

Wave Propagation Algorithms for Multidimensional Hyperbolic Systems

Randall J. LeVeque

Department of Applied Mathematics and Department of Mathematics, University of Washington, Box 352420, Seattle, Washington 98195-2420
E-mail: rjl@amath.washington.edu

Received December 4, 1995; revised September 23, 1996

A class of high resolution multidimensional wave-propagation algorithms is described for general time-dependent hyperbolic systems. The methods are based on solving Riemann problems and applying limiter functions to the resulting waves, which are then propagated in a multidimensional manner. For nonlinear systems of conservation laws the methods are conservative and yield good shock resolution. The methods are generalized to hyperbolic systems that are not in conservation form and to problems that include a “capacity function.” Several examples are included for gas dynamics, acoustics in a heterogeneous medium, and advection in a stratified flow on curvilinear grids. The software package CLAWPACK implements these algorithms in Fortran and is freely available on the Web. One and two space dimensions are discussed here, although the algorithms and software have also been extended to three dimensions. © 1997 Academic Press

1. INTRODUCTION

A new class of wave-propagation methods has been developed for solving multidimensional hyperbolic systems of partial differential equations, which includes (but is not limited to) nonlinear systems of conservation laws. These methods are based on solving Riemann problems for waves that define both first-order updates to cell averages and also second-order corrections. These correction terms are modified by limiter functions to obtain high resolution results. In one dimension this follows standard techniques developed over the past two decades for nonlinear conservation laws, particularly the Euler equations of gas dynamics.

The methods are extended to two and three space dimensions by a natural wave-propagation approach that captures the cross-derivative terms needed for second-order accuracy while allowing the use of simple one-dimensional limiters with good effect. Moreover, the methods are stable in general for Courant numbers up to 1, where the Courant number is measured relative to the maximum wave speed in any direction. This is an improvement over the stability bounds of the standard multidimensional Lax–Wendroff method, for example. Stability is discussed in Section 3.7.

In one space dimension the standard conservation law has the form

$$q_t + f(q)_x = 0, \quad (1)$$

where $q \in \mathbb{R}^m$ is the vector of conserved quantities. Here we also consider the more general variable-coefficient quasilinear form

$$q_t + A(q, x, t)q_x = 0, \quad (2)$$

and the methods are formulated in a general manner that also allows their application to hyperbolic systems that are not in conservation form, e.g., the variable-coefficient linear systems of equations for acoustic or elastic wave propagation in a heterogeneous medium.

The methods are based on solving Riemann problems for the wave structure and then introducing a *fluctuation splitting* technique that generalizes the notion of flux-difference splitting from conservation laws. The left-going and right-going fluctuations capture the net effect of all left-going and right-going waves, and these fluctuations are then split in the transverse direction in the generalization to more space dimensions.

A further generalization is obtained by using *capacity-form differencing*, which allows application to problems such as flow in a porous medium with variable porosity [2], or in a stratified flow with variable density (Section 3.9). This form also simplifies application on curvilinear grids (Section 3.10).

These methods have been implemented in the software package CLAWPACK (conservation laws package), a collection of Fortran routines freely available from `netlib` [28]. To browse through this package on the Web, the URL is

<http://www.amath.washington.edu/~rjl/clawpack.html>

This package includes numerous examples for a variety of different hyperbolic systems of equations, including most of the examples in this paper. A primary motivation for

this work was to make the sophisticated high-resolution methods developed largely in the gas dynamics community available to a wider range of users, and it is hoped that this will be useful to both students and researchers in many applications areas. The extensions discussed in this paper were motivated by specific applications problems, and a variety of other applications with sample numerical results can be found in the software and *User Notes* [29]. The software is described very briefly in Section 4, but few implementation details of `CLAWPACK` will be discussed here. These are provided in the `CLAWPACK` documentation and in the *User Notes* [29]. An overview of the package was given in [34] which describes much of the philosophy, although many of the implementation details presented there are now out of date.

