computational kernel of the spectral code, any savings in terms of CPU and/or storage in those routines is much appreciated. Using the Cray C90, this numerical technique reduces the core memory requirement from 34 mega-words to 22 mega-words and CPU time usage from 1.81 to 1.26 s per Runge–Kutta time step for a 512 × 512 grid based on the Chebyshev collocation points.

One additional numerical technique that can be used to exploit the symmetry of the problem in the y direction is the quarter-wave transform [25]. The implementation of such a scheme will be studied in future work.

3. Since the flow is confined within the physical domain at all time (for the spectral code), a simple boundary condition can be used. For the supersonic inflow, all flow quantities are specified using free stream values. The subsonic outflow is set to the previously computed values. The upper wall uses reflective boundary conditions, i.e., $v(x, y_{\max}) = -v(x, y_{\max})$.

4. For the number of collocation points ($\gtrsim 512$) used in this study, the fastest known method of computing derivatives and smoothing the solution is by the fast cosine transform (CFT), despite a non-optimal roundoff error of order $O(N^3\varepsilon)$, where ε is the machine zero [8].

5. The solution at any given time, though highly oscillatory, can be post-processed to regain an accurate, non-oscillatory solution away from the shock. The procedure used follows very closely the methodology applied by Don [7]. The procedure used in this research consists of two steps, namely, at each location $0 \leq y_j \leq y_{\max}$,

A. the location x_s and the strength J of the discontinuity are determined from the moments of the oscillatory discontinuous solution ($U_N(x)$ by a least squares fit of $U_N(x)$ to a step function $S_N(x, x_s, J)$,

B. $\hat{U}_N(x) = \mathcal{F}(U_N(x) - S_N(x, x_s, J)) + S(x, x_s, J)$, where $\mathcal{F}$ is the filtering (smoothing) operator, S is the Heaviside function with jump J at x_s and S_N is the N degree polynomial approximation of S.

Even though the global filter $\mathcal{F}$ with $\beta = 2$ (see Eq. (7)) can be used for the term $U_N(x) - S_N(x, x_s, J)$ and yields a reasonably good global solution, the fine scale structures within the hydrogen cylinder can be severely affected by the strong smoothing. To remedy this situation, the hydrogen mass fraction is used to delineate the boundary between the air and hydrogen interface. The interior of the hydrogen is determined by checking if the hydrogen mass fraction f_{H_2} is greater than some tolerance *tol* with *tol* = 0.01 typically. By marking the interior of the hydrogen cylinder, the global strong smoothing $\beta = 2$ is then applied to solution outside the hydrogen cylinder while a slightly weaker one $\beta = 3$ inside.

For those variables without discontinuous jump in the function, for example f_{H_2}, a simple smoothing with $\beta = 3$ is applied. No attempt is made to reconstruct the solution in the y direction other than by using a global smoothing with $\beta = 3$.

3.2. *ENO Code*

The ENO schemes of Osher and Shu are a class of high order finite difference schemes in conservative form. The reader is referred to [23, 24] for a detailed introduction. A brief summary of these ideas is also given in the Appendix, as well as a description of how to apply the ENO methodology to a real-gas simulation with multiple chemical species.

In this paper, we present results derived from an implementation of these ENO schemes on a CM-5. A straightforward domain decomposition was used to implement the method in parallel. Each vector unit maintains a separate rectangular submesh, with a border of "ghost" boundary points surrounding the submesh. For a scheme of numerical order N (in the L_1 norm) the border is N grid points wide. At each partial time step in the Runge–Kutta solver these ghost points are initialized from data at the adjacent subgrid, or if they are at the boundary of the numerical domain, they are initialized by applying the boundary conditions described in the appendix.

The program is written using a low-level compiler (GCC-AC) for efficiency. The computational kernel is implemented in vector-unit assembly language (CDPEAC). Performance on a 32-node machine is well in excess of one Gigaflop for the grid sizes considered here.

Both the third-order (ENO-3) and fifth-order (ENO-5) spatially accurate (in L_1 norm) schemes are used in this study. We used the standard flux based ENO scheme based on the line-wise global Lax–Friedrich flux splitting. The stencil biasing technique of Shu [22] is used to improve the stability of the fifth-order scheme.

4. ENO COMPARISON WITH EXPERIMENT

Two experimental results are available that permit a comparison of numerical experiments with reality. Haas and Sturtevant [11] studied in one set of experimental runs shock interactions with a helium cylinder. The cylinder is 5 cm in diameter, placed in an experimental section 8.9 cm wide. All gases are at atmospheric pressure and room temperature, taken as 298 K. In the experiment, the cylinder was contained by a nitrocellulose microfilm, which permitted some gaseous diffusion. The helium was estimated to be contaminated 28% by mass with air. The air surrounding the cylinder was continuously refreshed, and helium contamination was observed to be small enough to ignore. The Mach number of the incident shock was either 1.085 or 1.22. Visualization in the experiment is by spark shadowgraph, which provides a reasonably precise determination of the interface between helium and the surrounding air. The well-defined nature of the initial condition in this experiment, and the fact that the researchers took great care to provide accurate quantitative measurements of shock speeds allow a comparison with numerical simulations. The results of an interaction with a Mach 1.22 shock simulated via the ENO code is presented in the next section. This provides a good benchmark for our code.

Jacobs [12] presents an experiment with a helium cylinder interacting with shocks of Mach number 1.093 or 1.15. Unfortunately, the cylinder was created by introducing a helium jet at the bottom of an experimental section of a shock tube. The jet is then vented at the top of the section. The requirement that the jet retain a stable cross section limited the flow rate of the helium to 65.1 cm^3/s, emanating from a 0.9525 cm diameter nozzle. This low flow rate permits a substantial diffusion of helium into the surrounding air. Buoyancy also substantially affects the jet by thinning it. Flow visualization was obtained by seeding the helium by a fluorescent gas (biacetyl) and illuminating by a planar laser beam. The visualization thus avoids being contaminated by three dimensional effects due to the finite height of the helium cylinder, and this is an advantage of the experimental setup. Unfortunately the disadvantages of this apparatus seem to outweigh the advantages. The heavy biacetyl gas diffuses much more slowly into air than helium, and the extent of the diffuse helium air interface is not apparent. This presents two difficulties for numerical reproduction of the experiment. The first is that one can only guess at what initial conditions to use. The other is the fact that the very diffuse helium air interface is known to have the effect of suppressing perturbation growth generated by the Richtmyer–Meshkov instability. The resulting images, taken from the center of a diffuse jet, are very smooth. They obviously cannot resolve the issue of how three dimensional effects affected the helium air interface in Haas's experiment, as the interface is more stable and in any event, perhaps not visualized at all.

Both of these experiments applied weak shocks to the helium cylinder. The low Mach number of these shocks prohibit simulation by use of the spectral code applied here. Spectral codes place more demands on the applied boundary conditions in order to achieve stability. A consequence of the limited abilities of the current boundary conditions is an inability to handle the subsonic inflow conditions present in this type of problem.

4.1. *Numerical Comparison*

The result of the ENO numerical simulation of a Mach 1.22 shock–helium cylinder interaction (see Fig. 2) may be compared with the experiment in Fig. 3. The experimental shadowgraphs provide a visualization of the magnitude of the second derivative in density [13]. A reasonable comparison may be obtained with a numerical Schlieren photograph, which provides a picture of the norm of density gradient. We follow the methodology of Quirk and Karni [19] and plot a grayscale image of

$$\exp(-k|\nabla\rho|/\max(|\nabla\rho|)).$$

Esthetically pleasing results are obtained with $k = 120$. A diffuse interface between air and helium was used for the numerical initial condition. $\delta = 200$ was chosen in Eq. (6) so that the width of the interface comprised roughly three numeri-

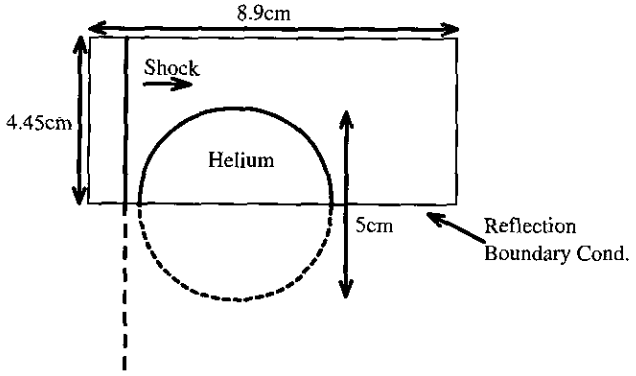


FIG. 2. Computational geometry used to emulate Sturtevant and Haas's experiment.

cal grid points. The radius of the helium cylinder $r_0 = 2.5$ cm as shown in Fig. 2.

Figures 3A–3H provide a picture of the early time behavior of the flow as the shock initially compresses the helium cylinder and refracts through it. We compare them with Haas and Sturtevant's results (frames a–h). Because the times measured experimentally did not have high absolute precision, some discrepancies may be seen in the relative position of shocks, especially in the earlier pictures.

Due to the low Mach number one might expect linear geometrical optics to provide a qualitative explanation of the results. At time 32 and 52 μs the shock reacts to the higher sound speed in helium by refracting and advancing ahead of the surrounding shock front in air. Meanwhile, the shock undergoes a complicated form of Mach reflection as it diffracts around the exterior of the cylinder, forming a four shock intersection. At 62 μs, the refracted shock emerges from the cylinder, leaving behind an internal reflection which forms two cusps inside the cylinder. At 82 μs, the shocks derived from the first internal reflection inside the cylinder emerge and cross just in front of it.

The later pictures present the long time evolution of the cylinder as induced vorticity causes it to first assume a kidney shape and then split into two vortices propagating in the streamwise direction. Reflections returning from the top boundary, provide a complex picture in frame (F). Initial perturbations in the surface of the cylinder continue to grow causing mixing at the interface.

It is worth making an additional remark regarding the last picture at time 983 μs. The vortex is well developed by this stage in the flow evolution, and it is causing fluid to be entrained inside the cylinder, effectively splitting it in two. The entrained fluid appears to be forming a "mushroom" shape as it penetrates into the cylinder from the top. This "mushroom" seems to be a robust feature of the flow. We have seen it in a variety of different simulations with different flow conditions. If the numerical scheme has enough resolution, it almost always seems to be there. But observing this phenomenon does require a high resolution numerical simulation, and we will use this fact later to provide an indication of when flow features are being properly resolved.

