Journal of Computational Physics **154,** 197–230 (1999)
Article ID jcph.1999.6312, available online at http://www.idealibrary.com on IDEAL®

Multiresolution Schemes for the Reactive Euler Equations[1]

Barna L. Bihari*,† and Donald Schwendeman‡

*Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, New York 12180-3590, and †Department of Computational Fluid Dynamics, Rockwell Science Center, 1049 Camino Dos Rios, Thousand Oaks, California 91360; ‡Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, New York 12180-3590

E-mail: *biharb@rpi.edu, †blb@infinity.risc.rockwell.com, ‡schwendd@rpi.edu

Received December 21, 1998; revised May 17, 1999

We present multiresolution (MR) schemes for the efficient numerical solution of the one-dimensional system of the reactive Euler equations, which has possibly stiff source terms. The original version of the method was developed by A. Harten (1995, *Comm. Pure Appl. Math.* **48**(12), 1305) for homogeneous hyperbolic conservation laws. By computing the cell average MR-representation of the solution, we obtain much information about the solution's regularity. This description of smoothness is then used to reduce the number of direct flux computations as well as the expensive high-order ENO (essentially nonoscillatory) reconstruction both of which are now performed only near discontinuities. Thereby, the numerical solution procedure becomes considerably more efficient. In the present case of the reactive Euler equations, the average efficiency factor measured by counting the number of actual flux computations ranges from about 5 to 12. This is on the same order of, and in some cases comes reasonably close to, actual speed-up factors obtained by code timings, which were between 3 to 5. The MR overhead rate was about 10% for the ENO and 36% for TVD schemes, respectively. The quality of the solution is shown to be the same as that of the finest grid. Detailed numerical and performance results are shown for up to fourth-order accuracy, for source terms ranging from moderate to extremely stiff.

Key Words: multiresolution scheme; stiff source terms; essentially nonoscillatory interpolation; conservation laws; reaction problems.

1. INTRODUCTION

Mathematical models of reactive gas flows couple the effects of non-reactive hydrodynamics with those of heat release due to chemical reactions. In the case of inviscid flow,

[1] Support for this research was given by the National Science Foundation (Grant DMS-9705051).

0021-9991/99 $30.00

the hydrodynamics is modeled by the usual Euler equations of gas dynamics. The chemical reactions may be modeled in different ways depending on the particular problem at hand and may involve many reacting species covering a wide range of reaction time scales. The simplest models involve a binary mixture of burnt and unburnt fuel. For these models, a progress variable which measures the local mass fraction of unburnt fuel is introduced in addition to a single reaction rate equation. The reaction rate equation, in turn, introduces a time scale to the problem which is often much faster than the usual hydrodynamic time scale as determined by the local sound speed rendering the equations stiff. Rapid temporal changes in the flow quantities can provoke rapid spatial variations which appear in the solution as sharp reaction fronts and/or shock waves.

Solutions of the reactive Euler equations are often computed numerically and it is a difficult task to determine these solutions accurately. The main problem is due to the reaction rate term which is typically a sensitive function of temperature. This sensitivity requires a fine spatial and temporal grid (at least locally) in order to resolve the fast reaction scales which is needed to obtain an accurate solution. Inadequate grid resolutions can lead to numerical errors which take the form of numerical oscillations and incorrect reaction front speeds, among others, and these errors have been reported in the literature for both the reactive Euler equations (e.g., see [14]) and for reduced models (see, e.g., [13, 4]).

Solution adaptive techniques provide a useful tool for the reactive Euler equations, and for stiff equations in general. In this paper, we propose a method based on finite-volume, multiresolution schemes as originally discussed by Harten in [8] for non-stiff problems. The goal is to obtain a high-order numerical solution on a fine grid everywhere. This is achieved by employing multiresolution techniques to adaptively select regions where the full ENO fluxes are computed and interpolate from coarser grids in the rest of the domain. In this way, adequate numerical resolution is provided at a reduced computational cost. For example, a decrease in the run time by a factor of 3 to 5 for the proposed method as compared to corresponding single-grid calculations is obtained for the test problems considered.

In what follows, we start by describing the governing equations, then we review the finite-volume high-order ENO scheme (Section 2). We then present two anomalies encountered by "naively" applying a well-tested finite volume scheme to the reactive Euler equations (Section 3). One is that of wrong reaction front speed, varying widly with time step, and the second is a density "spike." Neither of these problems is catastrophic in the sense of blow-up, or obvious spurious oscillations, making them all the more deceiving. The simple fix is to add more grid cells. Alternatively, one can also raise the order of accuracy, or a combination thereof. Once we are convinced that very high resolution is indeed necessary for the solution to be of any practical value to the user, we may attempt to reduce the cost. MR schemes do this by carefully choosing the regions where fluxes are actually computed. With minor adaptations, the original MR idea of [7], as generalized in [3], applies directly to the present set of equations as well (Section 4). While we expect a certain computational overhead along with a "programming overhead" when compared to the non-MR method, we hope that both are outweighed by significant run-time savings. The numerical results of Section 5 provide ample evidence that this is indeed the case.

2. THE GENERAL FRAMEWORK

2.1. *Governing Equations*

We seek an approximate solution to the one-dimensional Euler equations of gas dynamics with a fourth equation added for the progress variable λ. For completeness we include a full

nomenclature here. Cast in conservative differential form, the equations to be solved are

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = \mathbf{h}, \tag{2.1a}$$

on $(x, t) \in [a, b] \times (0, \infty)$, $[a, b] \subset \mathbf{R}$, with initial conditions

$$\mathbf{q}(x, 0) = \mathbf{q}_0(x). \tag{2.1b}$$

Here

$$\mathbf{q} = (e, \rho, \rho u, \rho\lambda)^T$$

denotes the solution vector,

$$\mathbf{f} = [(e + p)u, \rho u, p + \rho u^2, \rho u\lambda]^T$$

is the vector-valued flux function, and

$$\mathbf{h} = [0, 0, 0, \rho R]^T$$

is the source term. As usual, ρ, p, u, e stand for density, pressure, velocity, and total energy, respectively, with the ideal gas equation of state modified only for the heat release from the reaction,

$$p = (\gamma - 1)\left(e - \frac{1}{2}\rho u^2 + \beta\rho\lambda\right) \tag{2.2a}$$

where γ is the ratio of specific heats and β is the heat release. The reaction rate R is defined by

$$R = \kappa(1 - \lambda)\mathrm{e}^{-E\frac{\rho}{p}}, \tag{2.2b}$$

where κ is the rate constant and E is the activation energy.

This specific set of hyperbolic conservation laws is solved via a (by now) traditional finite volume scheme, cast in semi-discrete form,

$$(\mathbf{v}_j)_t = -\frac{1}{h}\left(\mathbf{f}_{j+\frac{1}{2}} - \mathbf{f}_{j-\frac{1}{2}}\right) + \bar{\mathbf{h}}_j = \mathbf{S}_j(\mathbf{v}), \tag{2.3}$$

where τ is the time step, and $\mathbf{f}_{j+1/2}$ is the numerical flux, a function of $2K$ variables,

$$\mathbf{f}_{j+\frac{1}{2}} = \mathbf{f}\left(\mathbf{v}^n_{j-K+1}, \ldots, \mathbf{v}^n_{j+K}\right), \tag{2.4}$$

obtained by solving a Riemann problem at the cell face between cells j and $j+1$. Such a numerical flux could come from an exact, or an approximate, but more efficient, "Riemann solver." Godunov's method is an example of the former, and Roe's linearized flux formula is an example of the latter, which is, in fact, what we utilize in all, except one, computations of this study. In the above, $\mathbf{v}^n_j$ is an approximation to the average of the exact solution $\mathbf{q}(x, t)$ in the cell $[x_{j-1/2}, x_{j+1/2}]$, $j = 0, \ldots, N-1$,

$$\mathbf{v}^n_j \approx \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{u}(x, t_n)\, dx, \tag{2.5}$$

with $t_n = n\tau$ and $h = \frac{b-a}{N}$, N being the number of (uniform) cells in the interval $[a, b]$.

2.2. *High-Order Reconstruction*

In practice, evaluation of the numerical flux (2.4) amounts to two *point values* of the solution's cell average array $\mathbf{v}$ immediately to the left and to the right of the cell face being used as inputs to the function, which in turn, generates the "Riemann flux" as output. For spatial orders of accuracy higher than one it must be ensured that these solution point values are indeed reconstructed to the desired order of accuracy—for first-order, direct substitution of the cell average values suffices. Moreover, the question of choosing a method which avoids or reduces interpolatory oscillations must be addressed. We wish to ensure as oscillation-free reconstruction as would be allowed by the order of accuracy required. So we select one that either decreases total variation (TVD) (e.g., [9]), introduces no new extrema (UNO) as in [10], or results in any newly created extrema which are of the size of truncation error (ENO) as first introduced in [11]. In this paper, we limit ourselves to orders of accuracy two through four, and therefore employ either a TVD or ENO reconstruction. The main reason for using TVD instead of second-order ENO is the mainstream nature of TVD schemes and our desire to show the utility of MR schemes as possible upgrades to existing codes that already use TVD; of course, the two are in fact very close.

The TVD limiter we use is the simplest variant of the minmod function:

$$M(a,b) = \text{minmod}(a,b) = \begin{cases} a, & \text{if } |a| \le |b|, ab > 0, \\ b, & \text{if } |a| > |b|, ab > 0, \\ 0, & \text{if } ab \le 0. \end{cases} \tag{2.6}$$

This limiter is actually applied to the characteristic fields $\mathbf{w}_{j'} = \mathbf{L}_j \mathbf{v}_{j'}$, $j' = j-1, j, j+1$, obtained by a diagonalization of the Jacobian matrix "locally frozen" at j: $\mathbf{A}_j = \mathbf{R}_j \mathbf{\Lambda}_j \mathbf{L}_j \approx \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$ by right and left eigenvector matrices $\mathbf{R}$ and $\mathbf{L}$, respectively. The point values of the characteristic variables are computed as

$$\mathbf{w}^l_{j+\frac{1}{2}} = \mathbf{w}_j + \frac{1}{2} M(\Delta_+ \mathbf{w}_j, \Delta_- \mathbf{w}_j) \tag{2.7a}$$

$$\mathbf{w}^r_{j+\frac{1}{2}} = \mathbf{w}_{j+1} - \frac{1}{2} M(\Delta_+ \mathbf{w}_{j+1}, \Delta_- \mathbf{w}_{j+1}) \tag{2.7b}$$

with the standard notation $\Delta_+ z_j = z_{j+1} - z_j$, $\Delta_- z_j = z_j - z_{j-1}$, and M given by (2.6). The reconstructed values obtained from (2.7) element by element are used to recover the solution point values by computing

$$\mathbf{v}^l_{j+\frac{1}{2}} = \mathbf{R}_j \mathbf{w}^l_{j+\frac{1}{2}}, \tag{2.8a}$$

$$\mathbf{v}^r_{j+\frac{1}{2}} = \mathbf{R}_{j+1} \mathbf{w}^r_{j+\frac{1}{2}}. \tag{2.8b}$$

Our ENO formulation, while it ensures the required high order of accuracy, is also the simplest possible RP (*r*econstruction by *p*rimitive function) algorithm, as originally described by Harten *et al.* in [11]. As detailed above for TVD, the ENO reconstruction must also be applied to the local characteristic variables and then transformed back to conservative variables in order to avoid oscillations produced by possibly colliding discontinuities (detailed justification and examples are given in [11]). In one dimension, the most straightforward ENO algorithm starts by setting up a Newton divided difference table for the primitive function of each element v_i, $i = 1, \ldots, 4$, of conservative variable $\mathbf{v}$. The entries in the difference table are denoted by $v_i[x_j, \ldots, x_{j+k}]$, $k = 1, \ldots, r$, where r is the required order

of accuracy. Each characteristic field has its own stencil, whose starting index $m_i^1(j)$ is first initialized as

$$m_i^1(j) = j. \tag{2.9a}$$

By descending on the tree of derivatives of increasing orders, it hierarchically adds a cell from the left or from the right of the current stencil to form the new, larger stencil until sufficient number of cells are gathered to yield a polynomial of the required order. The decision at each step is based on the recursive comparison,

$$m_i^{k+1} = \begin{cases} m_i^k(j) - 1, & \text{if } \left|w_i\left[x_{m_i^k(j)-1}, \ldots, x_{m_i^k(j)+k}\right]\right| \\ & \quad \le \left|w_i\left[x_{m_i^k(j)}, \ldots, x_{m_i^k(j)+k+1}\right]\right|, \\ m_i^k(j), & \text{otherwise,} \end{cases} \tag{2.9b}$$

where the difference table for "locally" characteristic variable w_i is obtained from the precomputed table for v_i by multiplication with the left matrix $\mathbf{L}_j$.

Note that only part of the difference table is traversed this way, and it is certainly possible to furnish pathological cases when the choices made in a hierarchical scheme of this sort are non-optimal. However, based on our extensive numerical experimentation as well as that of others it seems clear that the quality of numerical results is already excellent, and it is shown to be of the required order of accuracy (see [10, 11]). Also note that the stencil build-up prescribed by (2.9b) is already very expensive since every comparison there also includes a conservative-to-characteristic transformation. Once the stencil is chosen, all the required derivatives are known as well, so the final reconstruction step is an evaluation of the first derivative of a Taylor polynomial of the primitive function.

As can be seen from (2.9b), for an rth order reconstruction, there are generally r different stencils from which to choose. Near physical boundaries, however, the number of choices will decrease, and for a boundary cell there will be exactly one stencil. Depending on the physics of the problem, this may or may not be desirable, since it may result in a downwind, high-order stencil, which could potentially produce an instability. In our experience, present work included, one-sided high-order interpolation near the boundary is not an issue, except for periodic boundaries or cases where strong shocks are reflected (in the latter case ghost cells should be constructed to allow sufficient choice, as alluded to in [11]). Alternate boundary ENO treatments typically include artificial reduction of order of accuracy near the boundary or the use of ghost cells, both of which ultimately result in loss of accuracy in the maximum norm.

On the issue of reconstruction and order of accuracy, it should also be noted that evaluation of the source term $\bar{\mathbf{h}}_j$ in (2.3) is not, in general, a mere point-wise substitution of the cell average $\mathbf{v}_j$ into the source function coming from (2.1a). Derivation of the conservation form (2.3) from the PDE (2.1) requires $\bar{\mathbf{h}}_j$ to be evaluated as an integral, similar in nature to the form shown in (2.5). It so happens that in the first- and second-order cases treating the cell average as a point value, i.e., assuming $\bar{\mathbf{h}}_j = \mathbf{h}(\mathbf{v}_j)$, amounts to using the midpoint rule for the quadrature. This is still second-order accurate in space and thus preserves the overall accuracy of the TVD scheme. On the other hand, higher order schemes require quadratures of matching order and to this end we use Gaussian quadrature as it provides for minimum

number of evaluations n_r for a given order of accuracy:

$$n_r = \left\lfloor \frac{r-1}{2} \right\rfloor + 1. \tag{2.10}$$

To reconstruct point values at the appropriate quadrature locations we used central, or "as-close-to-central-as-possible," stencils. While the use of the ENO stencil, matching the one used in the convective part, seems appealing at first, numerical experimentation shows that for the purpose of the integral evaluation, the central stencil is actually slightly better: it is more accurate, as expected, and oscillations do not appear above and beyond that which the ENO treatment of the hyperbolic terms produce. The latter observation may be explained by the fact that the Gaussian quadrature points and weights are symmetric, thus fortuitous cancellations may help eliminate the effect of the Gibbs phenomenon.