The present paper contains a detailed discussion only of the one- and two-dimensional methods. Extension to three space dimensions has been carried out, as well, in joint work with Jan Olav Langseth and will soon be available in `CLAWPACK`. Details are discussed in [23, 24].

Recently `CLAWPACK` has been combined with the adaptive mesh refinement code of Marsha Berger [4, 5, 8–10]. This yields a very general adaptive refinement package that has all of the features of the algorithms presented in this paper, e.g., the ability to handle nonconservative hyperbolic equations, capacity form differencing, and extensions to curvilinear grids. The details of this implementation are presented elsewhere [11]. This `AMRCLAW` software is also freely available [12]. See

<http://www.amath.washington.edu/~rjl/amrclaw>

for details and some sample results.

While the multidimensional wave-propagation algorithms proposed here are somewhat different from multidimensional algorithms in the literature, there are many similarities with other approaches, at least in the case of a conservation law in standard form. In particular the basic idea of obtaining “high-resolution” methods based on some form of slope limiter or flux limiter has a long history of development. Many references, along with an overview of the one-dimensional theory, can be found in [32].

Radvugin [41] has also developed multidimensional upwinding schemes that are based on splitting the coefficient matrices as in Eq. (38), which is the basis of the multidimensional methods developed here. These methods are quite similar for the case of constant coefficient linear systems, although the limiters are applied differently. For nonlinear problems Radvugin’s approach is based on a flux-vector splitting, whereas the wave-propagation algorithms are based on flux-difference splittings or generalizations.

Another class of similar methods is exemplified by the method of Colella [13] (see also [3, 47, 59]). A comparison of the wave-propagation methods with these methods is

given in Section 5. A variety of other approaches to multidimensional methods have also been proposed. A few examples can be found in [15–19, 22, 50, 52, 57].

In the context of advection, an overview of related methods is given in [35]. That paper also describes the wave-propagation algorithms developed here in a relatively simple context, where the algorithms can be more easily understood geometrically.

2. ONE SPACE DIMENSION

The algorithms developed here are based on solving Riemann problems at the interface between grid cells. We first consider the standard conservation law (1) and briefly review the definition of Godunov’s method, writing it in wave-propagation and flux-difference splitting forms that will then be generalized to the quasi-linear system (2). Extension to high resolution second-order methods using limiters will be discussed in Section 2.4.

2.1. Conservation Laws

The Riemann problem for (1) consists of this conservation law together with piecewise constant initial data

$$q(x, 0) = q_0(x) = \begin{cases} q_l & \text{if } x < 0 \\ q_r & \text{if } x > 0. \end{cases} \quad (3)$$

With suitable restrictions on f , the solution is a similarity solution $q(x, t) = Q(x/t)$ which consists of a set of waves moving at constant speeds [14, 25].

For the second-order corrections discussed in Section 2.4, we will need to assume that in fact $q_r - q_l$ can be decomposed as

$$q_r - q_l = \sum_{p=1}^{M_w} \mathcal{W}^p, \quad (4)$$

where $\mathcal{W}^p \in \mathbb{R}^m$ is the jump across the p th wave, M_w is the number of waves, and each wave has an associated wave speed $\lambda^p \in \mathbb{R}$. This requires that all waves be discontinuities, i.e., no rarefaction waves. For nonlinear systems such as the Euler equations we assume an approximate Riemann solver such as Roe’s solver [42] (see Section 3.6) is used to produce this.