In Fig. 4 and Table I, we present a comparison of shock speeds and the speed of various elements of the flow field and compare them with Haas and Sturtevant's results. The results agree about as closely as might be expected. The most significant errors are obtained with the measurement of the vortex propagation speed V_v and the up-stream interface space V_{ui}. The error in these results may mostly be due to misinterpreting the definitions used by Haas and Sturtevant. The position of the upstream interface initially does not possess a constant velocity—it is accelerated over some distance by the flow. The velocity derived by a line fit thus varies depending on how many points are used. Values of velocity may be derived that vary from 160 to 200 m/s. We chose a high value within the 10% experimental error as a compromise between honesty and a desire to indicate that values measured by the experiment might be lower than in a calculation which ignores the contribution of the microfilm present in the experimental apparatus. The definition of the vortex head is not precisely clear. It was taken as the asymptotic slope of the downstream interface. We omit the measurement of V_{df} present in Haas and Sturtevant, as it is unclear where to measure it in the flow field. Another obvious comment to make about the measured speeds is that it is unfair to criticize Quirk and Karni's method for being only approximately conservative. The conservative ENO method essentially duplicates the shock speeds they reported.

These numerical results are obtained by applying a fifth-order ENO method for the spatial discretization, and a third order TVD Runge–Kutta solver in time. The computation is performed on a 496 × 248 point uniform grid, and it uses a moving coordinate system that translates with approximately the same mean velocity as the cylinder after the interaction. The characteristic boundary conditions seem to be sufficient to permit shocks to leave the computational region. Reflecting waves from these boundaries appear to be entropy waves which have a strength of approximately 1% of the pressure of the outgoing shock. For the times measured here, these entropy waves do not succeed in propagating back close enough to the cylinder to cause interactions.

5. MACH 2 SHOCK–HYDROGEN INTERACTION

As can be seen in the previous section, Haas and Sturtevant's shock–helium cylinder interaction is well simulated by an Euler code, even with no attempt made to model viscosity. But this experimental problem is not an ideal setting for studying shock–cylinder interactions.

The first objection is that the flow geometry itself complicates the phenomena under study. The cylinder is more than one-half the width of the experimental channel. Blockage effects in the flow field should thus be quite important. From the point

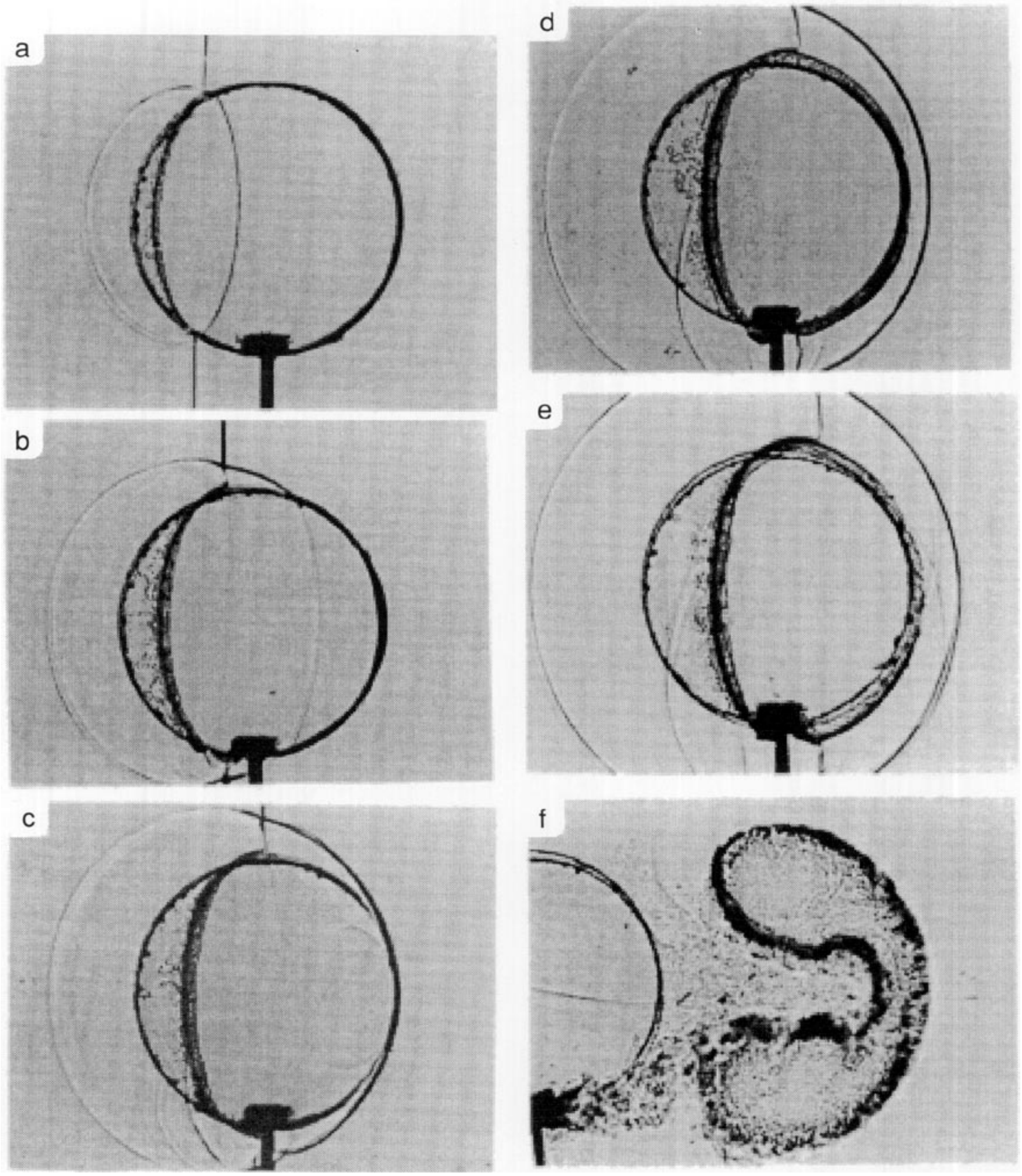


FIG. 3. Comparison of the experiment (a–h) and ENO calculation (A–H) for the Mach 1.22 shock helium cylinder interaction. The corresponding times are (a, A) 32 μs, (b, B) 52 μs, (c, C) 62 μs, (d, D) 72 μs, (e, E) 82 μs, (f, F) 472 μs, (g, G) 674 μs, and (h, H) 983 μs. Experimental pictures reproduced with the permission of Cambridge University Press from Hass and Sturtevant [11].

of view of studying how fluid instabilities are driven by the shocks present in the flow field, the picture is considerably complicated by the fact that reflections from the top and bottom walls of the flow section return quickly to influence the cylinder. The nearly discontinuous initial condition ensures that small wavelength perturbations will be amplified by the fluid instabilities [4]. A less complicated evolution may be obtained by starting with a more diffuse interface. Finally, once vortices form and begin propagating, they will be influenced by their images due to reflection and the boundaries, and travel faster than otherwise would be expected.

Another objection arises purely for numerical reasons. The

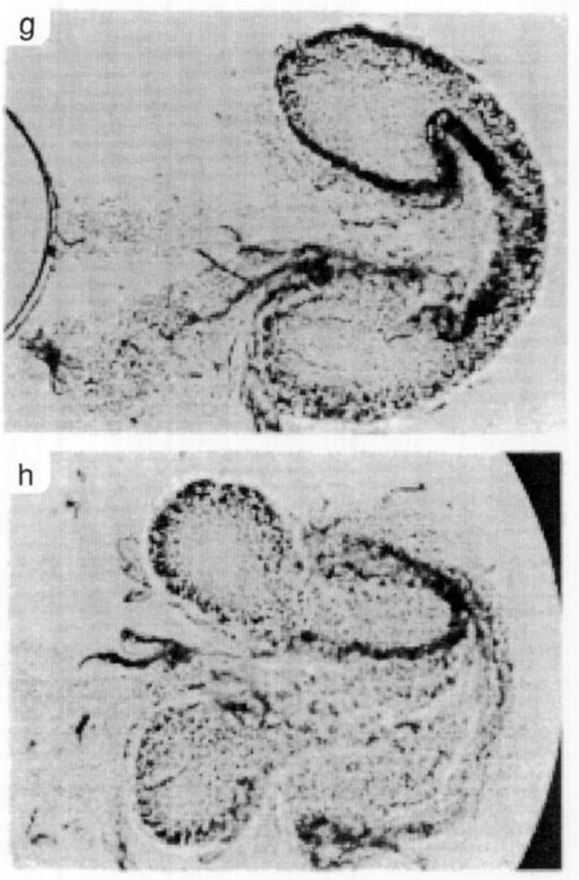


FIG. 3—*Continued*

central concern with these numerical simulations is the fact that in long time integrations of the flows, errors (dissipative, dispersive, roundoff, to name a few) arising from the spatial discretization are cumulative. Eventually the solution can become unreliable. The low Mach number of the interaction and denser gas within the cylinder imply that the characteristic time scale for the flow evolution will be quite slow (see Yang *et al.* [32] for a derivation of flow time scales). A faster Mach number and hydrogen within the cylinder imply a much faster time scale, and this substantially reduces the number of time steps required by the numerical scheme, even taking into account the reduced time step implied by the CFL restrictions of an explicit code. The resulting computation is far less computationally demanding, and it relies much less on the accuracy of the time integration method. The result is a much better setting in which to determine how schemes with different spatial accuracies affect the outcome of the simulation.

5.1. *Evolution of the Shock–Hydrogen Interaction*

In this case, a Mach 2 planar shock located at $x = 0.5$ cm moves downstream and interacts with a circular cylinder of hydrogen with center at $(x_c, y_c) = (2.75$ cm, 0 cm$)$ and radius $r = 2$ cm (see Fig. 5).

In Fig. 6, the density of the spectral simulation and the two best complete ENO runs for the Mach 2 shock interaction with the hydrogen cylinder at various time $t = 20, 40, 60, 80, 120$ μs are shown. All density contour plots have 40 contour lines between 0.05 and 1.2.

The ENO solutions are based on the third order version with 1472×736 grid points (left) and the fifth order version with 752×376 grid points (middle). The rightmost figure is the raw data of the spectral simulations with 512×512 Chebyshev collocation points. The post-processed data of the spectral solution is given in the left most column of Fig. 7. The size of the domain used is $0 \leq x \leq 20$ cm, -6.5 cm $\leq y \leq 6.5$ cm. In Fig. 7, the same spectral code is run with the same parameters as the previous case with the exception that the domain size in x is reduced to $0 \leq x \leq 12.5$ cm. With the same number of Chebyshev collocation points, this increases the effective resolution of the solution for the smaller domain, or equivalently increases the number of collocation points to $\approx 800 \times 512$ in the larger domain. Both the raw and post-processed density solution are given in the middle and the rightmost column respectively. The time shown are $t = 20, 40, 60, 80$ μs.