2.3. *Time Stepping*

Besides ease of programming, the semi-discrete formulation (2.3) also offers an uncoupling of spatial and temporal discretizations which allows for independent accuracy choices. For the temporal update we use either a second- or fourth-order Runge–Kutta scheme,

$$\begin{aligned} \mathbf{v}_j^{n+\frac{1}{2}} &= \mathbf{v}_j^n + \frac{1}{2}\tau \mathbf{S}_j(\mathbf{v}^n), \\ \mathbf{v}_j^{n+1} &= \mathbf{v}_j^n + \tau \mathbf{S}_j\left(\mathbf{v}^{n+\frac{1}{2}}\right), \end{aligned} \tag{2.11}$$

or

$$\begin{aligned} \mathbf{v}_j^{n+\frac{1}{4}} &= \mathbf{v}_j^n + \frac{1}{2}\tau \mathbf{S}_j(\mathbf{v}^n), \\ \mathbf{v}_j^{n+\frac{2}{4}} &= \mathbf{v}_j^n + \frac{1}{2}\tau \mathbf{S}_j\left(\mathbf{v}^{n+\frac{1}{4}}\right), \\ \mathbf{v}_j^{n+\frac{3}{4}} &= \mathbf{v}_j^n + \tau \mathbf{S}_j\left(\mathbf{v}^{n+\frac{2}{4}}\right), \\ \mathbf{v}_j^{n+1} &= \frac{1}{3}\left(-\mathbf{v}_j^n + \mathbf{v}_j^{n+\frac{1}{4}} + 2\mathbf{v}_j^{n+\frac{2}{4}} + \mathbf{v}_j^{n+\frac{3}{4}} + \frac{1}{2}\tau \mathbf{S}_j\left(\mathbf{v}^{n+\frac{3}{4}}\right)\right), \end{aligned} \tag{2.12}$$

which will be referred to as RK2 and RK4, respectively.

3. INITIAL RESULTS AND MOTIVATION

Before describing the multiresolution scheme, we first present two sets of experiments with the goal of shedding some light on some of the underlying numerical issues introduced by the reaction physics. The present authors feel that the choice of a numerical scheme should always be justified by its utility for the particular problem being solved, beyond its general theoretical or numerical appeal. In this context, we aim to justify the use of both the ENO and the MR methods and thus set the stage for the method that combines these two ideas.

For both of the example cases discussed here we use reflective BC at the left boundary and non-reflective BC at the right boundary. The boundary conditions are applied at the boundary face only and not in the reconstruction scheme, i.e., we do not duplicate or mirror

cells by way of ghost cells. At a boundary cell, the only way the ENO algorithm knows about the existence of a boundary is by a limited choice of stencils there (more precisely, exactly one, one-sided stencil).

3.1. *A Stiff Test Case*

For a full definition of the initial boundary-value problem (2.1) the initial conditions at $t = 0$ remain to be specified. As suggested by Kapila [12], initial conditions (2.1b) for this case are given by

$$\begin{cases} p = 1, \\ u = 0, \\ \lambda = 0, \\ T = 1 - \kappa\omega\left|x - \frac{a+b}{2}\right|, \end{cases} \tag{3.1a}$$

where ω is a slope constant and temperature T is nondimensionalized as

$$T = \frac{\gamma}{\gamma - 1}\frac{p}{\rho}. \tag{3.1b}$$

The rate model (2.2b) was slightly transformed via the substitutions $\kappa \to \kappa e^{1/\kappa}$ and $E \to \frac{\gamma-1}{\gamma\kappa}$ to arrive at

$$R = \kappa(1 - \lambda)e^{\frac{1}{\kappa}\left(1-\frac{1}{T}\right)}, \tag{3.2}$$

which now contains only only one parameter.

When applied to the inert Euler equations, the schemes described in Section 2 give predictable results, with shocks and contact discontinuities in the correct locations, no visible oscillations, and a noticeable improvement in discontinuity sharpness when the order of accuracy is increased. In fact, the second-order TVD scheme, subsequently called TVD2, is convergent under the well-known CFL-condition of unity as shown by Harten in [9].

In the absence of source terms, the additional conservation equation in $\rho\lambda$ by itself does not induce a qualitative change in the physics: the fourth eigenvalue of the Jacobian A is u, producing a second contact discontinuity with Riemann invariants p, u, and ρ. The CFL-number is thus left unchanged. It is the source term in (2.1) that requires a closer look, because its possibly violent growth puts a limit on the time step that could be orders of magnitude smaller than that coming from a CFL-condition. Still, one expects a straightforward computation once the stiffness of the source term is translated into a time step estimate. In order to do this, one might split off from the hyperbolic system and solve the ODE in $\rho\lambda$, and likewise, solve the remaining homogeneous hyperbolic system separetely as well. Namely, the ODE is solved with "frozen" flow variables. Such a splitting is suggested and used in [13, 4] among others. If we apply the splitting for the sake of a τ estimate only, in the second-order case we arrive at two competing restrictions on the time step,

$$\tau \le \min\left(\frac{\sigma h}{\max_j(|u_j| + c_j)}, \frac{2}{\max_j(|\nu_j|)}\right), \tag{3.3}$$

where σ is the CFL-number chosen, u_j and c_j are the velocity and sound speed at the jth cell, and ν_j is a measure of the source term when the latter is put in the form of the usual ODE test equation $y' = \nu y$.

Of course, the above exercise in ODE stability analysis must include a linearization of the source term ρR, R defined by (3.2), in the variable $\rho\lambda$. This is valid since the transcendental function in (3.2) is analytic. The coefficient ν of the linear term is thus given by

$$\nu_j = \left.\frac{\partial(\rho R)}{\partial(\rho\lambda)}\right|_j = \kappa \mathrm{e}^{\frac{1}{\kappa}\left(1-\frac{1}{T}\right)}\left[\frac{\gamma\beta(1-\lambda)}{\kappa T^2} - 1\right]\Bigg|_j,$$

with T and p defined by (3.1b) and (2.2a), respectively.

The first numerical experiment we conducted turned out to be a rather severe case in terms of source stiffness. The constants κ and ω for (3.1) and (3.2) were taken to be 0.06 and 0.1, respectively. The heat release parameter β was set at $\beta = 2.8$. The domain $[a, b]$ was set by $a = -1$, $b = 1$, and initially we used $N = 128$ computational cells. The solution for small times is dominated by the reaction kinetics, which, once activated, operate at very fast time scales. Initially, the temperature is slowly rising, and correspondingly fuel is being spent everywhere at a slow rate. Then, as the temperature increase becomes more rapid, the fuel ignites at the left boundary, which initiates a reaction front moving to the right. Once formed, the front moves at a speed that is much faster than the wave speeds given by $u, u \pm c$. The estimate (3.3) shortly transitions the time step to about three orders of magnitude smaller than that allowed by a reasonable CFL number, which in this first case was taken to be a conservative 0.5. Ignore, for the moment, all curves on Figs. 1a–1d except the ones corresponding to the $N = 128$ case (double thickness solid lines). These profiles seem to indicate that the numerical solution is stable and looks reasonable, although we certainly note the glitch in the density profile (Fig. 1b).

Short of an exact solution, first time experiments of this sort should be repeated on several grids of different densities, to determine adequacy of resolution and ultimately convergence. The results at $t = 0.305$ of six additional selected runs, part of this initial study, are depicted in Figs. 1a–1d. At first, the dramatic difference in the reaction front locations between the seven values of N may come as a surprise. In addition, the yet unexplained density "spike" mentioned earlier does not appreciably decrease in magnitude with a substantial increase in resolution from $N = 64$ to $N = 256$. At this point, it is difficult to say what the correct solution *should* be, whether it converges as either or both of τ and $h \to 0$, and what the trend is with respect to the spike which spreads across 2 or 3 cells. For the latter, it is clear that the sudden increase in density is unphysical and is ruled out as a possibly correct behavior.

Wrong reaction front locations have been reported by [13, 4] and others. In their systematic study of a simplified scalar model problem with polynomial source term, LeVeque and Yee [13] come to the conclusion that a condition of obtaining the right front speed is that the product $h\nu$ must be less than unity. In our case, the maximum attained value of ν over all its arguments is more than 25,000. To obey the condition suggested in [13], we would need about 12,500 cells! Needless to say, this is an extremely severe limitation on the spatial mesh size. When applied, it actually renders the second term in (3.3) unnecessary, since the time step based on the CFL-limit would already be smaller than that dictated by the source term.

As we successively refine the grid by doubling the number of cells, we find that beyond a certain grid spacing—which in this example turned out to be $N = 256$—the front location

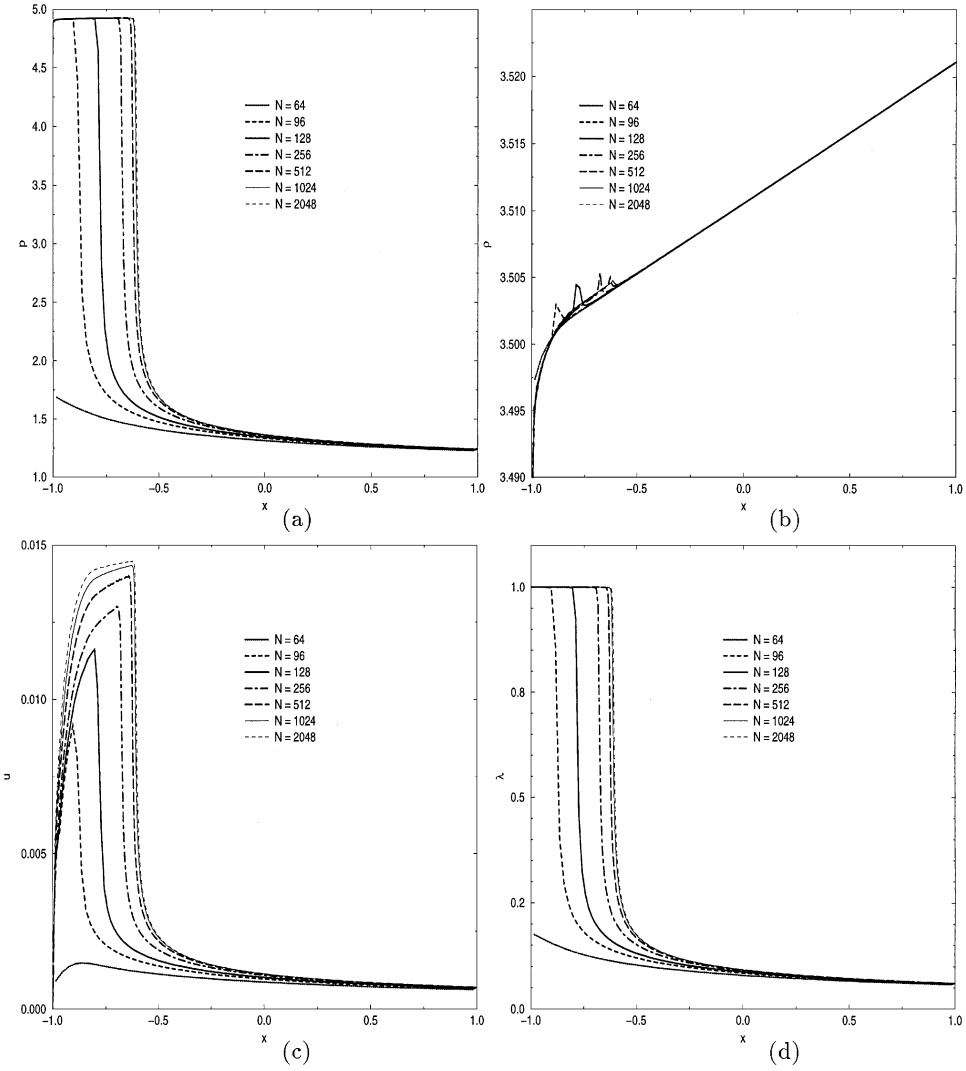

FIG. 1. Primitive variables at $t = 0.305$, using $N = 64, 96, 128, 256, 512, 1024, 2048$; (a) pressure, (b) density, (c) velocity, (d) progress variable.

indeed starts to converge, and simultaneously the density spike diminishes in magnitude (Fig. 1). However, to reach a converged state for the front and to completely eliminate the glitch, we had to refine to $N = 2048$, at which point we accepted the quality of the solution (see Fig. 2, a blow-up of Fig. 1b).

In order to gain a better understanding of the underlying phenomena, we continued to experiment with the same TVD2 scheme, but now holding the spatial resolution constant and halving the time step instead. The glitch in the density profile remained about the same size, but the front location became more accurate (Fig. 3). In addition, and as a validation of our ENO scheme, we tried running using the same setup (same size of space and temporal mesh, with RK4 (2.12) for the time update), but changing the spatial reconstruction method from TVD2 to ENO3 and ENO4. The results, focused on the critical region, are shown in Fig. 4, and they clearly show a remarkable improvement, due to high spatial accuracy, in reducing the density spike. We conclude that the lag in the reaction front is due to temporal

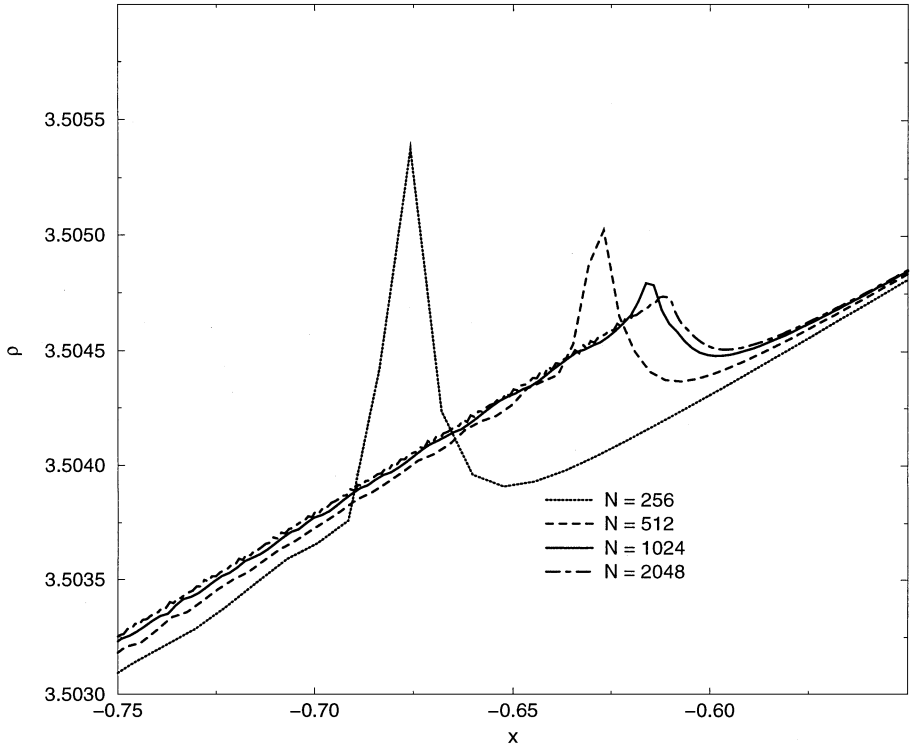

FIG. 2. Close-up of Fig. 1b. Density at $t = 0.305$, using $N = 256, 512, 1024, 2048$.