The first-order Godunov method is implemented in a form that does not require these waves explicitly, however. Instead it requires a *flux-difference splitting*, which is a decomposition of $f(q_r) - f(q_l)$ into a *left-going flux difference*, denoted symbolically by $\mathcal{A}^- \Delta q$, and a *right-going flux difference*, denoted by $\mathcal{A}^+ \Delta q$, with the property that

$$\mathcal{A}^- \Delta q + \mathcal{A}^+ \Delta q = f(q_r) - f(q_l). \quad (5)$$

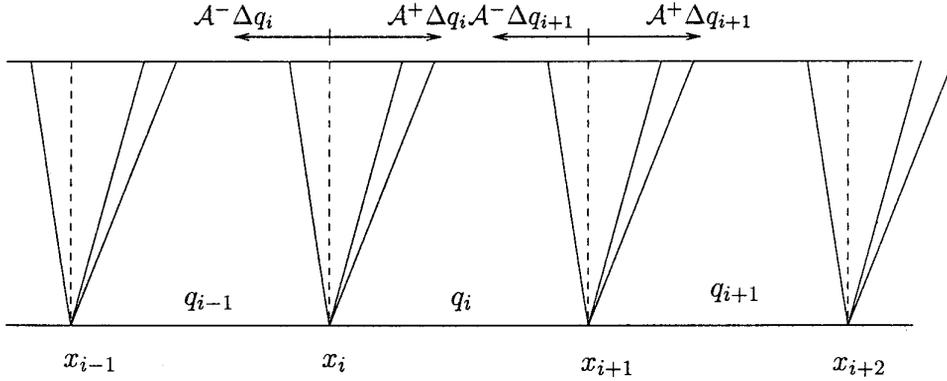


FIG. 1. Waves, interface values, and flux differences for Godunov's method.

For the classical Godunov method, let $q^* = Q(0)$ be the value along $x/t = 0$ in the solution to the Riemann problem. Then

$$\begin{aligned} \mathcal{A}^- \Delta q &= f(q^*) - f(q_l) \\ \mathcal{A}^+ \Delta q &= f(q_r) - f(q^*). \end{aligned} \quad (6)$$

If a wave decomposition of the form (4) is available, one could alternatively set

$$\begin{aligned} \mathcal{A}^- \Delta q &= \sum_p (\lambda^p)^- \mathcal{W}^p \\ \mathcal{A}^+ \Delta q &= \sum_p (\lambda^p)^+ \mathcal{W}^p, \end{aligned} \quad (7)$$

where $\lambda^+ = \max(\lambda, 0)$ and $\lambda^- = \min(\lambda, 0)$. In the case where a Roe solver is used for a nonlinear system, (7) may not satisfy the entropy condition unless an ‘‘entropy fix’’ is included in the definition of $\mathcal{A}^- \Delta q$ and $\mathcal{A}^+ \Delta q$ [21, 32, 45].

The notation $\mathcal{A}^- \Delta q$ and $\mathcal{A}^+ \Delta q$ is motivated by the case of a constant coefficient linear system

$$q_t + Aq_x = 0, \quad (8)$$

in which case

$$\mathcal{A}^- \Delta q = \mathcal{A}^-(q_r - q_l), \quad \mathcal{A}^+ \Delta q = \mathcal{A}^+(q_r - q_l), \quad (9)$$

where the matrices A^\pm are defined as

$$A^\pm = R\Lambda^\pm R^{-1}. \quad (10)$$

Here $R = [r^1 | r^2 | \dots | r^m]$ is the matrix of right eigenvectors (with some choice of normalization), $\Lambda = \text{diag}(\lambda^1, \dots, \lambda^m)$ is the eigenvalue matrix, and $\Lambda^\pm = \text{diag}((\lambda^1)^\pm, \dots, (\lambda^m)^\pm)$. Note that in this case $\mathcal{W}^p = \alpha^p r^p$, where $\alpha = R^{-1} \Delta q$.

Now consider a one-dimensional grid with cell average q_i^n in the grid cell $[x_i, x_{i+1}]$ at time t_n (see Fig. 1). Godunov's

method is obtained by constructing a solution over the time step as indicated in the figure. With piecewise constant initial data we can solve Riemann problems at each interface and piece these together to get the global solution for a sufficiently small time step. Averaging this solution over the i th grid cell at time t_{n+1} gives the new cell average q_i^{n+1} . Below we generally drop the superscript on q_i^n and denote q_i^{n+1} by \bar{q}_i .