The evolution of the flow in this case is very similar to that presented by the Mach 1.22 shock–helium cylinder interaction. The shock passes through the cylinder compressing it. The same four-shock intersection may be seen as the shock diffracts around the circular fuel-jet. A vortex forms, and it entrains fluid inside the cylinder. The difference in this case is the much higher speed of the interaction—the vortex forms while the shock is still diffracting around the cylinder.

It is worth noting that the vortex forming at time 40 μs in the spectral simulation (see middle column of Fig. 7) was not seen for low resolution/low order schemes. The vortex manifests itself as a growing instability in a kink at time 20 μs in the upstream the air–hydrogen interface. This kink was dismissed as a numerical artifact in the beginning of our research. We thought that it was a manifestation of the numerical "ringing" seen when using spectral methods for discontinuous solutions (Gibbs phenomena). It was a surprise when its existence was confirmed by running higher resolution and higher order ENO calculations. This points out the fact that the spectral methods can still retain high order information when applied to discontinuous problems, even when the computed solution is very oscillatory.

Numerically, despite the oscillatory pictures of the raw density data, the post-processed density data agree quite well with the high resolution/high order ENO schemes. The large scale features are basically consistent between the spectral and the high resolution ENO results. The most variation is seen in the internal structure of the hydrogen cylinder, and this is best seen by plotting the mass fraction f_{H_2} of hydrogen. All hydrogen mass fractions are plotted with 15 contour lines in the range 0.01–1.01.

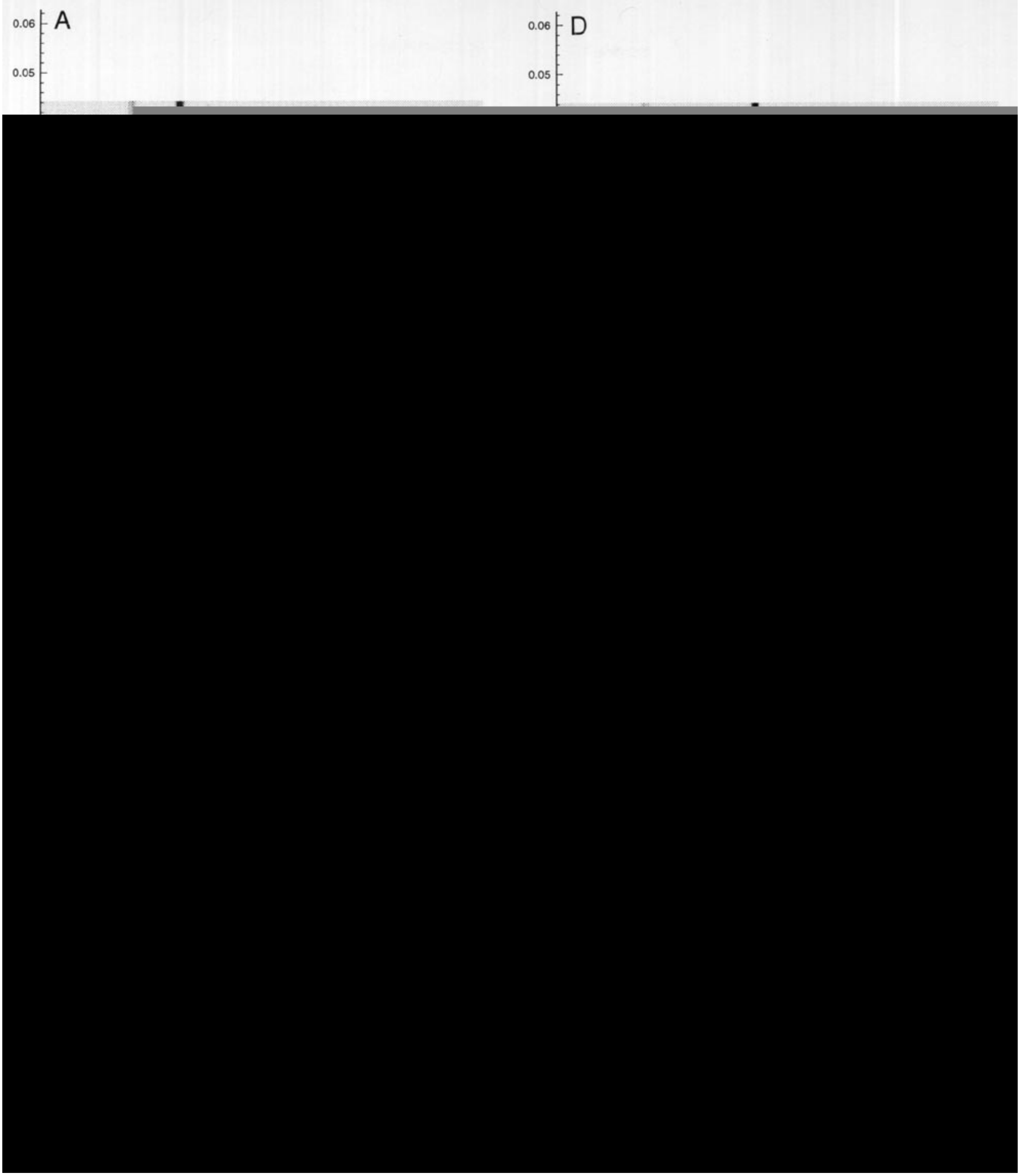
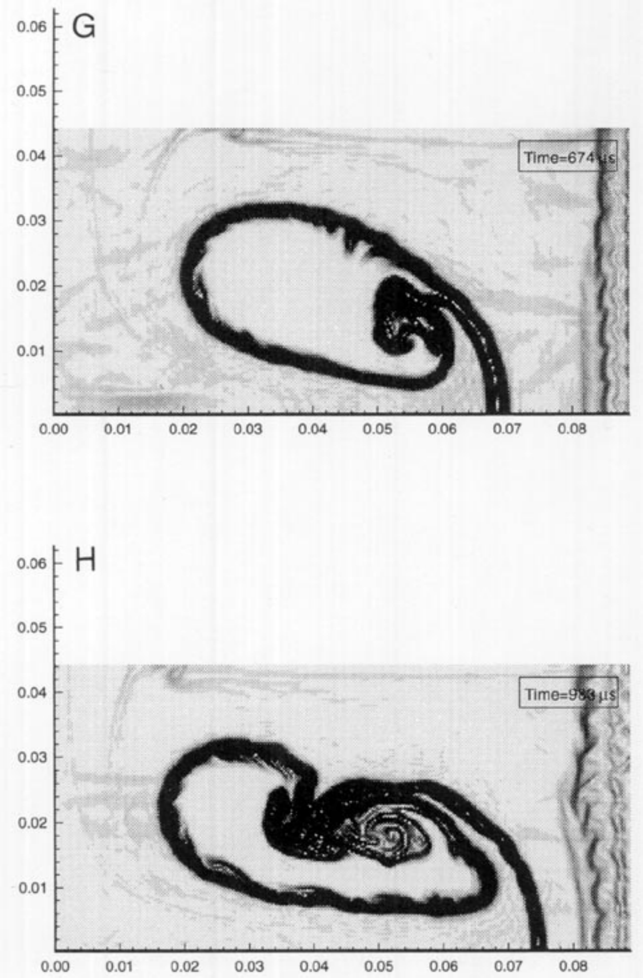


FIG. 3—*Continued*



Time = 40 μs. A consistent picture can be seen as early as time 40 μs. Figure 8 shows the results of the ENO third-order solution for grid size 496 × 248, 752 × 376, and 1472 × 736. Figure 9 shows the results of the ENO fifth-order solution for grid size 752 × 376 and 1472 × 736. It also shows the solution obtained by the spectral code with 512 × 512 Chebyshev collocation points.

At time $t = 40\ \mu$s, a small narrow jet of fluid begins penetrating the cylinder from the interior of the air jet and it rolls up to form a vortex. It is clearly seen in the figures as soon as enough grid resolution is attained. In a few relatively low grid resolution cases, the formation of the vortex is not as well defined as at higher grid resolutions or as in the spectral solution due to numerical dissipation. In general, these figures seem to be consistent and similar to each other. However, we might expect that the lower order ENO schemes would have increasing difficulties as later times are considered.

Time = 60 μs. At time $t = 60\ \mu$s, a jet of air penetrates the back top side of the cylinder for both the third-order and fifth order ENO schemes, and the spectral code (Figs. 10–12).

A "mushroom" shape rollup of the air can be seen clearly in Fig. 12. It is related to the Rayleigh–Taylor instability which describes denser fluid lying above a lighter one in a gravitational field. The heavier fluid forms fingers descending down into the lighter fluid below. A similar fingering can result from the impulsive acceleration due to a propagating shock.

In the lower resolution cases this feature is captured only as a single jet that rolls up into a vortex. Higher resolution images provide a complicated picture that is rapidly becoming turbulent. Surprisingly the third-order calculation on the 1472 × 736 grid produces a simple jet without a mushroom. But it is visible for the fifth-order scheme on the 752 × 376 grid, which still compares fairly closely with the spectral scheme. At later times, however, features increasingly disappear due to numeri-

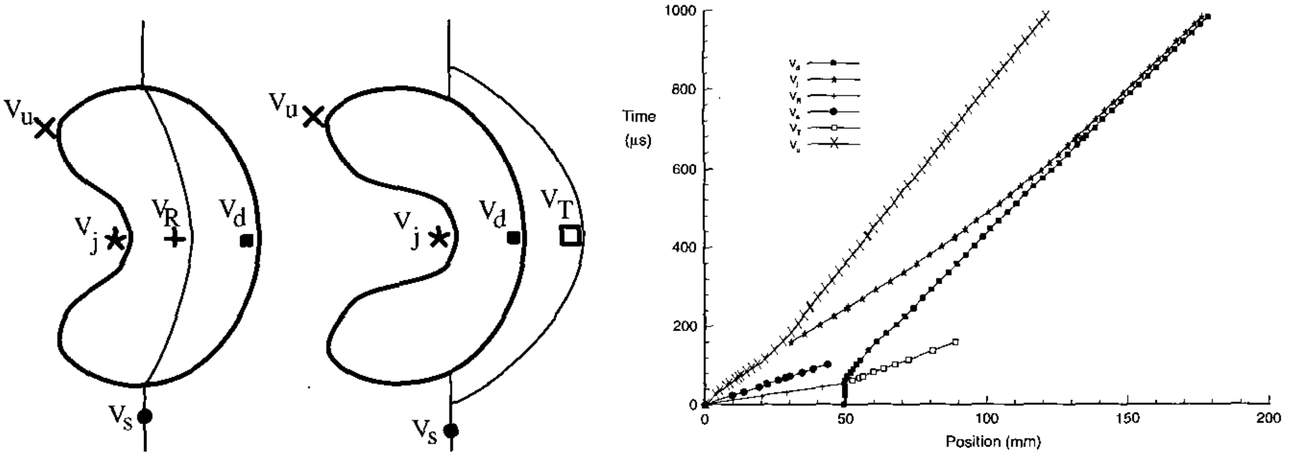


FIG. 4. Mach 1.22 shock–helium interaction. (Left) Locations where velocities are measured. V_s, shock velocity; V_R, refracted shock, V_T; transmitted shock; V_u, upstream edge; V_d, downstream edge; V_j; air jet head. (Right) x–t diagram.