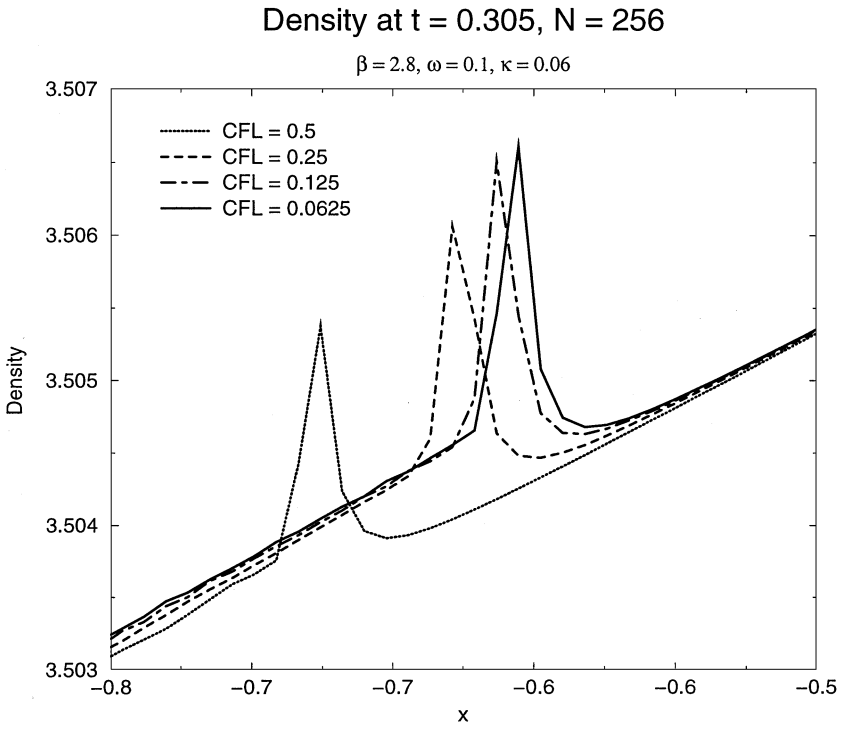

FIG. 3. Close-up of density plot at $t = 0.305$, using $N = 256$ and $\sigma = 0.5, 0.25, 0.125, 0.0625$.

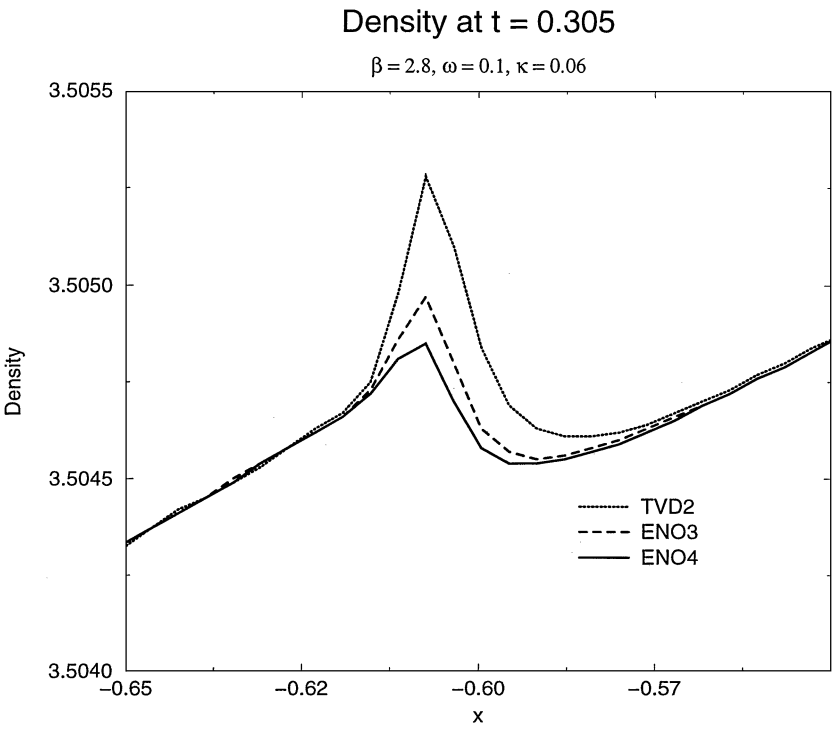

FIG. 4. Close-up of density plot at $t = 0.305$, using $N = 512$ and TVD2, ENO3, and ENO4 reconstruction.

errors, and the density spike is caused by spatial errors. Both can be remedied either by adding spatial/temporal mesh points or by increasing the respective order of accuracy.

There are many unanswered numerical issues worth exploring within the context of stiff source terms, both in terms of stability and accuracy, but which are outside of the scope of this paper. The goal of this example was merely to expose the crucial nature of grid resolution, resembling that of mesh Reynolds number stability limit on spatial step independent of time step, in the computation of viscous flows. Once the necessity to compute (at least locally) with extremely fine grids is established, one attempts to render such computations as efficient as possible, by adapting the mesh. In Section 4 we propose what be believe is a potential alternative to traditional adaptive grid methods, and which has been used successfully for scalar 1-D hyperbolic (see [8, 6, 2]) and viscous conservation laws (see [1]), scalar 2-D hyperbolic conservation laws (see [3]), and 1-D hyperbolic systems (see [5, 6, 8]).

3.2. *A Test Case for ENO*

While ENO schemes have been very successful in terms of the quality of the solution they produce, their computational and programming cost has been a discouraging factor, especially for multi-dimensional problems. Whenever used, however, ENO schemes have been successful particularly in capturing solutions with smooth, non-constant variations between discontinuities, where the high order of accuracy indeed made a great difference. In order to illustrate such a high pay-off scenario, we borrow a problem from [14] and use it to test our scheme. The governing equations remain the same, with the original rate model (2.2b) used with $\kappa = 42$ and $E = 10$, while the heat release β of (2.2a) is changed to 50.

The initial conditions (3.1) are taken to be

$$\begin{cases} p = 1, \\ u = 0, \\ \lambda = 0, \\ \rho = \frac{1}{1+3e^{-(6(x-a))^2}} \end{cases} \tag{3.4}$$

with the boundary conditions kept the same.

In preparation for the multiresolution scheme to be presented next, we benchmarked the third-order version of our ENO scheme (ENO3) against TVD2 on two different grids which were a factor of 10 apart in mesh density. We compare TVD2 and ENO3, both with $N = 200$, to TVD2 with $N = 2048$, the latter being considered as the most accurate of the three solutions.

While the governing equations are essentially the same as that of the previous example, and the initial conditions are similar as well, this problem exhibits a much different, in many ways more benign, behavior. A gasdynamic shock actually has time to form here, with a smeared reaction front following behind. The energy released from the chemical reaction continuously increases the pressure behind the shock, creating a smooth hump, which eventually catches up with the shock to form a detonation wave. We have stopped the calculation at $t = 0.5$, where the flow features are very complex and include smooth and discontinuous elements. Accurate modeling of the reaction front and shock location and its strength is critical and is a challenging test for any scheme.

The results of Fig. 5 show a dramatic improvement in approximating the "super-fine" TVD2 solution when TVD2 is replaced by ENO3. In particular, we note the large decrease in the size of the humps in all primitive variables and the large change in the reaction front location when the reconstruction accuracy is increased by one order. The shock location is also more accurate with ENO3, although not as pronounced as the other flow features. The latter is due to the fact that the temporal accuracy of ENO3 is also higher than that of TVD2, but since this problem is not stiff, the temporal error does not play as big of a role in approximating the shock speed as it did in approximating the front speed in Subsection 3.1. The large discrepancy in the smooth parts of the flow field are clearly due to the difference in spatial accuracy.

While our ENO scheme is truly of arbitrarily high order of accuracy, in our previous and current experimentations we found that, with some exceptions, the highest gain in overall accuracy is achieved by the step from second to third order. This was also true in the above test problem, which will be revisited below and solved via a multiresolution ENO3 scheme. On the other hand, in [14], Xu *et al*. prefer and successfully use a fifth-order ENO scheme.

4. THE MULTIRESOLUTION SCHEME

We now turn to present a variant of Harten's multiresolution scheme [8]. In our description of the method, we draw heavily on earlier work of Harten, as well as our own from [1–3]. The method that we refer to as *multiresolution scheme* was originally inspired by the strong interest in wavelets seen in recent years (see [7]). In fact, both in terms of its origin and its hierarchical nature, the types of multiresolution analysis first presented by Harten resemble wavelets and are indeed generalized (bi-orthogonal) forms of wavelets. As such multiscale decompositions of a function are very rich in regularity information, they proved to be

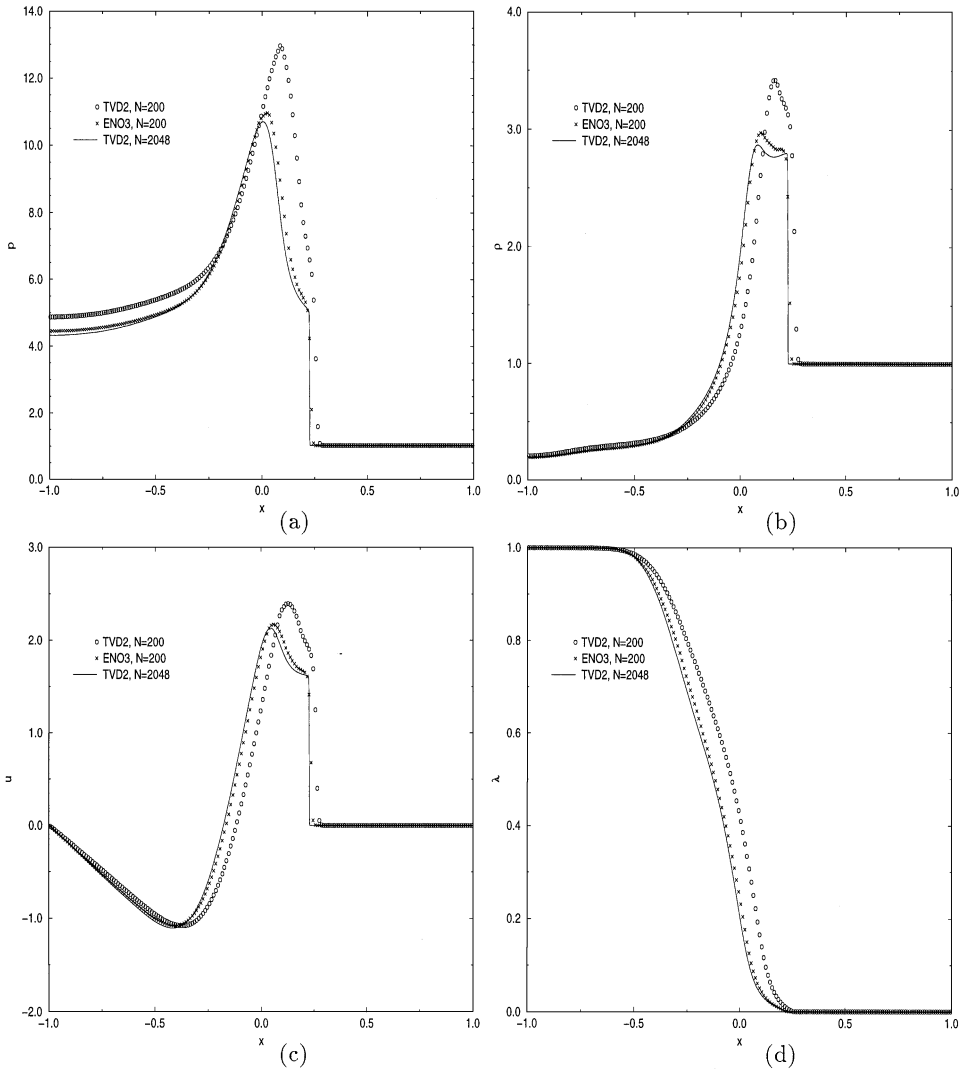

FIG. 5. Primitive variables at $t = 0.5$, using TVD2 with $N = 200$, ENO3 with $N = 200$, and TVD2 with $N = 2048$; (a) pressure, (b) density, (c) velocity, (d) progress variable.

useful tools in adapting finite volume schemes to employ different reconstruction and flux computation techniques at different locations. Multiresolution for cell averages is the natural decomposition to use when the solution is described by its cell averages and conservation is essential, yet it is but one example of the interpolatory MR of [7].

4.1. *Grid Hiearchy*

Given a numerical solution at time level n by its cell average array $\mathbf{v}^n$ over a grid defined by

$$G^0 = \left\{x^0_{j+\frac{1}{2}}\right\}_{j=0}^{N_0} = \left\{x_{j+\frac{1}{2}}\right\}_{j=0}^{N}$$

we construct a set of nested, successively coarser grids by agglomeration of fine grid cells into larger grid cells until we reach a state of maximal coarseness. In multiple dimensions

and for unstructured grids, this task by itself can become a challenging one. For our purposes, however, we coarsify each grid level by simply discarding every other cell face to successively arrive at

$$G^k = \left\{x^k_{j+\frac{1}{2}}\right\}_{j=0}^{N_k} = \left\{x^{k-1}_{j+\frac{1}{2}}\right\}_{j=0,\, j\,\text{even}}^{N_{k-1}} \tag{4.1}$$

under the assumption that the number of cells N_k on each level k is even; in all of our numerical experiments $N = N_0$ was a power of 2. The grid spacing on each level is then

$$h_k = \frac{b-a}{N_k}$$

with $N_k = N/2^k$. Although the method can be generalized to handle non-uniform grids, an arbitrary number of agglomerated cells and non-uniform agglomeration, 1-D applications thus far have not necessitated such extensions.

4.2. *Encoding*

With the grids defined above, the cell average representation of the solution on each coarser level becomes trivial: because of the additivity of the integrals and (2.5), the approximate solution on level k is represented by

$$\mathbf{v}^n_{j,k} \approx \frac{1}{h_k} \int_{x^k_{j-\frac{1}{2}}}^{x^k_{j+\frac{1}{2}}} \mathbf{u}(x, t_n)\, dx, \tag{4.2}$$

and therefore, given the approximate solution on the finest level, each coarser one will be a simple average of its fine grid counterparts. A representation *equivalent* to the fine grid solution can be arrived at by storing, in addition to the above coarse grid cell averages, the difference in information between the original fine grid solution and the one reconstructed from values on the next coarser grid. The latter is interchangibly referred to as "multiresolution coefficients," "error," or "regularity coefficients." As shown in [8] for 1-D, and in [3] for 2-D problems, these elements are proportional to gradients, or in their absence, jumps in gradients of the same order as the MR polynomial reconstruction, and that is the source of their utility in estimating regularity. This reconstruction is typically a central interpolation of the solution's primitive function, except for boundaries, where we again opt for one-sided, same-order interpolation, with information always coming from the interior of the domain. The entire process of obtaining an MR analysis

$$\mathbf{v}_M = \mathbf{M}\mathbf{v} = (\mathbf{d}^1, \mathbf{d}^2, \ldots, \mathbf{d}^L; \mathbf{v}^L)^T \tag{4.3}$$

is called the *encoding*. In (4.3), M is a matrix (see, e.g., [7, 5]), but we use it here only to symbolize the procedure; the $\mathbf{d}$'s are the MR-coefficients and L is the number of grid levels used. When each $\mathbf{v}^k_j$ is a vector, as in the case of the equations of gas dynamics, (4.3) is interpreted component-wise. For all of the numerical examples presented below, we used the scalar algorithm presented in [2] for each conservative variable, with a slight modification to account for boundaries. At each grid level k, $k = 1, 2, \ldots, L$, we first compute the next

coarser grid representation of the solution

$$\mathbf{v}_j^k = \frac{1}{2}\left(\mathbf{v}_{2j}^{k-1} + \mathbf{v}_{2j+1}^{k-1}\right), \tag{4.4a}$$

and then the predicted values from these coarse values, along with the prediction errors:

$$\begin{aligned} \mathbf{v}_{pr} &= \sum_{l=1-s}^{s-1} \gamma_{j,l}\mathbf{v}_{j+l}^k \\ \mathbf{d}_j^k &= \mathbf{v}_{2j}^{k-1} - \mathbf{v}_{pr}, \end{aligned} \tag{4.4b}$$

where j ranges from 0 to $N_k - 1$. In (4.4b), $\bar{r} = 2s - 1$ is the order of accuracy of the interpolation for obtaining $\mathbf{v}_{pr}$, the predicted value of $\mathbf{v}$, and the $\gamma_{j,l}$ are the coefficients which, for most of the domain, are based on a central stencil. Near the boundaries they come from stencils shifted so they do not cross the boundary.

4.3. *Decoding*

The reverse process, the *decoding*, conveniently denoted by $\mathbf{M}^{-1}$, converts the MR representation into the cell average array on the finest level. Starting on the coarsest level, it works its way up to the finest one by reconstructing values on each consecutive finer level via the same interpolation as used by the encoding, and adding the corresponding error stored in the $\mathbf{d}$'s. That is, for each k, $k = L, L-1, \ldots, 1$, we compute

$$\begin{aligned} \mathbf{v}_{pr} &= \sum_{l=1-s}^{s-1} \gamma_{j,l}\mathbf{v}_{j+l}^k \\ \mathbf{v}_{2j}^{k-1} &= \mathbf{v}_{pr} + \mathbf{d}_j^k \\ \mathbf{v}_{2j+1}^{k-1} &= 2\mathbf{v}_j^k - \mathbf{v}_{2j}^{k-1}. \end{aligned} \tag{4.5}$$

4.4. *Truncation*

The encoding/decoding modules can be applied to any scalar or vector data in cell average form and can in fact be used to process data that are much more general in nature than those originating from conservation laws. The size of the MR representation can be reduced by eliminating the coefficients which are sufficiently small, thus obtaining a "data compression" of the input. This procedure is termed *truncation* but is sometimes also referred to as thresholding. In absence of truncation, an encoding piped into a decoding yields a fine grid data set identical to the original one, otherwise there will be a difference; but this difference has been shown to be bounded (see [8, 3]). Moreover, the bound can be controlled by changing the tolerance parameter of the truncation. We use a truncation-like process to check for regularity, where a comparison against the tolerance value is used to set or clear elements of a flag array. These cell-by-cell flags indicate whether the particular cell is in the "smooth" or "non-smooth" (high-gradient) region, and therefore provide a switch for the algorithm to employ different treatments in each. Customized for nonlinear hyperbolic

conservation laws with discontinuities propagating with finite speeds, the procedure reads

$$\begin{aligned}
&\textbf{if } \left(\left|\mathbf{d}_j^k\right| \le \epsilon_k\right) \textbf{ then } \mathbf{d}_j^k = 0 \\
&\textbf{else} \\
&\qquad i_{j-1}^k, i_j^k, i_{j+1}^k = 1 \\
&\qquad \textbf{if } \left(\left|\mathbf{d}_j^k\right| \ge 2^{\bar{r}}\epsilon_k \textbf{ and } k > 1\right) \textbf{ then } i_{2j}^{k-1}, i_{2j+1}^{k-1} = 1.
\end{aligned} \tag{4.6}$$

Here it is assumed that initially $i_j^k = 0$ and $\epsilon_0 = \epsilon$, and that $\epsilon_k = \frac{1}{2}\epsilon_{k-1}$.

In the comparison of $\mathbf{d}_j^k$ to the tolerance ϵ, both of which are vector quantities, we must be careful about the potentially different scales present among their elements. We took the conservative approach where we check each element of $\mathbf{d}_j^k$ separately against an appropriately scaled ϵ, and if *any* of the comparisons fail, we flag the cell. One can use the l_∞-norm of the regularity coefficients (as is done in [8, 5]), or any other norm, and compare against a scalar ϵ. Clearly, the choice of ϵ and the type of comparison is crucial when it comes to accuracy and performance of the MR-scheme. The tolerance should be on the order of the truncation error on the fine level, because the goal is to provide a solution which is of the same quality as that on the finest grid. As for the comparison, our "exclusive" checking of each variable separately seems to be the safest, but it may not be necessary everywhere. For the purposes of this paper, the guarantee that the algorithm does not "miss" any irregularities was more important than marginal improvements in efficiency. For a mature code, however—just as for most adaptive grid codes—the user should be able to supply a sensitivity variable chosen from a menu of available ones, whose behavior is then used in the decision on refinement.

4.5. *Flux Computation*

Once the flag array i is set, we are ready to take advantage of the regularity information by avoiding flux computations in smooth regions. Flux computations start on the coarsest level L:

$$\mathbf{f}_{j-\frac{1}{2}}^L = \mathbf{f}(\mathbf{v}^0). \tag{4.7a}$$

We then proceed to compute on each finer grid level by looking at the i_j^k's. Unlike in [8, 5, 1, 2], where the fluxes were interpolated *as point values* in smooth regions, our current scheme skips flux computations there altogether, and instead it interpolates the right hand side (denoted by **S** in (2.3) and subsequently abbreviated as RHS) *as cell averages*. It can easily be shown that in 1-D these two interpolations are equivalent. However, the latter offers a much greater generality in multiple dimensions (see [3]), as well as potential advantages when source terms are added. The algorithm to pre-compute the fluxes now follows (for $k = L, L-1, \ldots, 1$):

$$\begin{aligned}
&\mathbf{f}_{2j-\frac{1}{2}}^{k-1} = \mathbf{f}_{j-\frac{1}{2}}^k \\
&\textbf{if } i_j^k = 1 \textbf{ then } \mathbf{f}_{2j+\frac{1}{2}}^{k-1} = \mathbf{f}(\mathbf{v}^0).
\end{aligned} \tag{4.7b}$$

Note in (4.7b) the "copy" operation from coarse to fine at every other cell face, taking advantage of fact that these faces coincide due to the particular agglomeration scheme. The

flux $\mathbf{f}$ is computed according to (2.4), and its argument is the numerical solution $\mathbf{v}^0$ on the *finest* level, providing a consistency between flux computations on different levels that is ultimately necessary for stability. The boundaries are already taken care of in (4.7a) on the coarsest grid, where the indexing runs across $N_L + 1$ cell faces.

4.6. *Right Hand Side Computation*

Once the fluxes are available where required, computation of the RHS starts, again on the coarsest level, using the precomputed coarse-grid fluxes and adding the source term:

$$\mathbf{S}_j^L = -\frac{1}{h_L}\left(\mathbf{f}^L_{j+\frac{1}{2}} - \mathbf{f}^L_{j-\frac{1}{2}}\right) + \mathbf{h}(\mathbf{v}^0). \tag{4.8a}$$

With the coarse grid RHS in place, we successively compute each finer grid RHS by using either the fluxes or interpolating from coarser levels:

$$\begin{aligned}
&\textbf{if } i_j^k = 1 \textbf{ then}\\
&\qquad \mathbf{S}^{k-1}_{2j} = -\tfrac{1}{h_{k-1}}\left(\mathbf{f}^{k-1}_{2j+\frac{1}{2}} - \mathbf{f}^{k-1}_{2j-\frac{1}{2}}\right) + \mathbf{h}(\mathbf{v}^0)\\
&\qquad \mathbf{S}^{k-1}_{2j+1} = -\tfrac{1}{h_{k-1}}\left(\mathbf{f}^{k-1}_{2j+\frac{3}{2}} - \mathbf{f}^{k-1}_{2j+\frac{1}{2}}\right) + \mathbf{h}(\mathbf{v}^0)\\
&\textbf{else}\\
&\qquad \mathbf{S}^{k-1}_{2j} = \textstyle\sum_{l=1-s}^{s-1} \gamma_{j,l}\mathbf{S}^k_{j+l}\\
&\qquad \mathbf{S}^{k-1}_{2j+1} = 2S^k_j - \mathbf{S}^{k-1}_{2j}.
\end{aligned} \tag{4.8b}$$

The interpolation of the RHS in (4.8b) is done analogously to (4.5), with the assumption that in smooth regions the error $\mathbf{d}$ corresponding to $\mathbf{S}$ is small so it would be truncated. Since the interpolation is done in pairs, for one of them an additional small savings can be realized by using the fact that the cell average on the coarse level is the arithmetic average of its two "children" on the next finer level.

The algorithm (4.8a–4.8b) is general enough to handle any type of source term, provided, that the bundling of $\mathbf{h}$ into the interpolation of $\mathbf{S}$ is accurate wherever it is performed. In other words, we must make sure that the "smooth regions" obtained from algorithm (4.6) will also be smooth when it comes to interpolation of the source term, otherwise error estimates of the sort given in [8] will not hold. To ensure this, we must modify the truncation procedure (4.6) to identify the smooth regions as those where *both* the fluxes *and* the source terms contained in $\mathbf{S}$ can be interpolated. This may, and in the case of the stiff case presented in Subsection 3.1 will, considerably increase the number of cells requiring exact computations, since when the source term is orders of magnitude larger than the flux value, the errors introduced in the interpolation formula used in (4.8b) will be dominated by the former and will therefore contaminate the entire RHS. In our case the source term is actually scalar and is much more inexpensive to compute than the fluxes, so we chose to compute it everywhere on the finest grid and to add it to the RHS *after* (4.8) is executed *without* source terms. It should be emphasized, however, that for non-stiff cases, and especially when source term evaluations are expensive, (4.8) should be used as shown, instead of performing multiresolution analysis on the source terms separately (as in [5]). We will show one example of such a case below.

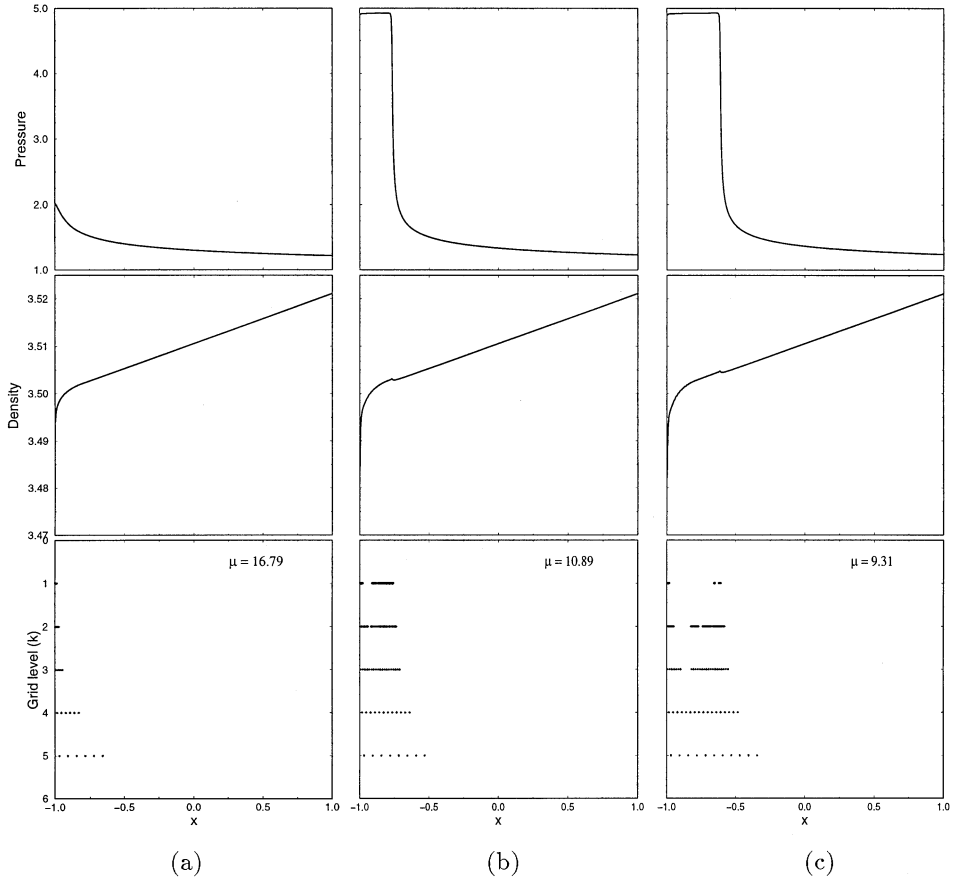

FIG. 6. TVD2 solution and MR diagram with IC (3.1) at (a) $t = 0.3$, (b) $t = 0.303$, (c) $t = 0.305$, (d) $t = 0.31$, (e) $t = 0.315$, and (f) $t = 0.32$.

5. NUMERICAL EXPERIMENTS

We present this collection of numerical results with the goal of demonstrating the performance of the multiresolution method for this particular class of problems. We also hope to illuminate some of the issues involved in using the MR scheme for systems of hyperbolic conservations laws with source terms.

In gauging the performance of an MR scheme, we typically use the following criteria, in decreasing order of importance:

(i) Quality. The MR solution should be of the same quality as that on the finest grid.

(ii) Preciseness. The MR scheme should be able to clearly identify the regions where real flux computations are needed (i.e., regions of refinement).

(iii) Speed-up. There should be a measurable speed-up in terms of actual code timings.

Criteria (ii) and (iii) are obviously related, but in the case of scalar equations, for example, where fluxes are inexpensive to compute, only the former criterion is meaningful. Item (ii) depends only on the MR algorithm itself, whereas (iii) depends on the class of the problem, programming styles, and computer architecture, yet it is also a rough measure of the overhead introduced by the MR modules.