TABLE I

Velocities Measured from the ENO Calculation Compared with Haas and Sturtevant's Experiment and Quirk and Karni's Calculation

	Velocity							
	V_s	V_R	V_T	V_{ui}	V_{uf}	V_{di}	V_j	V_v
Computation	424	936	377	186	115	144	226	146
Experiment	410	900	393	170	113	145	230	128
Quirk and Karni	422	943	377	178	—	146	227	—
Our error	+3.4%	+4.0%	−4.1%	+9.4%	+1.8%	−0.7%	−1.7%	+14%

cal diffusion, and the ENO schemes presents a misleadingly simple picture of the internal structure (see Figs. 6 and 7).

6. DISCUSSION

The truly surprising result of the calculations described above is the amount of resolution that is required to properly resolve features within the circular fuel-jet. Pictures consistent with the expected physics are visible at quite low resolutions. For example, a beautiful roll-up in the cylinder due to vortex formation is visible in the third order ENO calculation on the 752 × 376 grid (Fig. 10). This picture is essentially unchanged for somewhat lower resolutions. Since this is precisely the physics that one might expect, it is natural to believe that the flow has been resolved.

But the third-order 1472 × 736 calculation is beginning to show that the picture is not as simple as one might have hoped, and higher order calculations demonstrate clearly that more complicated phenomena are afoot. The spectral scheme provides especially valuable hints in this regard. It demonstrates these phenomena at lower grid resolutions and computational cost.

To demonstrate that this general picture remains valid even when the physical model is substantially changed, we present one plot derived from a real gas Navier-Stokes simulation which includes chemistry modeled by a one step reversible reaction $2H_2 + O_2 \leftrightarrow 2H_2O$. We show the hydrogen mass fraction raw data at time 55 μs in the right-most picture of Fig. 12. Comparing with the left-most picture, the shape of the mushroom and the internal structures are slightly modified by the combustion and viscous process, but the general features are quite similar for both the Euler and the reactive Navier–Stokes simulations.

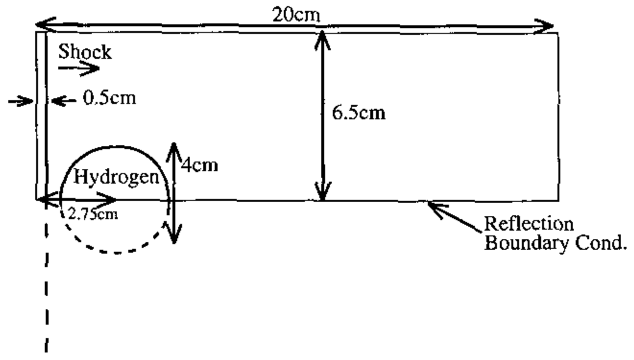


FIG. 5. Computational geometry used for the Mach 2 shock–hydrogen cylinder interaction.

The results presented here hint very strongly at the necessity for high order schemes for calculating long-time evolution of flow fields. The major disadvantage these schemes have is the difficulty in applying these methods to irregular adaptively refined grids. The particular ENO scheme chosen is based on calculating derivatives of fluxes. It requires four times less computational work to implement this scheme than a finite volume ENO scheme (based on cell averages) in two dimensions. But unlike the finite volume schemes, there is currently no variant of the method with higher than second order accuracy that remains conservative when the grid is locally refined.

The computational burden of this particular flow problem may be greatly reduced by taking advantage of an adaptive grid. Quirk and Karni [19] reported that the cost may be decreased by up to a factor of 50 using the adaptive mesh refinement, especially for earlier times in the problem. The fact that they used a second order method does affect the reliability of their results for late times, and might contribute to a reluctance on their part to report these results.

An additional complication of adaptive grids may be worth mentioning—the effective numerical viscosity present in their scheme depends on the local grid resolution, and this varies considerably over the computational region. It is possible that this introduces local perturbations in the flow which are amplified by the fluid instabilities. The results depicted in Fig. 3 use a regular grid, and the method chosen to do upwinding is a Lax–Friedrich flux using a wave speed chosen as the maximum over an entire line in the computational region, as described in Section A.1. Thus the effective numerical viscosity is quite uniform. If we change the flux to a local Lax–Friedrich flux, the numerical viscosity becomes more irregular. In Fig. 13, we present the results of a calculation using a local Lax–Friedrich flux were the maximum wave speed over the nearest six grid points is used. The result is very different from the previous case (Fig. 3H). The vortex has broken up, and the "mushroom" formed by the air penetrating the downstream side of the cylinder is not recognizable. The picture presented is much less realistic and correlates much less well with the Mach 2 case.

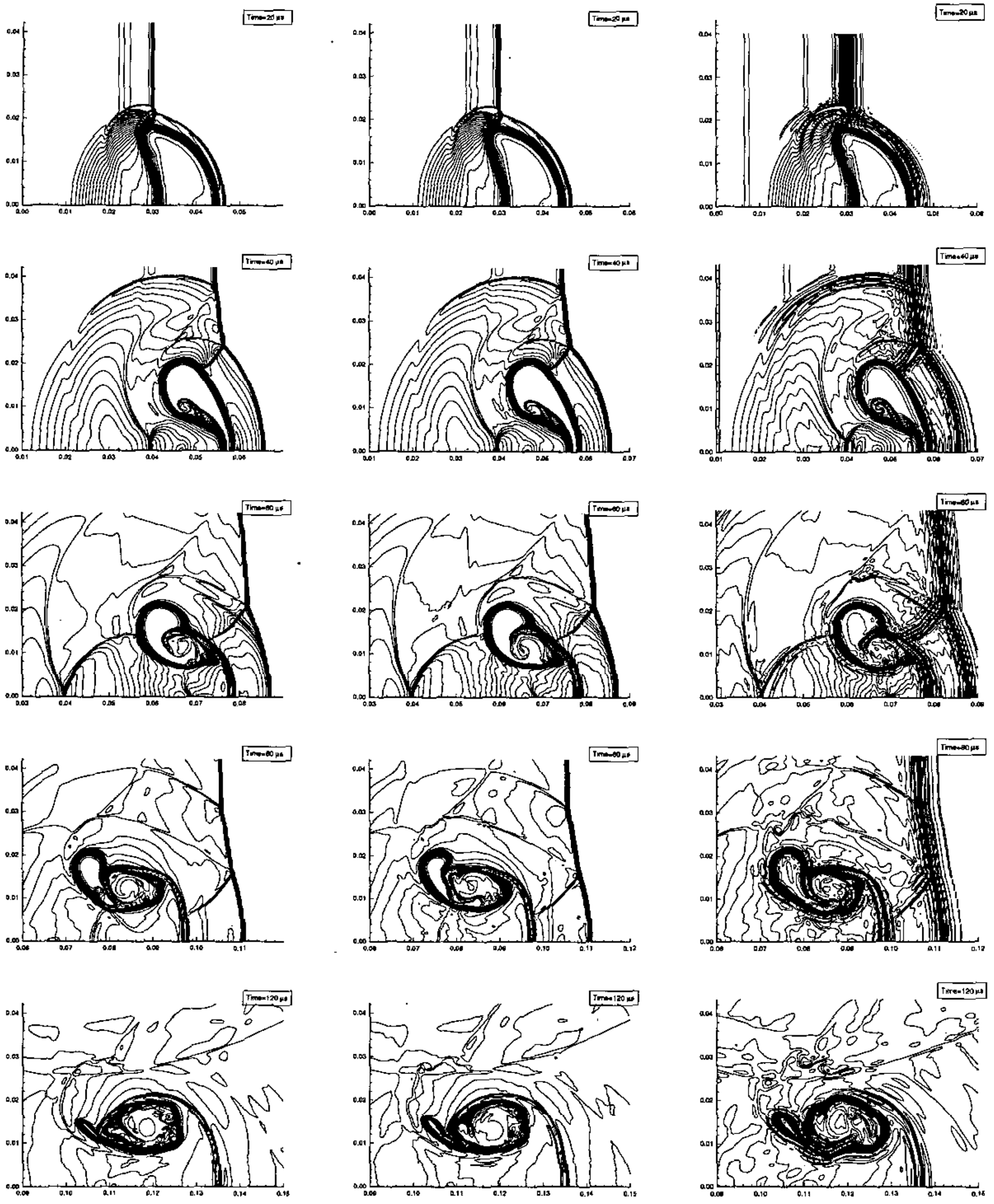
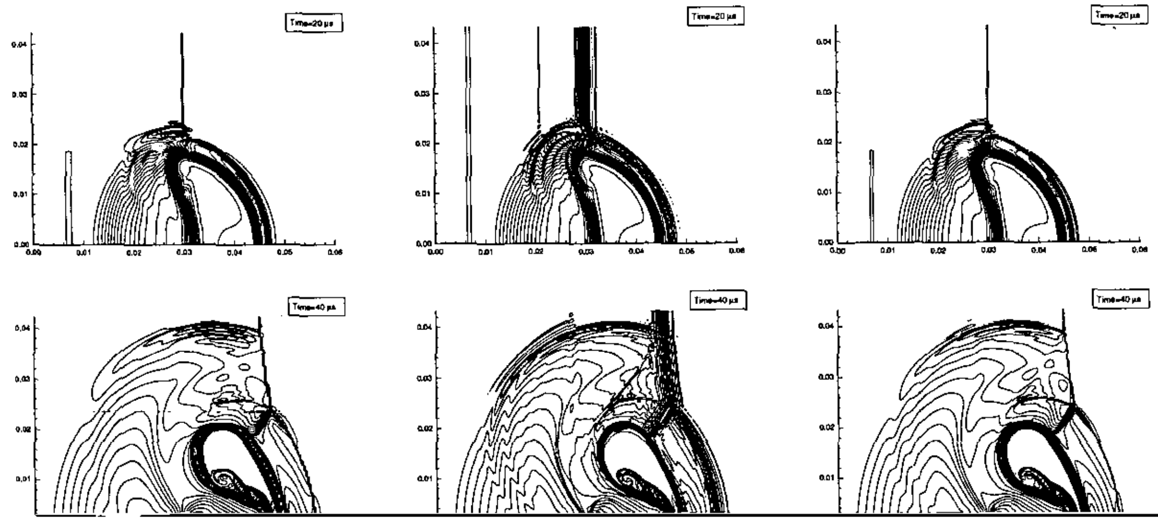


FIG. 6. Density contour plot of the ENO code versus specral code for the Mach 2 shock–hydrogen cylinder interaction at $t = 20, 40, 60, 80, 120$ μs. ENO third order with 1472 × 736 grid (left), ENO fifth order with 752 × 376 grid (middle) and Spectral raw data with 512 × 512 Chebyshev collocation points (right) with $0 \leq x \leq 20$ cm, -6.5 cm $\leq y \leq 6.5$ cm.