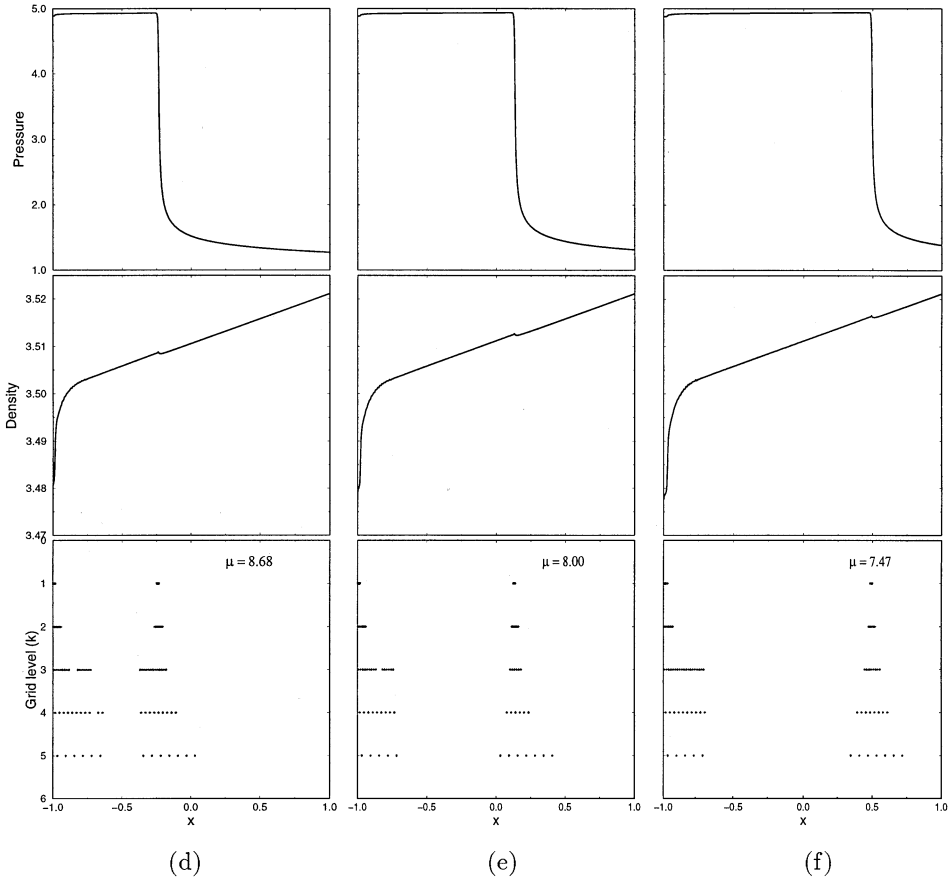

FIG. 6—*Continued*

The "quality" of the MR solution is always compared to the corresponding non-MR one, where it is implicitly assumed that the underlying finite volume scheme is basically correct in approximating the true solution. In proposing the MR method for various problems, we claim that the quality is not compromised by the adaptive speedup. The errors are thus defined as

$$e_\infty^n = \frac{1}{4}\left\langle \max_{0\le j\le N_0-1} \left|\mathbf{v}_j^{0,n} - \mathbf{w}_j^n\right|, \mathbf{b}\right\rangle, \tag{5.1a}$$

$$e_p = \frac{1}{4}\left\langle \left[\frac{1}{N_0}\sum_{j=0}^{N_0-1} \left|\mathbf{v}_j^{0,n} - \mathbf{w}_j^n\right|^p\right]^{\frac{1}{p}}, \mathbf{b}\right\rangle, \qquad p = 1, 2, \tag{5.1b}$$

where $\langle ., . \rangle$ denotes the dot product, and the absolute value is interpreted element-wise. In (5.1), $\mathbf{w}_j^n$ is the non-MR, fine grid cell average value at cell j and time step n and $\mathbf{b} = \{b_i\}_{i=1}^4$ is a scale vector to ensure that the possibly large scale differences between the conservative variables $\mathbf{v} = \{v_i\}_{i=1}^4$ are taken into account. The b_i's are the inverse of the solution's range at each time step and are computed by

$$b_i = \frac{1}{\max_{0\le j\le N_0-1}\left(v_j^n\right)_i - \min_{0\le j\le N_0-1}\left(v_j^n\right)_i} \tag{5.2}$$

so that each conservative variable is projected back into the interval [0, 1]. The errors e_p, $p = 1, 2, \infty$, become scalar quantities and are but one measure of deviation from $\mathbf{w}$. We chose these definitions to be backward compatible with the definitions used in [1–3, 6, 8], to which (5.1) would simplify when applied to those scalar problems.

The measure of "preciseness," originally called the *efficiency factor* μ in [8], can be defined as the ratio of the number of fine grid flux computations to that of actual flux computations performed by the MR scheme,

$$\mu = \frac{N_0}{N_L + |D_n|}, \tag{5.3}$$

where D_n is the set of all flags i_j^k, set or cleared in (4.6), which are 1. In (5.3) we have taken into account the fact that fluxes on the coarsest grid are computed everywhere, according to (4.7a). The definition (5.3) of the efficiency works only in one dimension, where μ also equals the *data compression* of the solution at time step n. In general, the efficiency is obtained by actually counting the calls to the flux routine (see [3]).

Of course, both the quality and efficiency will strongly depend on the tolerance $\epsilon = \{\epsilon_i\}_{i=1}^4$ used in the truncation process (4.6). It is a vector quantity, which is also appropriately scaled by the same parameter $\mathbf{b}$ defined in (5.2),

$$\epsilon_i = b_i \epsilon_r, \tag{5.4}$$

where ϵ_r is a reference quantity, indicating the maximum deviation in the l_1 norm from a solution ranging over [0, 1], and introduced by truncation in one time step. We have used values which are of the same order as those of our earlier works on scalar problems [1–3]. Although they seem to work well enough, as shown below, there is still much theoretical work to be done on providing a quantitative formula on what this tolerance value should be, as well as on understanding the accumulation of errors in time as a function of ϵ. In our experimentations we found that very small tolerance values produce low efficiency factors, but so do large tolerances too, since they first allow oscillations to develop which will, in turn, grow and propagate, thus eventually degrading the data compression. It appears that an "optimum" ϵ_r exists in between. In principle, one may require that ϵ_r be on the order of the global l_1 error from the *exact solution*. Once a "good" tolerance ϵ_g is found for a grid resolution N_g that is fine enough to solve the problem accurately, for all other grids N_f (on the finest level) the following formula holds,

$$\epsilon_f = \epsilon_g \left(\frac{N_g}{N_f} \right)^r,$$

where r is the spatial order of accuracy. This recipe should be used in order of accuracy or grid refinement studies.

For the "speed-up" we measure the *actual* run time that a case took with and without MR, from start to finish, including input, initialization, time integration, output, etc. The ratio ζ of the two numbers then tells us how many times the MR scheme is faster:

$$\zeta = \frac{\text{run time without MR}}{\text{run time with MR}}. \tag{5.5}$$

This is the time savings a user would in fact see and is therefore a quite realistic measure of the efficiency gain that can be promised. Theoretically speaking, it may have been a

more accurate approach to include timing routines between time steps and thus eliminate the constant overhead associated with each run. To still provide some measure of the latter, we shall also include profiling results with percentage breakdowns of major code segments. We finally wish to mention that code optimization was minimal, both in terms of compiler flags and special programming tricks to boost efficiency, where we usually opted for clarity instead. The runs were performed on two architectures, an SGI INDY and a SUN Ultrasparc; actual numbers shown will correspond to the former, but it was encouraging to see that speed-up ratios for the SUN were within 10% of those on the SGI, partially substantiating our claim that speed-ups quoted are machine independent.

5.1. *The Very High Reaction Rate Example*

We now return to the example of Subsection 3.1, where reaction model (3.2) with $\kappa = 0.06$ was used together with IC (3.1). In terms of the more conventional model (2.2b) this corresponds to the activation energy of about 17 and a reaction rate of over 10^6, causing an extremely fast reaction time scale to emerge and then dominate the gas dynamics. When

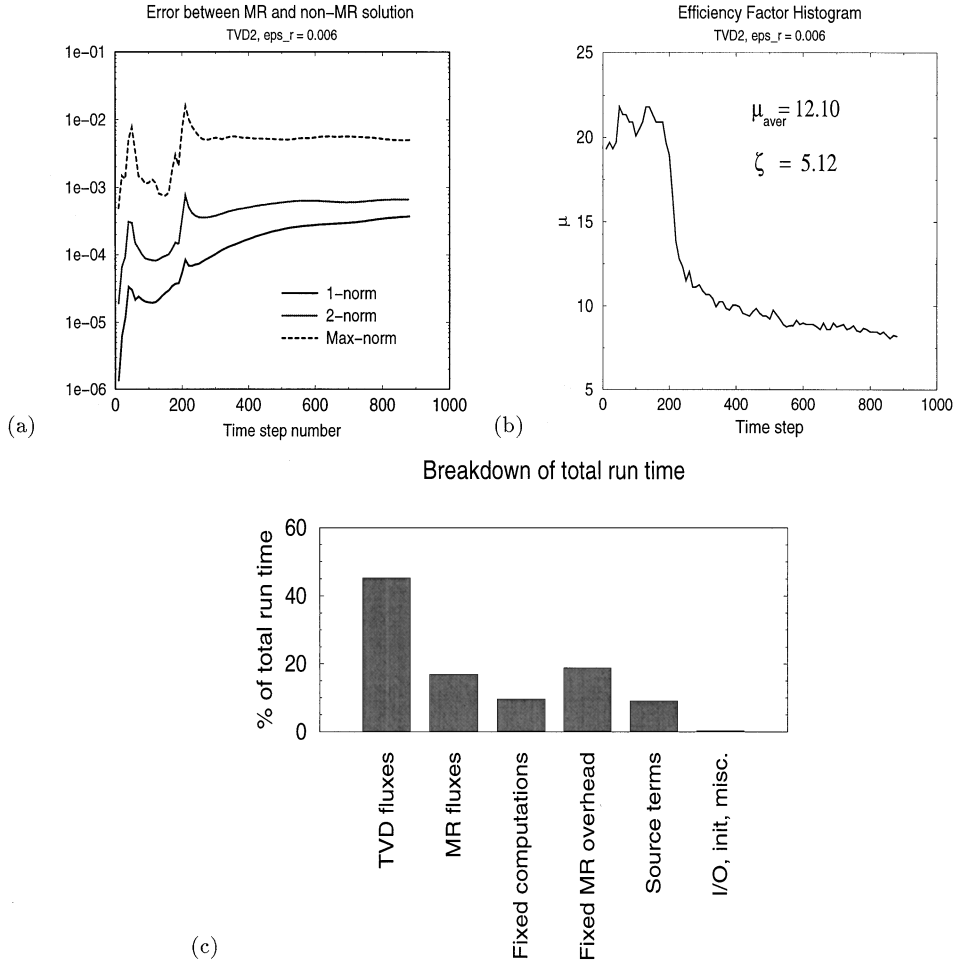

FIG. 7. MR performance for TVD2 with IC (3.1): (a) errors, (b) efficiency, and (c) execution profile.

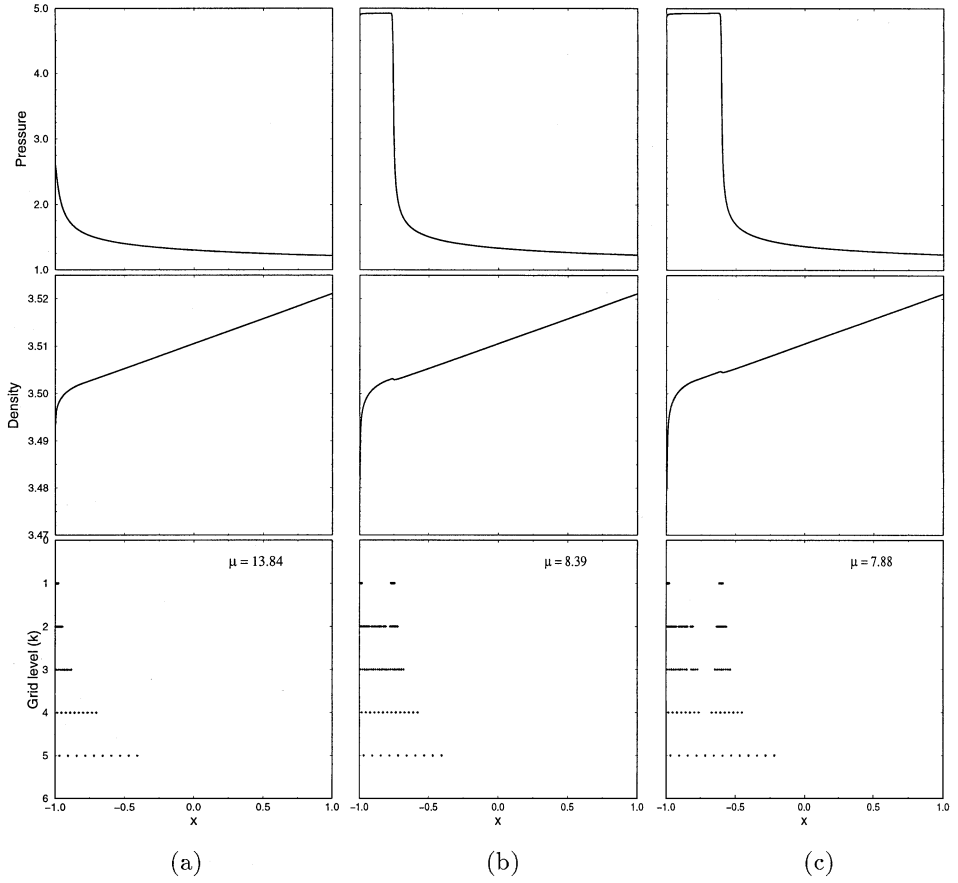

FIG. 8. ENO3 solution and MR diagram with IC (3.1) at (a) $t = 0.3$, (b) $t = 0.303$, (c) $t = 0.305$, (d) $t = 0.31$, (e) $t = 0.315$, and (f) $t = 0.32$.

the effect of the shallow slope of $\omega = 0.1$ is added, the result is a "slowly brewing" fluid everywhere initially since the reaction is taking place in the entire domain at a low rate. Once the temperature rise triggers full ignition at the left boundary, the reaction front sweeps across the domain from left to right at such a high speed that the fluid dynamics does not have time to catch up. There are no distinguishable shocks, expansions, or contacts in this problem, only the reaction front which remains extremely sharp throughout.

As explained in some detail in Subsection 3.1, this problem requires an extremely fine grid to solve with an acceptable accuracy. The flow is unsteady, so at some point in time a fine grid will be needed at any given location in the domain. To improve the efficiency, one can resort only to an adaptive method of some sort. As such, the multiresolution method seems to be an ideal candidate, given its built-in ability to detect discontinuities. When the scheme described in Section 4 is applied to system (2.1), we obtain the solutions depicted in Figs. 6, 8, and 10, where we took a snapshot of the solution at six different times for each run. The only difference between the runs is the order of accuracy. We chose a grid of $N = 1024$ for all cases, which seems to be fine enough to give a solution quality which is reasonable with TVD2 and good with higher order ENO. Each of the six snapshots in each figure contains two primitive variables, pressure and density, plus a multiresolution

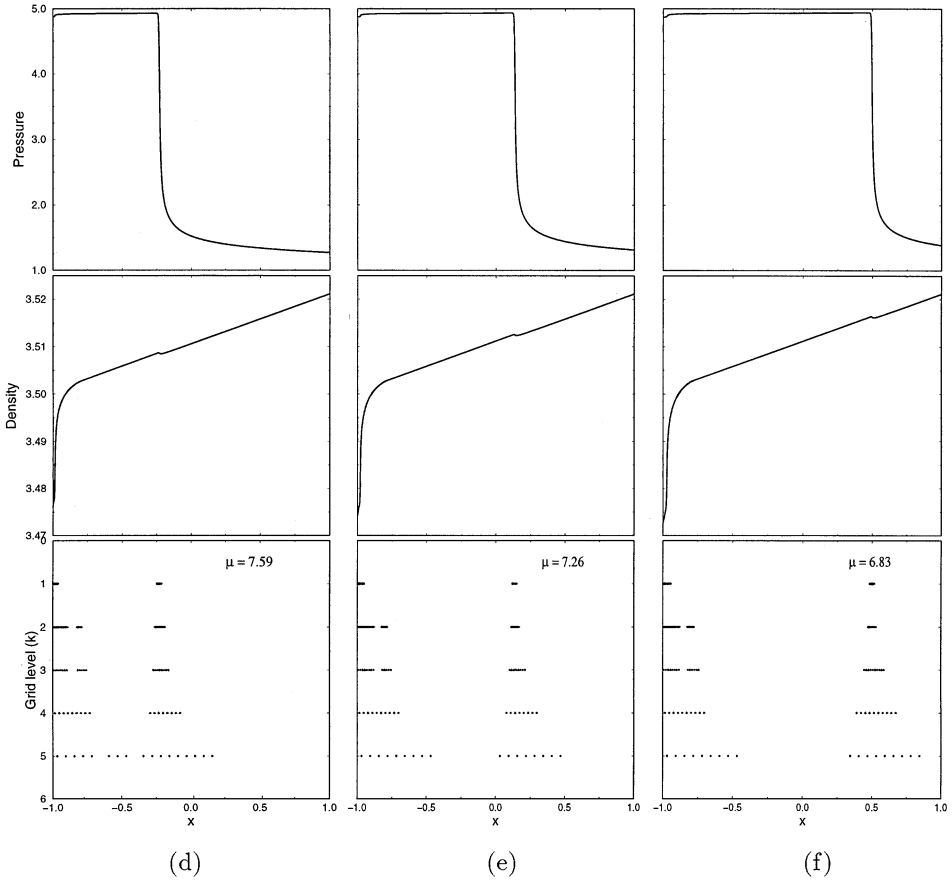

FIG. 8—*Continued*

diagram to show its performance. The MR diagram has a symbol at the cell centroid of each cell whose multiresolution coefficients are larger than the tolerance, and thus its *children cells* are flagged for "exact computations" by the truncation process (4.6). Note that these marks correspond directly to the array i_j^k of module (4.6); elements of i_j^k set to 1 designate the *parents* of those cells that need to be computed via the traditional flux computation.