6.1. *Computational Cost*

As the previous discussion makes clear, high order schemes provide more accurate solutions and resolve features better for a given grid resolution. They are computationally much more demanding. But for a given amount of CPU resources, they seem to provide better resolved results. The fifth-order ENO schemes require roughly 60% more CPU time per time step, at a CFL number $\frac{1}{3}$ to $\frac{1}{2}$ that of the third-order schemes, and so are up to five times as expensive. But for these explicit schemes,


Time=20 µs
Time=20 µs
Time=20 µs
Time=40 µs
Time=40 µs
Time=40 µs
0.00
0.01
0.02
0.03
0.04
0.05
0.06

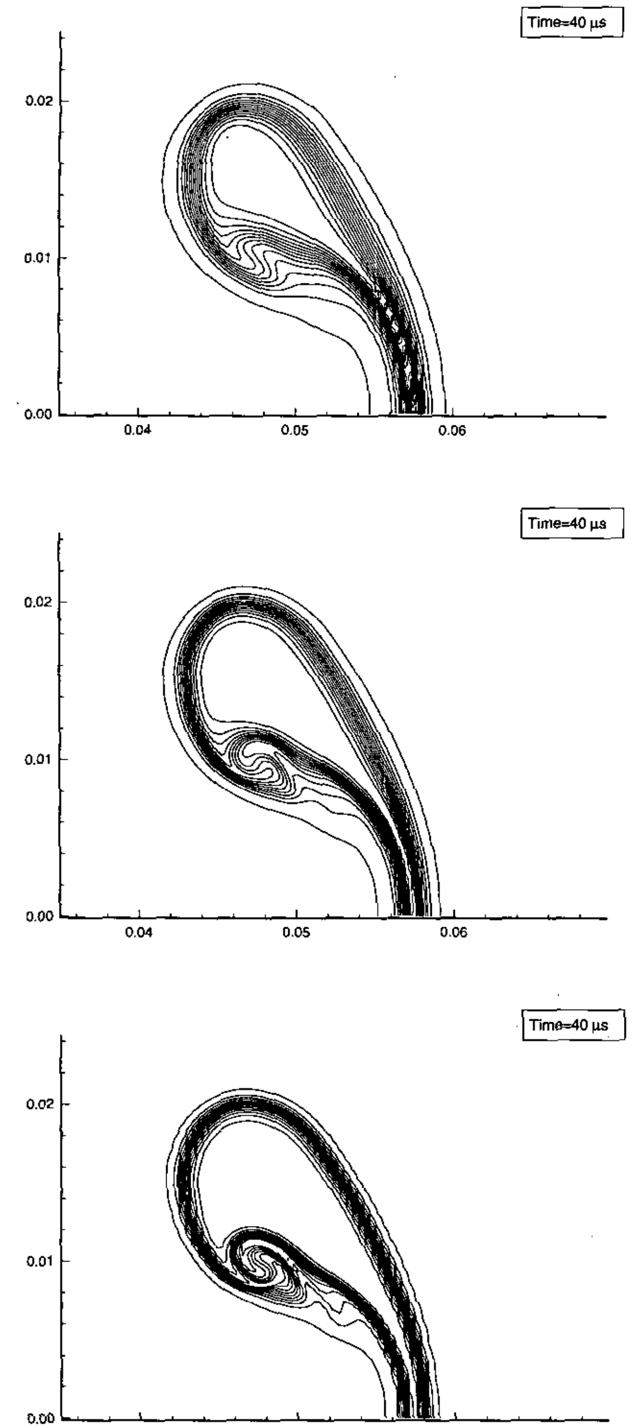


FIG. 8. Hydrogen mass fraction for the Mach 2 shock–hydrogen cylinder interaction at $t = 40$ μs. ENO third order with grid size 496 × 248 (top), 752 × 376 (middle) and 1472 × 736 (bottom). The domain size is 0 cm ≤ x ≤ 20 cm, −6.5 cm ≤ y ≤ 6.5 cm.

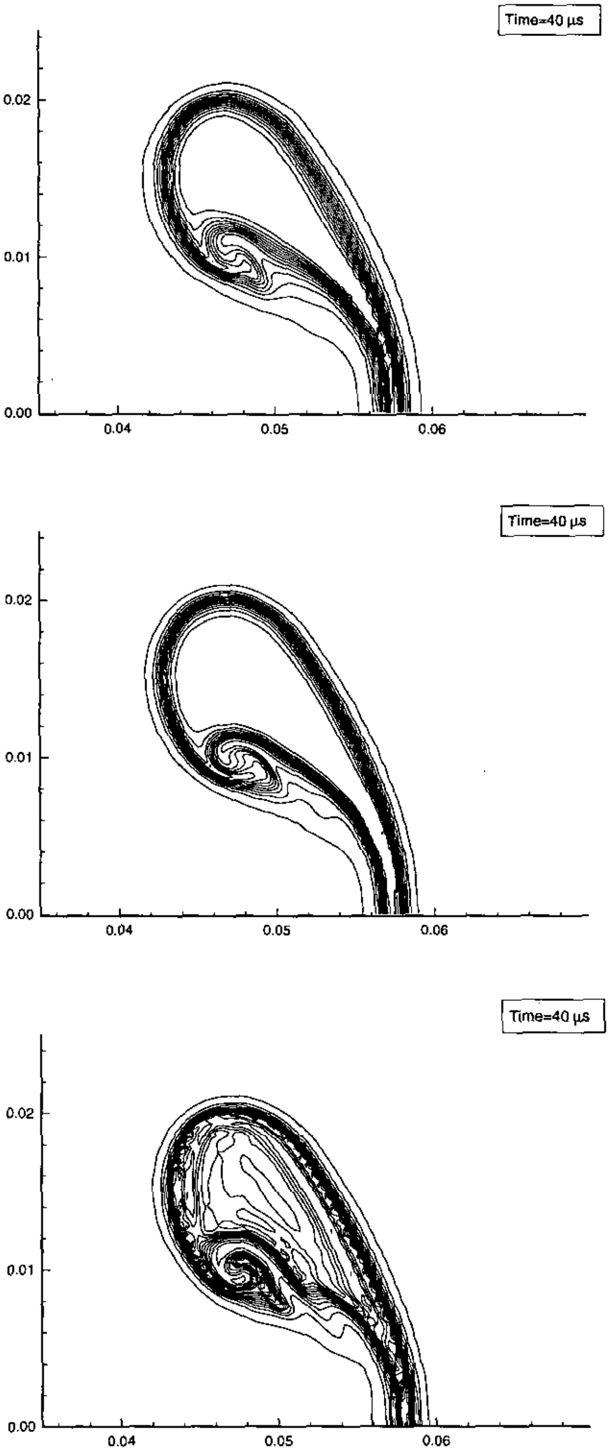


FIG. 9. Hydrogen mass fraction for the Mach 2 shock–hydrogen cylinder interaction at $t = 40$ μs. ENO fifth order with grid size 496 × 248 (top), 752 × 376 (middle), and spectral raw data with 512 × 512 Chebyshev collocation points (bottom). The domain size is 0 cm ≤ x ≤ 20 cm, −6.5 cm ≤ y ≤ 6.5 cm.

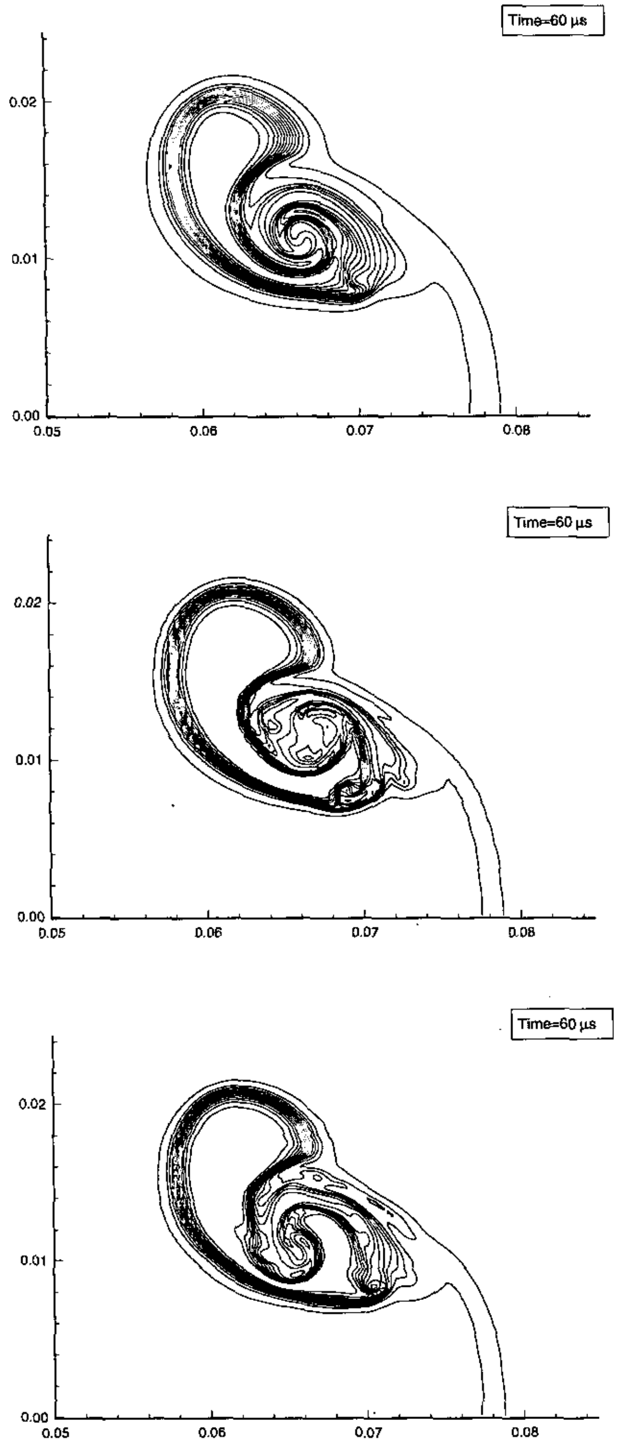


FIG. 10. Hydrogen mass fraction for the Mach 2 shock–hydrogen cylinder interaction at $t = 60$ μs. ENO third order with grid size 752 × 376 (top), and 1,472 × 736 (middle). ENO fifth order with grid size 752 × 376 (bottom). The domain size is 0 cm ≤ x ≤ 20 cm, −6.5 cm ≤ y ≤ 6.5 cm.

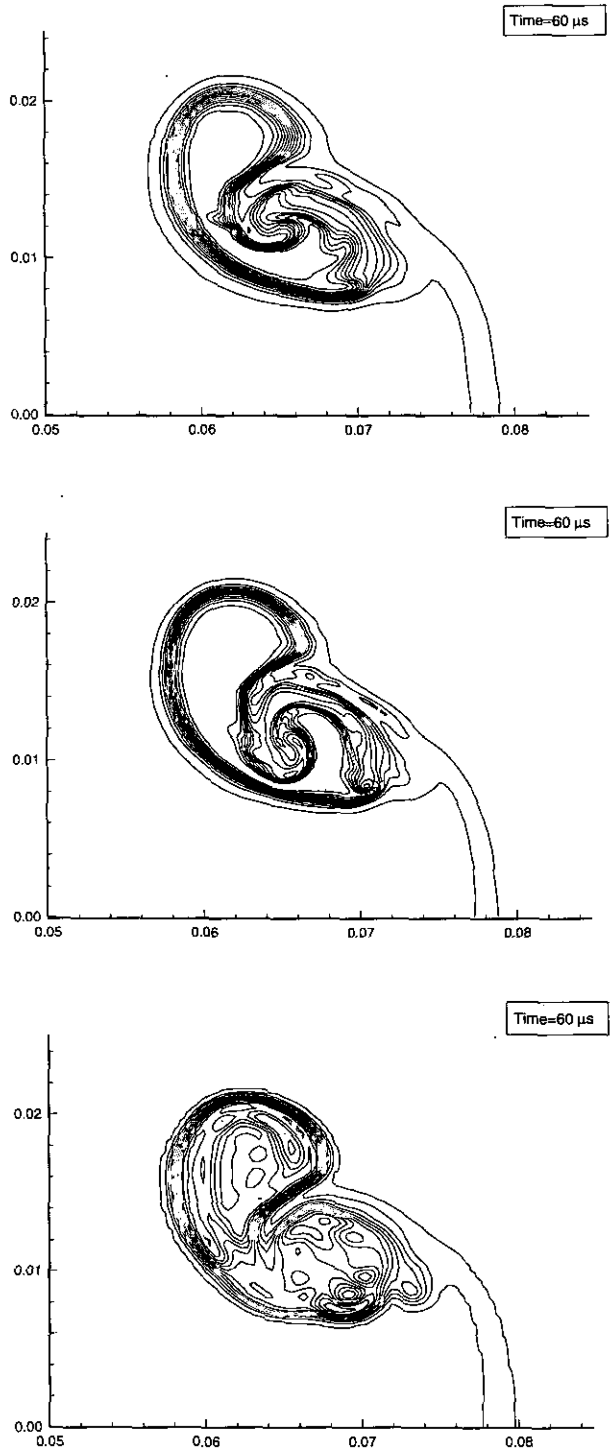


FIG. 11. Hydrogen mass fraction for the Mach 2 shock–hydrogen cylinder interaction at $t = 60$ μs. ENO fifth order with grid size 496 × 248 (top), 752 × 376 (middle), and post-processed spectral data with 512 × 512 Chebyshev collocation points (bottom). The domain size is 0 cm ≤ x ≤ 20 cm, −6.5 cm ≤ y ≤ 6.5 cm.

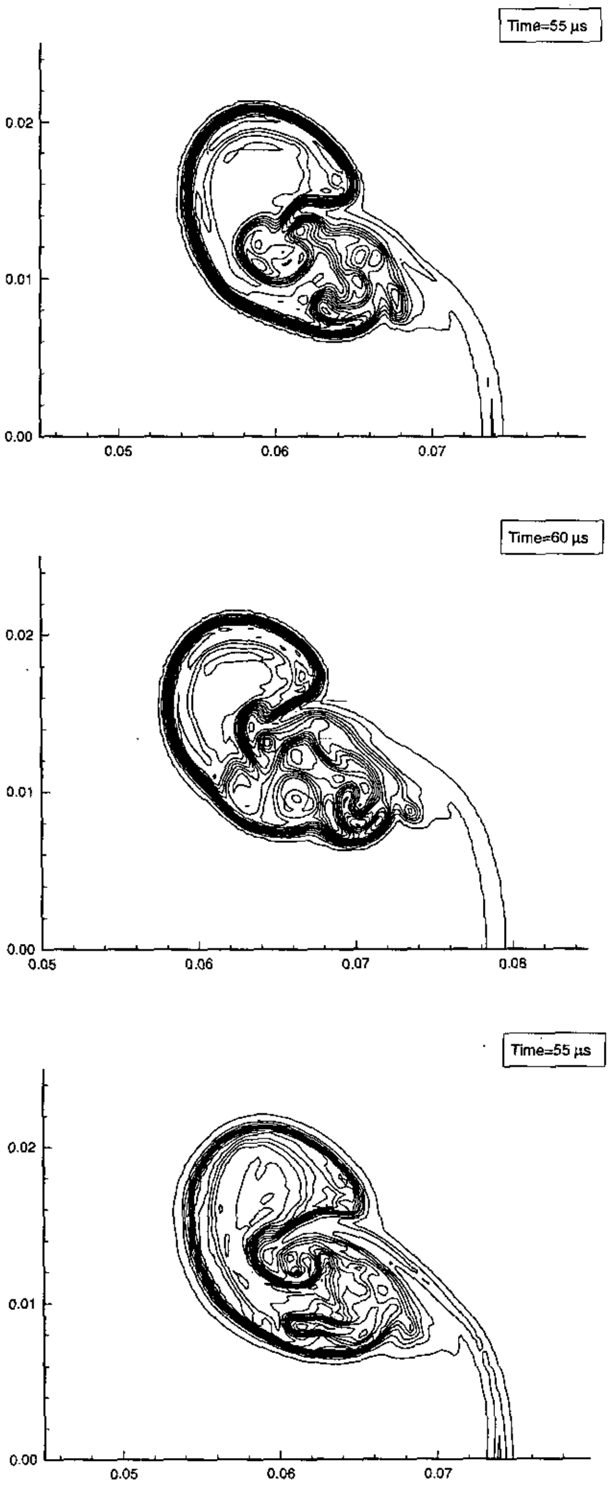


FIG. 12. Hydrogen mass fraction for the Mach 2 shock–hydrogen cylinder interaction. Grid size is 512 × 512 Chebyshev collocation points. The domain size is 0 cm ≤ x ≤ 12.5 cm, −6.5 cm ≤ y ≤ 6.5 cm. Post-processed spectra data at t = 55 µs (top) and t = 60 µs (middle). Spectra raw data with viscosity and combustion at t = 55 µs with 0 cm ≤ x ≤ 20 cm, −6.5 cm ≤ y ≤ 6.5 cm (bottom).

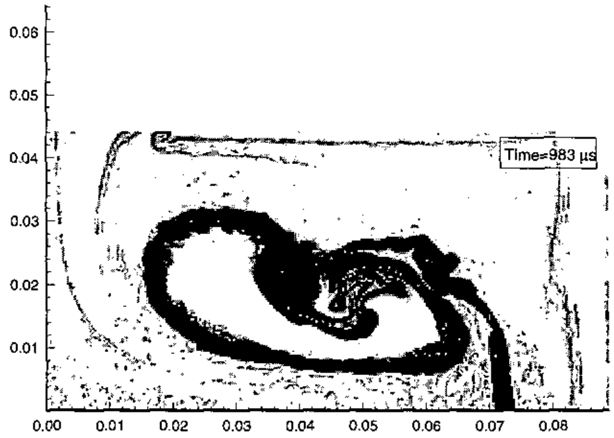


FIG. 13. Mach 1.22 shock–helium cylinder interaction at time 983 µs, using a local Lax–Friedrich flux.

high order/high resolution ENO schemes when the two are compared on the same grid. Boundary conditions are more difficult for the spectral methods however, and it may be possible to carry out ENO calculations on a small enough domain so that they are competitive. In any event, finite difference methods like ENO schemes are much easier and more efficient to parallelize, and may be expected to have an advantage as larger parallel machines become available.

7. CONCLUSION

We have applied two high order shock capturing methods to the study of a air shock–fuel cylinder jet interaction. A spectral code and a parallel ENO finite difference code are applied at a variety of resolutions and numerical orders to a

TABLE II

CPU Time Requirements for the Mach 2 Shock–Hydrogen Cylinder Problem Calculated to 135 µs

Grid ($N \times M$)	Order	No. of time steps	CPU time (s)	Nodes	Machine
240 × 120	3	1,908	1,072	32	CM-5
496 × 248	3	4,492	8,470	32	CM-5
752 × 376	3	7,534	31,486	32	CM-5
240 × 120	5	5,858	4,930	32	CM-5
496 × 248	5	12,962	37,824	32	CM-5
752 × 376	5	20,448	136,143	32	CM-5
1,472 × 736	3	15,508	18,396	512	CM-5
1,472 × 736	5	42,183[a]	77,760	512	CM-5
512 × 512	Spectral	3,650	4,562	1	C90
800 × 512	Spectral	6,900	13,800	1	C90

[a] By extrapolation from a partial computation.