5.1.1. TVD scheme. A detail examination of Fig. 6 reveals that the multiresolution scheme for TVD2 performs as expected in terms of preciseness, in particular in being able to pick out and follow the reaction front reliably and accurately. Flagged cells are created at the left boundary *before* the reaction wave forms (Fig. 6a), because of the large slope in density (and velocity, not shown). An MR "spike" remains there for the rest of the computation. As the front forms and moves to the left, as early as $t = 0.303$ (Fig. 6b), we already see a cleavage from the left spike, two levels deep. In time, this split between the two spikes widens and deepens, and by $t = 0.31$, it cuts across all levels. The location of the front is always precisely identified by the location of the second, very narrow MR spike which moves along with it. As a secondary feature, we note the development of a "shoulder" in the density profiles, which are most clearly shown by the MR diagrams in Figs. 6c–6e, where we can clearly identify a 3-level high, stationary spike several coarse grid cells in width. Finally, from $t = 0.305$ on we note, albeit barely visible, oscillations on a very fine scale in the shoulder area of the density

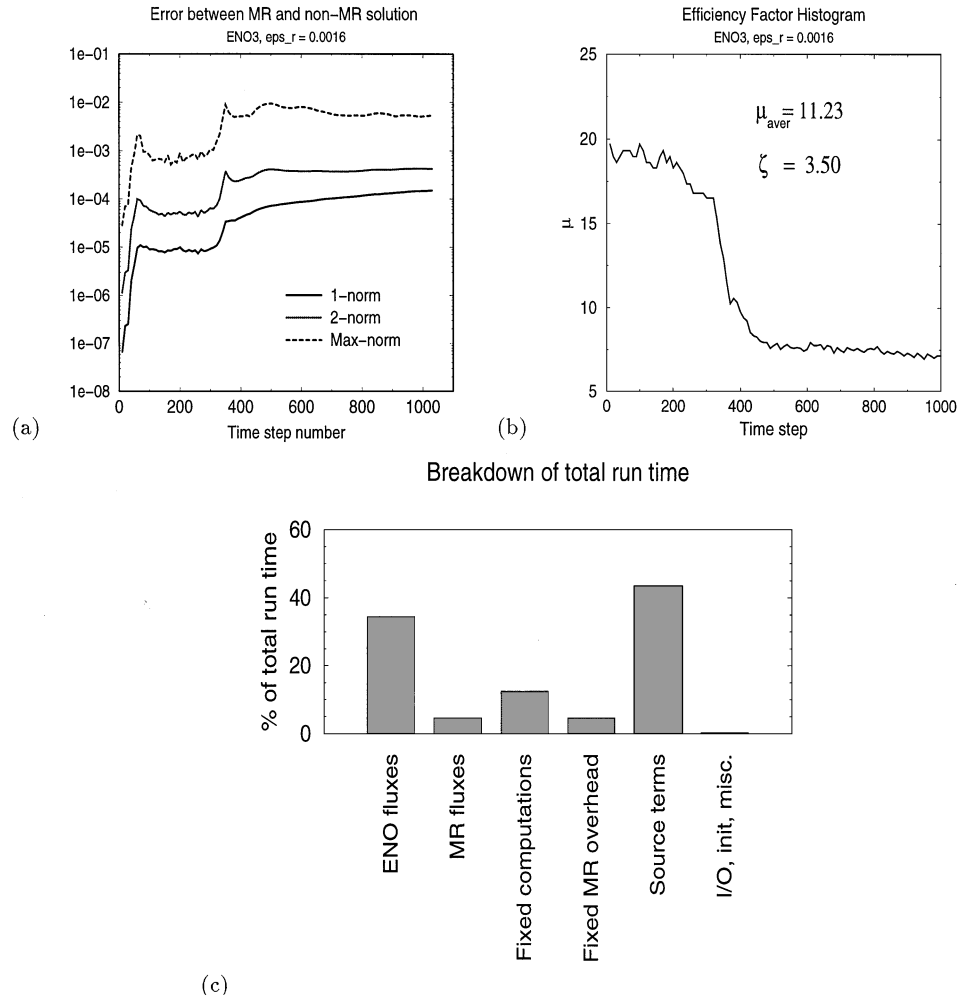

FIG. 9. MR performance for ENO3 with IC (3.1): (a) errors, (b) efficiency, and (c) execution profile.

profiles. These tiny oscillations are caused by the MR interpolation, but they do not grow appreciably in time, and they are certainly within the MR tolerance, as shown by the diagrams.

The tolerance level for this case was set at $\epsilon_r = 0.006$, and a histogram of the errors, compared to the non-MR solution, shows that both the l_1 and l_2 errors are well below this tolerance, which is rarely exceeded even by the l_∞ error (see Fig. 7a). It should be emphasized that our theoretical "guarantee" in terms of quality extends only over one stage of one time step. However, as we have seen in several earlier works, if ϵ_r is chosen judiciously, the MR-error will grow sublinearly, suggesting that it is always less than the global error (compared to the exact solution), which is linear in time. In fact, e_∞ is practically constant after about the first one-fourth of the run.

Once the correctness of the MR solution is established, the efficency histogram gives us the second most important indicator of performance. As shown in Fig. 7b, μ ranges from 8 to 22, with an average value of 12.1 over about 900 time steps. When compared to previous test cases, this is clearly an exceptional performance (see [1–3, 5, 6, 8]), partly due to the nature of the problem which is very well suited for MR: (i) few, very sharp discontinuities,

(ii) a very fine grid is required. In addition to the snapshots of Fig. 6, Fig. 7b also shows that μ stays around 8–9 for most of the front's propagation through the domain. This plot is the most convincing when it comes to satisfaction of the "preciseness" criterion: the irregularity (i.e., front) does not change its width, and there are no new ones forming, thus we know that after separation from the left boundary μ should stay fairly constant.

When the code was timed with and without MR, the speed-up factor ζ came out to be 5.12. This is an impressive result, even if one considers that the efficiency was more than twice that, since there is a fixed amount of computation both in the MR and non-MR parts of the code (e.g., source terms are computed everywhere). We found it illuminating to plot the percentage of time spent in six major parts of the code. The bar-chart of Fig. 7c shows that even with an average efficiency of 12.1, the single largest chunk of time is spent in the (old) flux routines ("TVD fluxes"). The "MR fluxes" portion mainly corresponds to the RHS interpolations in (4.8), and it contributes 16.9% to the total run time, whereas the share of encoding and truncation ("fixed MR overhead") is 18.9%, bringing the total MR portion to 35.8%. Since no special reconstruction procedure was necessary for the source terms, their calculation amounted to 9.1% only.

5.1.2. Third-order ENO scheme. In Fig. 8 we provide a collection of snapshots for the ENO3 scheme with the MR adaptation, in the same format as Fig. 6. One of the few differences between the two solutions is the disappearance of the density spike from the ENO3 solution. Features also tend to be sharper, and the front is "ahead" of its TVD2 counterpart. Correspondingly, the MR diagrams in the ENO3 scheme are crisper also. For example, the separation between the spikes at $t = 0.305$ already spans four MR levels (Fig. 8c), and, in the same plot, the shoulder already appears. The shoulder is definitely more well-defined in the density profiles, but in terms of MR diagram also: the secondary spike goes all the way up to the second finest level and stays there for the whole run (Figs. 8c–8f).

An important difference between the TVD2 and ENO3 runs was the reduction of the tolerance ϵ_r to 0.0016, which was warranted by the higher spatial and temporal accuracies. Since the underlying scheme is now more accurate than the second-order TVD scheme, we should also expect it to be approximated by the MR scheme to a higher accuracy. A corresponding reduction in e_1 and e_2 is found when Fig. 9a is compared to Fig. 7a, while e_∞ is roughly the same. Qualitatively speaking, the errors behave a little better in terms of tapering off in time, but overall they exhibit the same trend as for the TVD2 scheme.

The efficiency histogram of Fig. 9b again shows the high initial efficiencies due to smoothness, which drops off as the front develops and stays at around 7.5 as it propagates. The average efficiency over more than 1000 time steps is 11.23. The actual speed-up turned out to be $\zeta = 3.5$. While this result is still very pleasing, it is lower than the TVD2 case, even when the difference in the corresponding average efficiencies is taken into account. The detailed code profile of Fig. 9c provides the answer. While the share of ENO flux computations decreases by about 10%, the percentage of source term calculations jumps from 9.1% to 43.3%! As mentioned before, we chose to compute source terms everywhere exactly, since otherwise the truncation process (4.6) would have had to be altered, resulting in lower efficiencies. For TVD2 this posed little extra computation, since reconstruction was trivial. Because of the explicit and extra number of evaluations (see (2.10)) required by higher order accuracy, this fixed extra cost increased considerably, adding to the total run time. We wish to point out that the "MR fluxes" and "fixed MR overhead" categories are about the same in absolute value between Fig. 7c and Fig. 9c, but their relative contribution

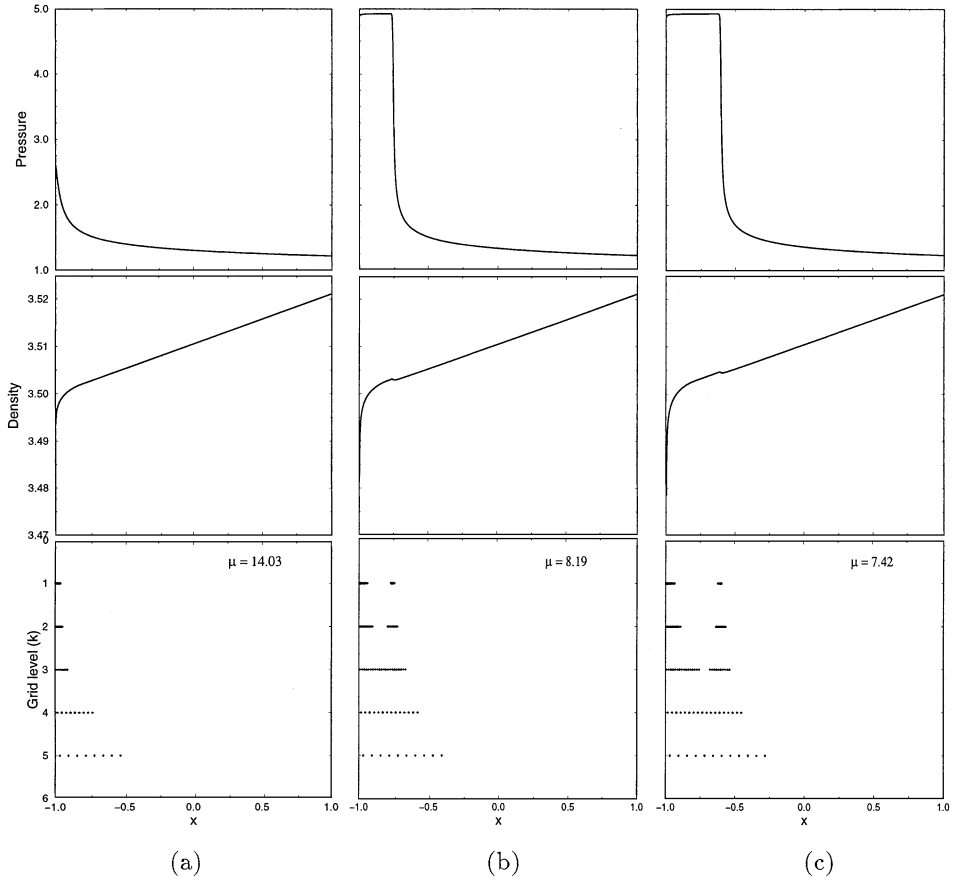

FIG. 10. ENO4 solution and MR diagram with IC (3.1) at (a) $t = 0.3$, (b) $t = 0.303$, (c) $t = 0.305$, (d) $t = 0.31$, (e) $t = 0.315$, and (f) $t = 0.32$.

to the total run time is much smaller for ENO3, since the overall scheme is more expensive. The total MR cost is thus only 9.4%, and when projected to the non-MR scheme it becomes a mere 2.7%. Yet the speed-up factor was less than a third of the efficiency.

5.1.3. Fourth-order ENO scheme. We ran the ENO4 scheme, not just to test a yet higher order accurate scheme, but to examine the behavior of the MR interpolation when $s = 3$ in (4.4b) and (4.8b). Since we must preserve the overall accuracy of the spatial discretization while we interpolate the RHS via (4.8b), the parameter s must be chosen so that $\bar{r} \geq r$. With ENO4, $\bar{r}$ becomes 5, and the MR stencil now spans 5 coarse grid cells. Typically, smooth parts will require fewer and discontinuities will require more flagged cells than with $\bar{r} = 3$. As shown in Fig. 10, the solution itself is visually indistinguishable from the ENO3 solution, but the MR diagrams are noticeably different. The MR spikes are wider, and the distinctive shoulder present in both the TVD2 and the ENO3 solutions is now reduced to a wider flagged area near the left boundary, only from the third level on (Figs. 10c–10f). The explanation lies in the two orders higher interpolation accuracy which is now able to describe the shoulder at coarser levels.