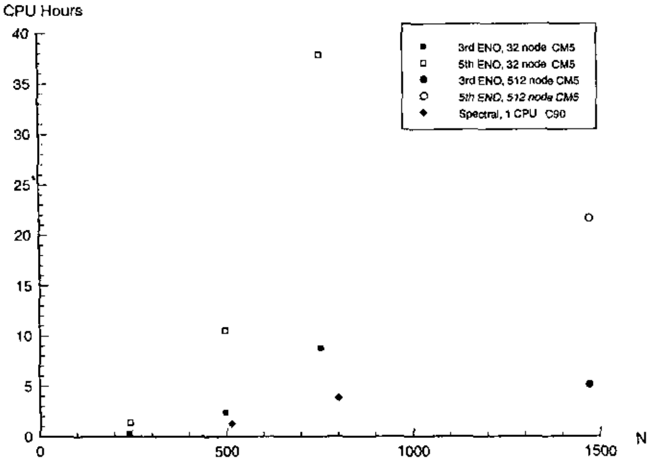


FIG. 14. CPU hour usage of each scheme for different problem sizes. N is the number of grid points in x.

Mach 2 interaction. The ENO code was validated by comparison with an experimental Mach 1.22 interaction with a helium cylinder jet. The optimized parallel ENO implementation makes practical the use of high resolutions and fifth-order spatially accurate schemes, facilitating comparison with the spectral simulation. The higher the resolution and the order of the ENO scheme, the better the comparison for long-time integration with the spectral method. Post-processing of the raw spectral data generates an accurate solution from the spectral computation.

In part II of this paper series, we present results for simulations with viscosity modeling and simple chemistry. We report quantitative results for mixing efficiencies achieved and compare with analytical models for the vortex core spacing, vortex strength, and the time scales in the problem. We discuss the effect of varying the cross-sectional shape of the cylinder, small-wavelength perturbations of its surface, and how diffuse the cylinder interface is initially on the resulting flow and mixing efficiency. We also consider the effect of multiple jets interacting with a shock.

APPENDIX A: ENO SCHEME

The ENO schemes of Osher and Shu are a class of high order finite difference schemes in conservative form. We summarize briefly here the ideas behind the discretization. Consider the model equation

$$u_t = -g(u(x))_x, \tag{9}$$

which is a one-dimensional scalar conservation law. We may discretize the spatial derivative in the following way:

$$u_t(x_j) = \frac{\hat{g}(x_{j-1/2}) - \hat{g}(x_{j+1/2})}{\Delta x}. \tag{10}$$

Here we assume a regular grid of points x_j. The grid spacing is $\Delta x = x_{j+1} - x_j$. $\hat{g}(x_{j+1/2})$ is a consistent low order approximation to the flux $g(u(x))$ evaluated at a point midway between x_j and x_{j+1}. In flux-based ENO schemes, this approximation is clearly arranged so that $(1/\Delta x)[\hat{g}(x_{j-1/2}) - \hat{g}(x_{j+1/2})]$ is actually a high order approximation to the derivative $-g(u)_x$. This may be accomplished in the following way. Let

$$P(x_{j+1/2}) = \sum_{k\leq j} g(u(x_k)). \tag{11}$$

This defines $P(x)$ on the points $x_{j+1/2}$. Given a desired order s_0 of approximation of the spatial discretization, a stencil of points is chosen $\{x_{s+1/2}, x_{s+3/2}, \ldots, x_{s+s_0+1/2}\}$ consisting of $s_0 + 1$ grid points. We may then compute a polynomial interpolant of P. The obvious way is to use a Lagrange polynomial interpolant

$$P(x) = \sum_{k=s}^{s+s_0} \prod_{\substack{l=s \\ l\neq k}}^{s+s_0} \left(\frac{x - x_{l+1/2}}{x_{k+1/2} - x_{l+1/2}}\right) P(x_{k+1/2}). \tag{12}$$

Then $\hat{g}(x_{j+1/2}) = \Delta x(\partial P(x)/\partial x)|_{x_{j+1/2}}$ will yield a high order approximation to the derivative in eq. (9):

$$\left.\frac{\partial g(u)}{\partial x}\right|_{x=x_j} = \frac{\hat{g}(x_{j-1/2}) - \hat{g}(x_{j+1/2})}{\Delta x} + O(\Delta x^{s_0}). \tag{13}$$

This can be explained in the following way: suppose a function $h(x)$ exists which is a deconvolution of g

$$g(x) = \frac{1}{\Delta x}\int_{x-\Delta x/2}^{x+\Delta x/2} h(\zeta)\, d\zeta. \tag{14}$$

Then $g'(x_j) = (1/\Delta x)[h(x_{j+1/2}) - h(x_{j-1/2})]$ exactly.

So we need only find a high order approximation to h, and this will serve for $\hat{g}$. Now

$$\frac{1}{\Delta x}\int_{-\infty}^{x_{j+1/2}} h(\zeta)\, d\zeta = \sum_{k\leq j} \frac{1}{\Delta x}\int_{x_{k-1/2}}^{x_{k+1/2}} h(\zeta)\, d\zeta = \sum_{k\leq j} g(x_j), \tag{15}$$

so by (11) $\int_{-\infty}^{x} h(\zeta)\, d\zeta = \Delta x P(x)$, and $\Delta x P'(x) = h(x)$. Therefore Δx times a high order approximation to the derivative of $P(x)$ will serve for $\hat{g}$. The solution is to interpolate P with a polynomial of order s_0. Then $\hat{g}$ will be approximate h to order $s_0 + 1$ and the difference in (10) will be accurate to order s_0.

A.1. *Choosing a Stencil*

Missing in the above discussion is a method for choosing a stencil of points $\{s_{s+1/2}, \ldots, x_{s+s_0+1/2}\}$ to be applied. In fact, the stability of the numerical scheme is determined by this choice. For example, it is known that a consistent bad choice of the stencil will yield a linearly unstable scheme. A method that has been seen to work well in practice is to construct the stencil recursively by comparing magnitudes of undivided difference coefficients.

The undivided difference coefficients $D[x_{s-1/2}, \ldots, x_{s+n+1/2}]$ are defined inductively in the following way:

$$\begin{aligned} D[x_{s-1/2}, x_{s+1/2}] &= g(x_s) \\ D[x_{s-1/2}, \ldots, x_{s+n+1/2}] &= D[x_{s+1/2}, \ldots, x_{s+n+1/2}] \\ &\quad - D[x_{s-1/2}, \ldots, x_{s+n-1/2}]. \end{aligned} \tag{16}$$

Stencil construction in an ENO scheme works in the following way: If a stencil

$$\{x_{s+1/2}, \ldots, x_{s+n+1/2}\}$$

is chosen for the scheme of order n, then either the stencil $\{x_{s-1/2}, \ldots, x_{s+n+1/2}\}$ or

$$\{x_{s+1/2}, \ldots, x_{s+n+3/2}\}$$

are possible choices for the stencil of the scheme of order $n + 1$. The choice between stencils is made depending upon the magnitude of the difference coefficients. The first stencil is chosen if its associated difference coefficient is less in magnitude that the difference coefficient associated with the second stencil:

$$|D[x_{s-1/2}, \ldots, x_{s+n+1/2}]| < |D[x_{s+1/2}, \ldots, x_{s+n+3/2}]|. \tag{17}$$

A minor modification of this algorithm has proven to work better in practice [22]. The idea is to preferentially choose stencils that are closer to being centered differences by biasing the comparison in (17) by a multiplied constant. That is the approach used in this paper. The choice of the initial stencil for the zeroth order scheme is determined by upwinding considerations and is described below.

A.2. *Upwinding*

In order to achieve stability the numerical scheme must take into account the natural propagation of waves in the problem. Doing this well in practice is a difficult exercise for a nonlinear system of equations. We begin by considering the scalar equation (9) again. In this paper, we apply a version of the Lax–Friedrich flux splitting,

$$g(u) = g^+(u) + g^-(u), \tag{18}$$

where $g^{\pm}(u) = \frac{1}{2}(g(u) + \alpha u))$, and α is a wave speed, taken as $\max|g'(u)|$ for a global Lax–Friedrich splitting, for a less diffusive flux $\alpha(x_{j+1/2}) = \max_{u_j}^{u_j+1/2}|g'(u)|$. Compromises between these two extremes are also possible, where a maximum is computed locally over some number of grid points.

$\hat{g}^+(x_{j+1/2})$ and $\hat{g}^-(x_{j+1/2})$ are calculated separately in the same manner that $\hat{g}(x_j)$ was above. The initial starting stencil is taken as $\{x_{j-1/2}, x_{j+1/2}\}$ for $\hat{g}^+$ and $\{x_{j+1/2}, x_{j+3/2}\}$ for $\hat{g}^-$. For $\hat{g}^{\pm}$, $P^{\pm}(x_j) = \sum_{k\leq j} g^{\pm}(x_k)$ is interpolated, instead of $P(x_j)$ as before. Having calculated $\hat{g}^{\pm}$, we revise our definition of $\hat{g}$,

$$\hat{g} = \hat{g}^+ + \hat{g}^-.$$

It is this quantity that is actually used in Eq. (10) to discretize Eq. (9).

A.3. *Systems*

A linear system of equations

$$\mathbf{U}_t = -A\mathbf{U}_x \tag{19}$$

may be discretized by diagonalizing the system, and then solving for each component as before. For example, if A can be written $A = L^{-1}\Lambda L$, where Λ is diagonal, then $L\mathbf{U}_t = -\Lambda L\mathbf{U}_x$. Define a new state vector $\mathbf{Y} = L\mathbf{U}$. Then in terms of this new parameterization

$$\mathbf{Y}_t = -\Lambda\mathbf{Y}_x.$$

The system is completely decoupled, and each component may be handled independently as before.

This approach may be applied locally in nonlinear systems to approximately decouple them as well. Suppose we desire a spatial discretization of the system

$$\mathbf{U}_t = -\mathbf{F}(\mathbf{U})_x = -A(\mathbf{U})\mathbf{U}_x$$

of the form

$$\mathbf{U}_t(x_j) = \frac{1}{\Delta x}(\hat{\mathbf{F}}(x_{j-1/2}) - \hat{\mathbf{F}}(x_{j+1/2})).$$

In order to compute $\hat{F}$ at each half grid point $x_{j+1/2}$, we construct a different flux splitting

$$\mathbf{F}(x) = L^{-1}(x_{j+1/2})(\mathbf{F}^+(x) + \mathbf{F}^-(x)),$$

where

$$\mathbf{F}^{\pm}(x) = L(x_{j+1/2})(\mathbf{F}(\mathbf{U}(x)) \pm \overline{\Lambda}(x_{j+1/2})\mathbf{U}(x)).$$

$L^{-1}(x)$, $\Lambda(x)$, and $L(x)$ are derived from the diagonalization of a matrix $A(x)$ as with the linear case. The question is, what matrix A to use at the half grid points $x_{j+1/2}$. Now $A(x_j) = \partial\mathbf{F}(\mathbf{U})/\partial\mathbf{U}|\mathbf{U}_{U(x_j)}$. We need some average of $A(x_j)$ and $A(x_{j+1})$ to use for $A(x_{j+1/2})$. There is more than one possible way of taking this average, but for the purposes of this paper, adequate performance is obtained by a simple average:

$$A(x_{j+1/2}) = \left.\frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}}\right|_{\frac{U(x_j)+U(x_{j+1})}{2}}.$$

$\overline{\Lambda}$ Is a diagonal matrix of maximum wave speeds in the problem. It may be taken as the global maximum of $|\Lambda_{ii}|$ where Λ is the matrix of eigenvalues of A. Or some localized maximum may be used.

A.4. *Boundary Conditions*

Two types of boundary are implemented in the current code. In both cases ghost points outside the computational domain are employed. For example, consider the boundary $x = 0$. The computational domain consists of grid points (x_j, y_k) for $j \geq 0$ and less than some maximum, and y in some bounded range as well. The ghost grid points $(x_{-1}, y_k), \cdots, (x_{-s_0-1}, y_k)$ are computed from the known data.

For reflecting boundary conditions, set $\mathbf{U}(x_{-l}, y_k) = \mathbf{U}(x_{l-1}, y_k)$ for $l = 1, \ldots, s_0$, for all elements of the state vector $\mathbf{U}$ except the y momentum, ρv. The sign of this element is inverted

Jacobians $\partial\mathbf{F}(\mathbf{U})/\partial\mathbf{U}$ and $\partial\mathbf{G}(\mathbf{U})/\partial\mathbf{U}$. The Diagonalization for $\partial\mathbf{G}(\mathbf{U})/\partial\mathbf{U}$ may be obtained from that of $\partial\mathbf{F}(\mathbf{U})/\partial\mathbf{U}$ by a suitable relabeling of variables and row and column exchanges in the equations, due to the isotropic nature of the fluxes in the problem.

With some difficulty we may obtain

$$\frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} = L^{-1}\Lambda L$$

with

$$L^{-1} = \begin{pmatrix} I - \mathbf{f} \otimes \mathbf{b}_3' & \mathbf{f} & \mathbf{f} & 0 & \mathbf{f} \\ -\mathbf{b}_3' & 1 & 1 & 0 & 1 \\ -u\mathbf{b}_3' & u - c & u & 0 & u + c \\ -v\mathbf{b}_3' & v & v & 1 & v \\ -H\mathbf{b}_3' & H - cu & H - 1/b & v & H + cu \end{pmatrix}$$

$$b_1 = \frac{1}{\rho c^2}\frac{\partial p}{\partial e}$$

$$b_{3i} = \frac{1}{\rho c^2}\frac{\partial p}{\partial f_i}$$

$$b_2 = \frac{1}{\rho c^2}\left(\frac{\partial p}{\partial e}\left(\frac{u^2+v^2}{2} - e\right) + \rho\frac{\partial p}{\partial \rho} - \sum_{i=1}^{N_s-1} f_i \frac{\partial p}{\partial f_i}\right)$$

Using the thermodynamic model presented in section 2.1, computing the partial derivatives of p is straightforward,

$$p = \left(\sum_{i=1}^{N_s-1}\frac{f_i}{M_i} + \frac{1-\sum_{i=1}^{N_s-1} f_i}{M_{N_s}}\right)\rho R T(e, \mathbf{f})$$

so

$$\frac{\partial p}{\partial f_i} = \left(\frac{1}{M_i} - \frac{1}{M_{N_s}}\right)\rho R T + \frac{p}{T}\frac{\partial T}{\partial f_i}.$$

Now

$$e(T, f_i) = \sum_{i=1}^{N_s-1} f_i\left(\int^T \frac{1}{M_i}(C_{p_i}(T) - R)\,dT + h_i^0\right) + \left(1 - \sum_{i=1}^{N_s-1} f_i\right)\int^T \frac{1}{M_{N_s}}(C_{pN_s}(T) - R)\,dT.$$

and $de = (\partial e/\partial T)\,dT + (\partial e/\partial f_i)\,df_i$, so

$$\left.\frac{\partial T}{\partial f_i}\right|_e = \frac{\partial e/\partial f_i}{\partial e/\partial T}$$

$$\frac{\partial e}{\partial f_i} = \int^T\left(\frac{C_{p_i}}{M_i} - \frac{C_{p_{N_s}}}{M_{N_s}}\right)dT + h_i^0$$

$$\frac{\partial e}{\partial T} = \overline{C}_p - \overline{R}.$$

Thus

$$\frac{\partial p}{\partial f_i} = \left(\frac{1}{M_i} - \frac{1}{M_{N_s}}\right)\rho R T - \frac{p}{T(\overline{C}_p - \overline{R})}\frac{\partial e}{\partial f_i}.$$

Calculating $\partial p/\partial e$ is somewhat easier,

$$\frac{\partial p}{\partial e} = \rho\overline{R}\frac{\partial T}{\partial e} = \rho\overline{R}\Big/\frac{\partial e}{\partial T} = \frac{\rho\overline{R}}{\overline{C}_p - \overline{R}}.$$

Simplest of all,

$$\frac{\partial p}{\partial \rho} = \frac{p}{\rho}.$$

Due to the fact that u is a repeated eigenvalue in Λ the above diagonalization is not unique. The above choice is designed to produce a matrix L that is as sparse as possible. This has the effect of reducing the number of operations carried out in the inner loop of the computational kernel, and is the most obvious way of reducing computational cost of the procedure.

ACKNOWLEDGMENTS

The first author is grateful to Professor David Gottlieb for his continuous encouragement and moral support during the course of this research. Both authors thank Dr. Phil Drummond for his insight and helpful discussions in the physical aspects of the subject, and Professor Chi-Wang Shu for many helpful discussions of the numerical aspects of this work. Computing time on the Cray YMP/C90 was provided by the U.S. Army Corps of Civil Engineering, Waterway Experiment Station, High Performance Computer Center. Computing time on the Thinking Machines CM-5 was provided by the Army High Performance Computing Research Center at the University of Minnesota and by the Northeast Parallel Architecture Center in Syracuse. The authors acknowledge the support of this research by AFOSR Grant 93-1-0090 and DARPA Grant N00014-91-J-4016. The first author also acknowledges support by NSF Grant DMS-92-11820. The second author also acknowledges support by ARO Grants DAAH04-93-G-0329 and DAAH04-94-G-0205.

REFERENCES

1. S. Abarbanel and D. Gottlieb, *Information content in spectral calculations*, in *Progress and Supercomputing in Computational Fluid Dynamics*, edited by E. Murman *et al.* Proc. of U.S.–Israel Workshop (1985), p. 345.
2. I. Bell, *SIAM J. Sci. Comp.* **15,** 127 (1994).
3. R. Bonazza, M. Brouillette, D. Goldstein, J. F. Hass, G. Winckelmans, and B. Sturtevant, *Bull. Am. Phys. Soc.* **30,** 1742 (1985).
4. M. Brouillette and B. Sturtevant, *J. Fluid Mech.* **263,** 271 (1994).
5. W. Cai, D. Gottlieb, and C. W. Shu, *Math. Comp.* **52**(186), 389 (1989).
6. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, Berlin/New York, 1987).
7. W. S. Don, *J. Comp. Phys.* **110,** 103 (1994).
8. W. S. Don and A. Solomonoff, *SIAM J. Sci. Comp.*, in press.
9. D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications* CBMS Conference Series in Applied Mathematics, Vol. 26 (SIAM, Philadelphia, 1977.
10. L. F. Henderson, *J. Fluid Mech.* **26,** 607 (1966).
11. J. F. Haas and B. Sturtevant, *J. Fluid Mech.* **181,** 41 (1987).
12. J. W. Jacobs, *J. Fluid Mech.* **234,** 629 (1992).
13. H. W. Liepmann and A. Roshko, *Elements of Gasdynamics* (Wiley, New York, 1957).
14. F. E. Marble, in *Recent Advances in the Aerospace Sciences,* edited by Corrado Casci, (Plenum, New York, 1985, p. 395.
15. F. E. Marble, E. E. Zukoski, J. W. Jacobs, G. J. Hendricks, and I. A. Waitz, AIAA Paper 90-1981 (unpublished).
16. B. J. McBride, S. Heimel, J. G. Ehlers, and S. Gordon, NASA SP-3001, 1963 (unpublished).
17. E. E. Meshkov, *Izv. Akad. Nauk. SSSR Mekh. Zhidk. Gaza* **4,** 101 (1969).
18. J. M. Picone and J. P. Boris, *J. Fluid Mech.* **189,** 23 (1988).

19. J. J. Quirk and S. Karni, ICASE report 94-75, NASA CR-194978; submitted for publication.
20. G. Rudiger and L. M. Somers, *J. Fluid Mech.* **7,** 161 (1960).
21. R. Samtaney and N. J. Zabusky, *J. Fluid Mech.* **269,** 45 (1994).
22. C. W. Shu, *J. Sci. Comp.* **5**(2), 127 (1990).
23. C. W. Shu and S. Osher, *J. Comp. Phys.* **77**(1), 439 (1988).
24. C. W. Shu and S. Osher, *J. Comp. Phys.* **83**(1), 32 (1989).
25. P. N. Swarztrauber, in *Parallel Computations*, edited by G. Rodrigue (Academic Press, New York, 1982, p. 51.
26. Y. Maday and E. Tadmor, *SIAM J. Numer. Anal.* **26,** 854 (1989).
27. E. Tadmor, *SINUM* **26,** 30 (1989).
28. E. Tadmor, Shock capturing by the spectral viscosity method, in *Proceedings of ICOSAHOM 89* (Elsevier, North-Holland, IMACS, 1989.
29. V. T. Ton, Ph.D. thesis, UCLA, 1993 (unpublished).
30. I. A. Waitz, *AIAA J.* **31**(6), 1014 (1993).
31. X. Yang, N. J. Zabusky, and I. L. Chen, *Phys. Fluids A* **2,** 892 (1990).
32. J. Yang, T. Kubota, and E. E. Zukoski, *J. Fluid Mech.* **258,** 217 (1994).
33. S. G. Zaitsev, E. V. Lazareva, V. V. Chernukha, and V. M. Belyaev, translated from *Teplofizika Vysokikh Temperature* **23,** 535 (1985).