The truncation tolerance was again reduced from the ENO3 value, now to $e_r = 0.0004$, reflected by the l_1 error shown in Fig. 11a. While all of e_1, e_2, and e_∞ are assymptotically

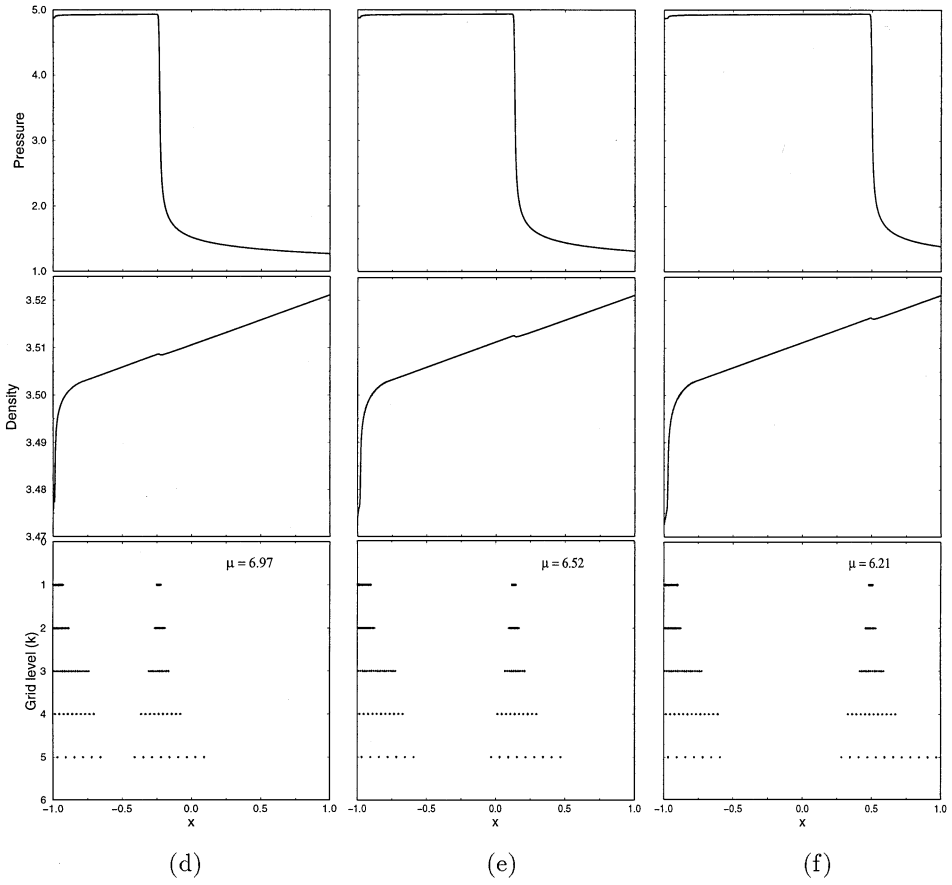

FIG. 10—*Continued*

converging to a constant value, the errors are larger than those for ENO3, e_1 excepted. This is probably due to the higher order accurate encoding being truncated via a tolerance value that was only four times smaller.

Because of the smaller threshold tolerance and wider MR stencils near points of irregularities, we expect lower efficiencies as well. Figure 11b shows the histogram, where the average efficiency was $\nu = 9.75$, and the speed-up was $\zeta = 3.46$. The ratio of these two numbers is actually lower than in the ENO3 case, mainly because ENO4 is more expensive and the source term calculations are of about the same cost in both cases. Comparing the breakdowns shown in Figs. 9c and 11c we see that the balance between the "ENO fluxes" and "source terms" categories is now more favorable toward the former, otherwise the trends are the same. The two MR categories take up a total of 10% of the run time.

5.2. *Low Reaction Rate, Shocked Flow*

In revisiting the non-stiff problem of Subsection 3.2 (with reaction model (2.2b) and IC (3.4)) we examine the behavior of the MR algorithm for a problem where the grid fineness required is not as severe, but where the complex flow features present other challenges. We ran this problem with the same parameter setup as we did earlier, except now using a fine grid resolution of $N = N_0 = 512$ and the MR algorithm of Section 4 with $\epsilon_r = 0.00125$.

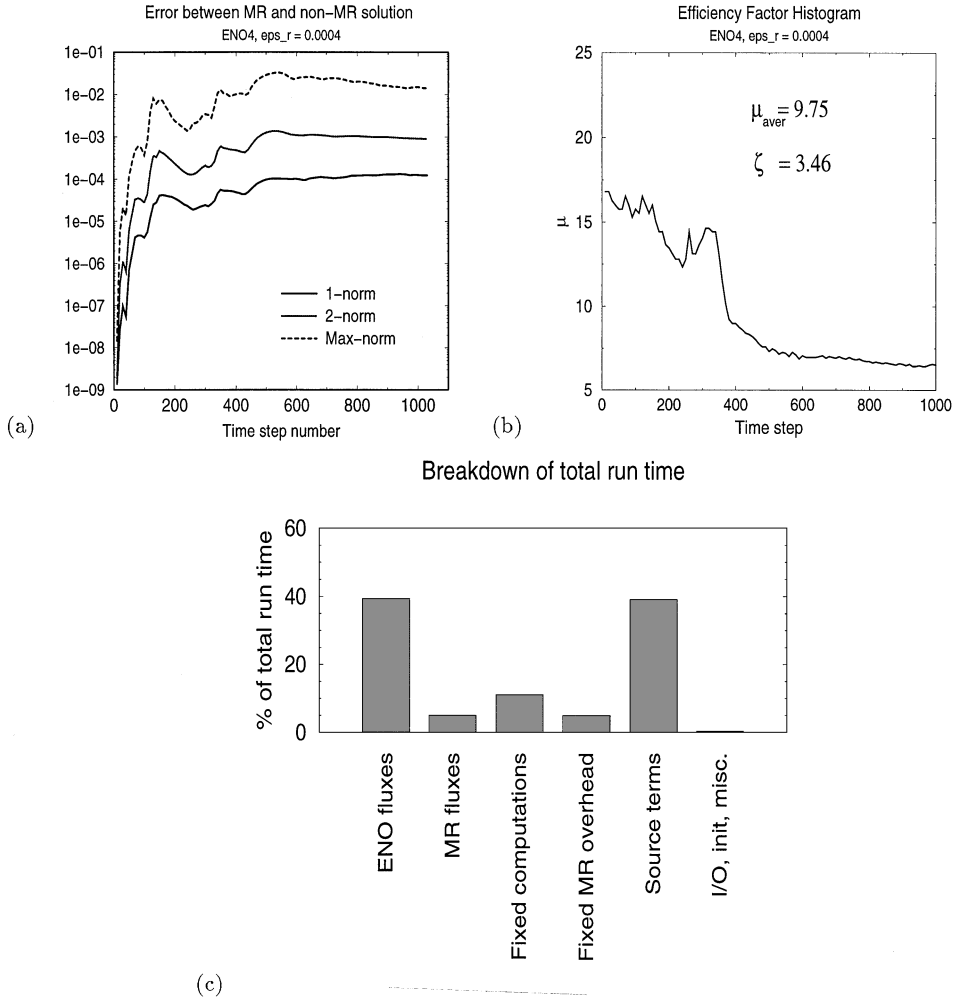

FIG. 11. MR performance for ENO4 with IC (3.1): (a) errors, (b) efficiency, and (c) execution profile.

All experiments in this subsection were conducted using the ENO3 reconstruction, which, as seen in Subsection 3.2 above, seemed to yield a considerably higher quality than the TVD2 scheme. We hope to show that the added cost can be substantially mitigated by the multiresolution speed-up. The results that follow will also point out some of the issues that adversely affect RHS interpolation in some cases.

5.2.1. Initial application of the MR scheme. When we first applied the MR scheme to the low reaction rate problem, at $t = 0.5$ we obtained the pressure profile of Fig. 12a, which is comparable to that of Fig. 5a, run earlier. However, the corresponding MR diagram of Fig. 12b is somewhat disappointing: on the second finest level the majority of the cells behind the shock are flagged and all subsequent levels are totally dense. Keeping in mind that the actual MR coefficients **d** in (4.4b) come not from the primitive, but from the conservative variables, we then plotted the energy (Fig. 12c). The many oscillations in the smooth expansion area are now obvious. This explains the strange MR diagram and the lower than expected efficiency factor of 2.94 there. Note that the shock is oscillation free, and the MR diagram is also correct in its neighborhood.

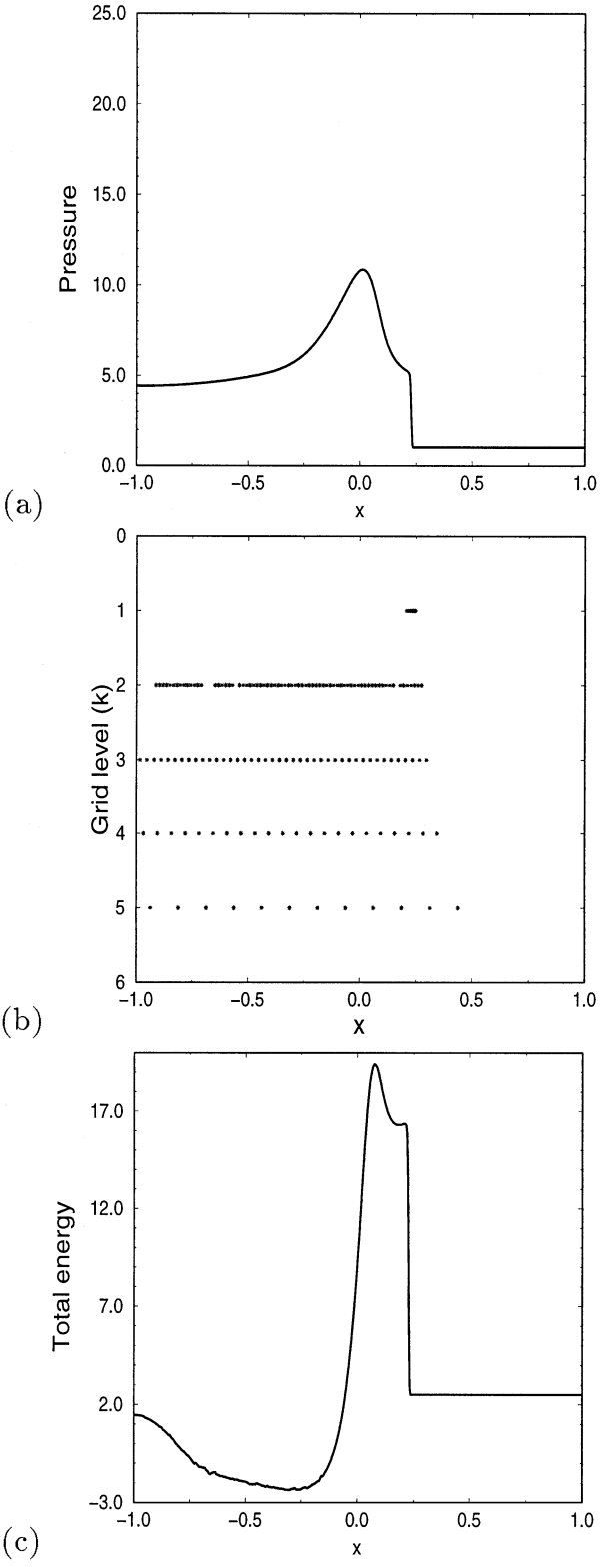

FIG. 12. Solution at $t = 0.5$ with IC (3.4): (a) pressure, (b) MR diagram, and (c) total energy.

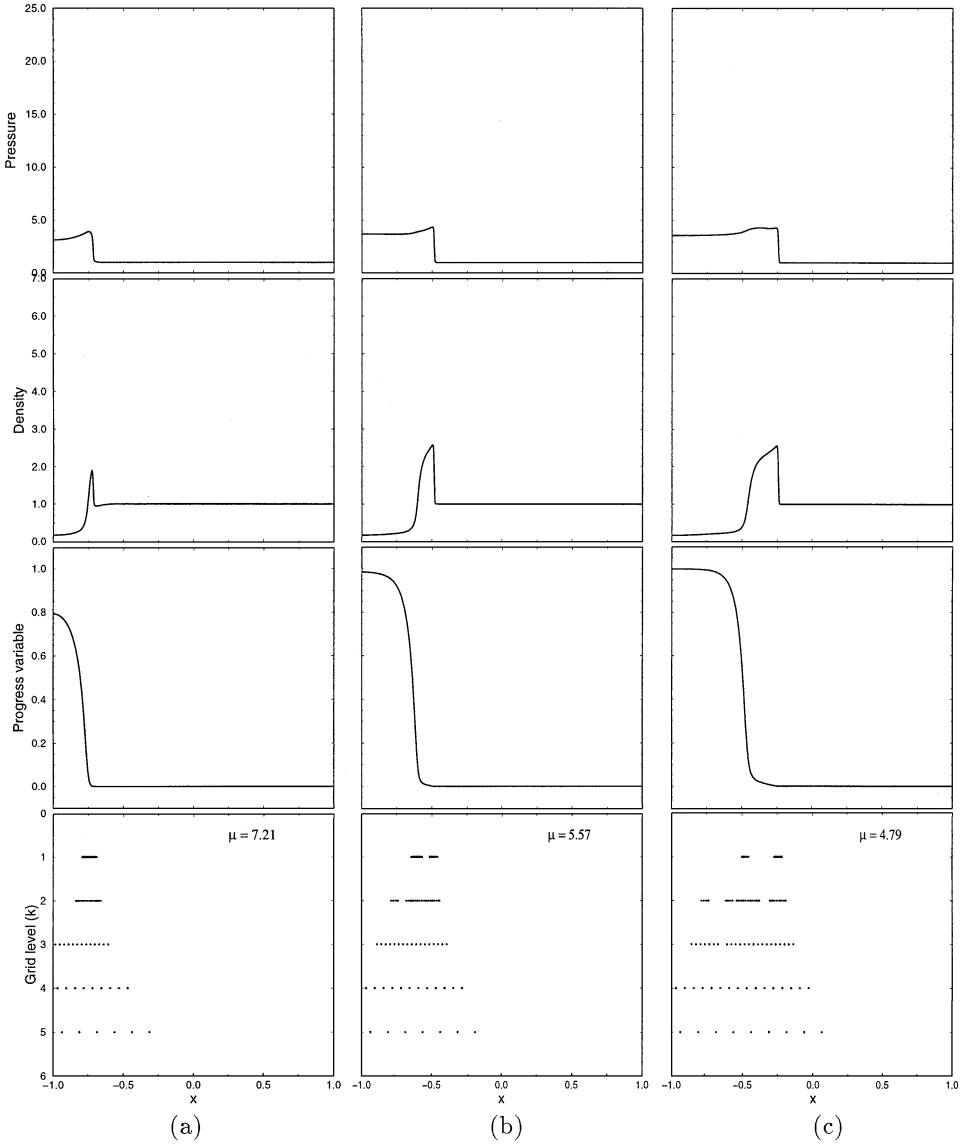

FIG. 13. ENO3 solution and MR diagram with IC (3.4) at (a) $t = 0.1$, (b) $t = 0.2$, (c) $t = 0.3$, (d) $t = 0.4$, (e) $t = 0.5$, and (f) $t = 0.6$.

5.2.2. A different Riemann solver. Since the non-MR solution did not contain these non-fatal, yet bothersome oscillations, the multiresolution scheme was immediately suspect. (Reverting to the lower order TVD2 scheme, with MR turned on, we obtained an energy profile similar to that of Fig. 12c, so the higher order ENO was not the culprit either.) The puzzle was finally solved by the realization that in Roe's Riemann solver, rarefactions are modeled as "expansion shocks." However small, these are discontinuities which, in reality, should be smooth transitions. Sonic glitches caused by Roe's linearization are well known in the literature and are easily fixed by adding a small amount of dissipation, usually only to the characteristic field that is zero. In this case, however, the entropy fix did not help either. What in fact happened was that the fluxes contained these tiny glitches in flagged cells,

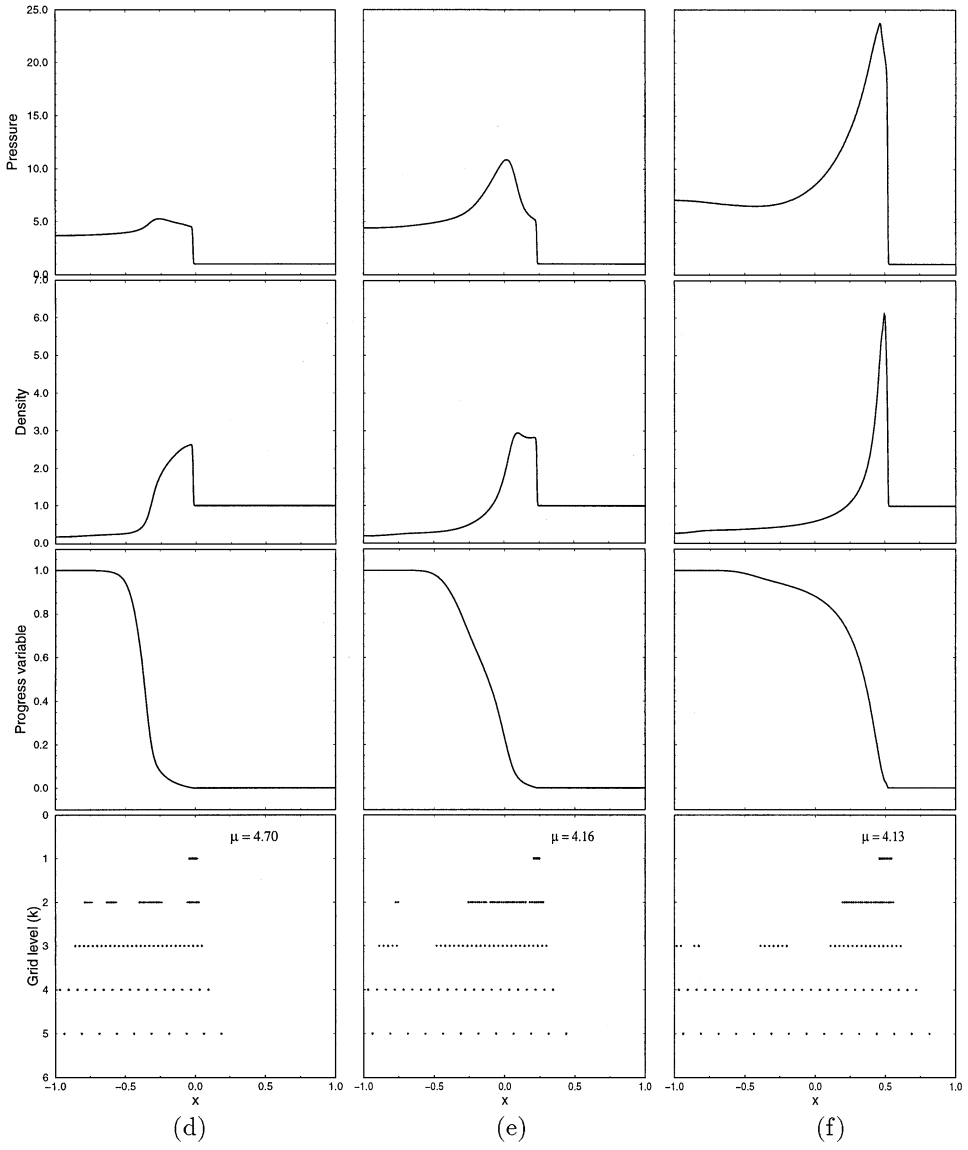

FIG. 13—*Continued*

and in others—in smooth areas—the RHS (and thus indirectly the flux) was interpolated. In this case the latter source of fluxes was actually more accurate than the former, so-called "actual" or "exact" fluxes. This inconsistency, we conjecture, is usually not noticeable, as in the stiff case discussed earlier, and indeed, from the pressure plot of Fig. 12a one cannot tell either. In this case the multiresolution analysis, being so sensitive to oscillations, served also as the indicator of a problem of its own causing. (Of course, one may also argue that even the energy profile of Fig. 12c is acceptable from a practical point of view, and a μ of around 3 is also satisfactory.)

The ultimate test of this explanation is a replacement of Roe's scheme with another Riemann solver. When the exact (nonlinear) Riemann solver (also called "Godunov's Riemann solver" by some) was used, these oscillations indeed disappeared. Figure 13

shows six snapshots of this solution, progress variable and MR diagrams included. Initially the reaction front and the shock are two distinct features, which is well illustrated by the separation on the finest level on both Figs. 13b and 13c. The front becomes smoother in time, and, except for the shock's vicinity, all other finest level flagged cells disappear by $t = 0.4$. A comparison of the MR diagrams on Figs. 13e and 12b (both at $t = 0.5$) reveals a noticeable improvement in efficiency from 2.94 to 4.16. At $t = 0.6$, immediately before the reaction wave coalesces with the shock, there are no significant MR coefficients on the two finest levels, except for the main discontinuity. Some of the marks in the MR diagram are simply due to steep (transient) gradients in the conservative variables, especially in the energy profile (e.g., the left secondary MR-spike(s) in Figs. 13b–13e).

Satisfied with the solution we finally obtained, we are ready to discuss the MR performance. Figure 14a shows that all the errors are on a moderate, seemingly linear growth path. Even by the end of the run, both e_1 and e_2 are below the allowed tolerance value. The behavior is markedly different from those shown in Figs. 7a, 9a, and 11a, due to the dynamics of the problem. This is also reflected by the efficiency histogram of Fig. 14b, where μ is between about 4 and 7, with an average value of 5.15. The speed-up was 2.54.

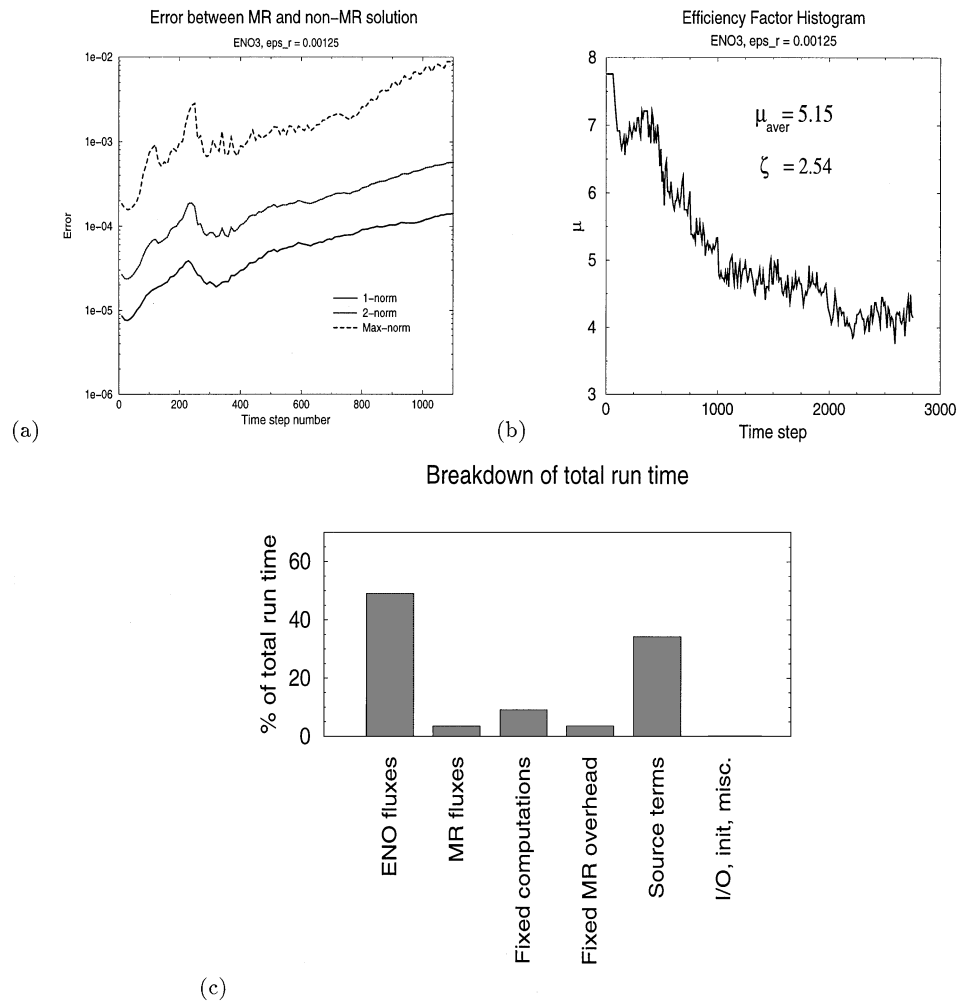

FIG. 14. MR performance for ENO3 with IC (3.4): (a) errors, (b) efficiency, and (c) execution profile.

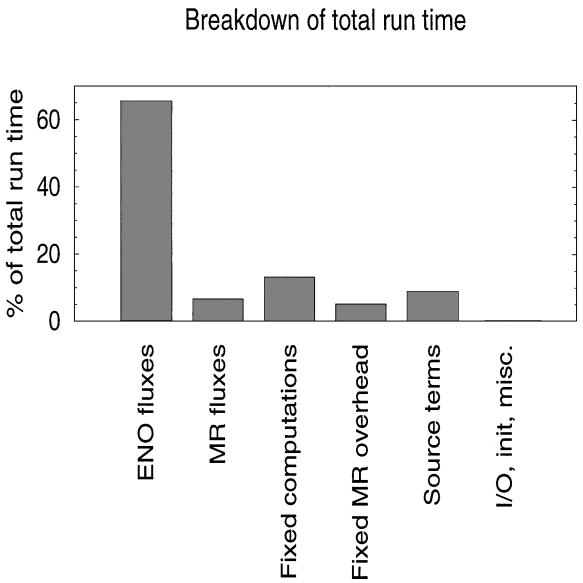

FIG. 15. Execution profile for ENO3 with IC (3.4) when source terms also undergo data compression.

These numbers are very encouraging considering the solution is complex, the grid is twice as coarse as the one used in Subsection 5.1, and source terms are still computed everywhere. This latter fact is well illustrated in Fig. 14c, where source term computations take up as much as 34.3% of the run time.

5.2.3. Source term computation via multiresolution. Up to now, we had two reasons for computing source terms on the finest grid and without MR: (i) for TVD2 this was inexpensive; (ii) when the source terms are stiff, the truncation process must be modified accordingly. For the second case studied in this paper, however, neither of these motivations are applicable, and it would indeed be interesting to compare the two different approaches, holding all other run parameters fixed. We expect no quantitative change in the solution when (4.8) is applied in its original form.

The run held no surprises: on the average, the errors were about 5–10% less than those shown in Fig. 14a, and the efficiencies were also very close to those of Fig. 14b, yieldig an average μ of 5.19 (instead of the previous 5.15). The improvement in the run time, however, was significant: the new run finished about 32% sooner, bringing the speed-up to 3.73 (from 2.54)! The ratio of the efficiency versus speed-up was thus the lowest of all numerical experiments, which suggests that in such cases μ can actually be used as a rough a priori estimator of the actual MR run time. The bar-chart in Fig. 15 shows a distribution where source computations are brought down to 9%, and the total MR cost is 12%. Since the share of MR routines always goes up with the efficiency, it is interesting to also project the MR time onto the original run time to get a μ-independent estimate; in this case the MR cost is 3.22% of the fine grid run time. Even considering the fact that the exact Riemann solver is more expensive than Roe's, these results are very promising.

6. CONCLUDING REMARKS

We have studied the application of the multiresolution method to conservation laws with source terms. After the use of the powerful ENO and MR techniques was briefly justified,

we presented a scheme that combines these two ideas in a unified form that allows for a natural and efficient MR treatment of source terms. When the proposed scheme was applied to two particular cases, results have shown that on the average the actual speed-up ranges from about one-third to two-thirds of the efficiency factor, where the latter was between 5 and 12. The total run time was thus reduced by 3 to 5 times. The overhead due to MR routines was typically about 10% for ENO, and about 36% for TVD. While more research into some of the detail issues is necessary, this one-dimensional application of the multiresolution scheme to the reactive Euler equations has shown a particularly close fit between the method and the physics.

On the natural question of how the method would fare in two or three dimensions with complex geometry, it is important to point out that the performance will likely be highly problem dependent. For unstructured grids, the MR interpolation would have to be a high order, multi-dimensional, central interpolation of the cell averages, whereas in the structured grid case the tensor product approach would suffice. In the latter case, the scalar results from [3] are especially encouraging. In both cases, however, whenever high order ENO is justified by the problem at hand, we feel that the multiresolution method could significantly reduce the run time. In smooth regions the fixed-stencil MR interpolation would not only replace the flux computations which includes multiple quadrature points and Riemann solvers for each face, but would also bypass the ENO stencil selection which can be prohibitively expensive in multiple dimensions.

REFERENCES

1. B. L. Bihari, Multiresolution schemes for conservation laws with viscosity, *J. Comput. Phys.* **123**, 207 (1996).
2. B. L. Bihari and A. Harten, Application of generalized wavelets: an adaptive multiresolution scheme, *J. Comp. Appl. Math.* **61**, 275 (1995).
3. B. L. Bihari and A. Harten, Multiresolution schemes for the numerical solution of 2-D conservation laws, I, *SIAM J. Sci. Comput.* **18**(2), 315 (1997).
4. P. Colella, A. Majda, and V. Roytburd, Theoretical and numerical structure for reacting shock waves, *SIAM J. Sci. Stat. Comput.* **7**(4), 1059 (1986).
5. B. Gottschlich-Müller and S. Müller, *Multiscale Concepts for Conservation Laws*, IGPM-Report 128, RWTH Aachen, June 1996.
6. A. Harten, Adaptive multiresolution schemes for shock computations, *J. Comput. Phys.* **115**, 319 (1994).
7. A. Harten, Discrete multiresolution analysis and generalized wavelets, *Appl. Numer. Math.* **12**, 153 (1993).
8. A. Harten, Multiresolution algorithms for the numerical solutions of hyperbolic conservation laws, *Comm. Pure Appl. Math.* **48**(12), 1305 (1995).
9. A. Harten, High resolution schemes for hyperbolic conservation laws, *J. Comput. Phys.* **49**, 357 (1983).
10. A. Harten and S. Osher, Uniform high order accurate nonoscillatory schemes, I, *SIAM J. Numer. Anal.* **24**, 279 (1987).
11. A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, Uniform high order accurate essentially non-oscillatory schemes, III, *J. Comput. Phys.* **71**, 231 (1987).
12. A. K. Kapila, private communications.
13. R. J. LeVeque and H. C. Yee, A study of numerical methods for hyperbolic conservations laws with stiff source terms, *J. Comput. Phys.* **86**, 187 (1990).
14. S. Xu, T. Aslam, and D. S. Stewart, High resolution numerical simulation of ideal and non-ideal compressible reacting flows with embedded internal boundaries, *Combust. Theory Model.* **1**, 113 (1997).