By integrating the conservation law over this grid cell one can then show by a standard argument (e.g., [32]) that the new cell average is given by

$$\bar{q}_i = q_i - \frac{\Delta t}{\Delta x} (f(q_{i+1}^*) - f(q_i^*)), \quad (11)$$

where q_i^* is the intermediate state arising in solving the Riemann problem at x_i . This numerical method is clearly in standard conservation form and is typically stable for Courant numbers up to 1.

For purposes of generalizing the methods, however, we will rewrite Godunov's method in a different form. Manipulating (11) with the help of the expressions (6), we can derive

$$\bar{q}_i = q_i - \frac{\Delta t}{\Delta x} (\mathcal{A}^+ \Delta q_i + \mathcal{A}^- \Delta q_{i+1}). \quad (12)$$

Here $\mathcal{A}^+ \Delta q_i$ is the right-going flux difference from solving the Riemann problem between q_{i-1} and q_i . From the interpretation (7) we see that this models the combined effect on the cell average q_i of all waves entering the cell from the left edge. Similarly, $\mathcal{A}^- \Delta q_{i+1}$ is the left-going flux difference from the Riemann problem between q_i and q_{i+1} , and models the combined effect of all waves entering the cell from the right (see Fig. 1).

The form (12) will be used in general. For a system of conservation laws, this method is conservative and consistent for any flux-difference splitting that satisfies (5).

2.2. Nonconservative Systems

The extension to more general hyperbolic systems is best illustrated by a simple example. Consider the equations for acoustics in a heterogeneous medium. The equations can be written as a first-order variable coefficient linear system

$$\begin{aligned} p_t + K(x)u_x &= 0 \\ \rho(x)u_t + p_x &= 0, \end{aligned} \quad (13)$$

where the unknowns are the pressure perturbation $p(x, t)$ and the velocity $u(x, t)$. The variable coefficients are the density $\rho(x)$ and bulk modulus of elasticity $K(x)$. We can write this system as

$$q_t + A(x)q_x = 0,$$

where

$$q = \begin{bmatrix} p \\ u \end{bmatrix}, \quad A = \begin{bmatrix} 0 & K(x) \\ 1/\rho(x) & 0 \end{bmatrix}.$$

We assume that the i th cell has material parameters ρ_i and K_i and set

$$A_i = \begin{bmatrix} 0 & K_i \\ 1/\rho_i & 0 \end{bmatrix}.$$

The sound speed in the i th cell is $c_i = \sqrt{K_i/\rho_i}$. The solution to the Riemann problem between states q_{i-1} in cell $i-1$ and q_i in cell i consists of two waves. The left-going wave moves into cell $i-1$ with velocity $\lambda_i^1 = -c_{i-1}$ and the jump across this wave must be a scalar multiple of the eigenvector r_{i-1}^1 of the matrix A_{i-1} ,

$$\mathcal{W}_i^1 = \alpha_i^1 \begin{bmatrix} -c_{i-1} \\ 1/\rho_{i-1} \end{bmatrix}.$$

The right-going wave moves into cell i with velocity $\lambda_i^2 = c_i$ and the jump across this wave is a multiple of r_i^2 ,

$$\mathcal{W}_i^2 = \alpha_i^2 \begin{bmatrix} c_i \\ 1/\rho_i \end{bmatrix},$$

for some scalar α_i^2 . The state between the two waves must be continuous across the interface (the proper physical jump condition) and so

$$q_{i-1} + \mathcal{W}_i^1 = q_i - \mathcal{W}_i^2.$$

From this we obtain a linear system which can be solved for the wave strengths α_i^1 and α_i^2 , yielding

$$\begin{aligned} \alpha_i^1 &= (-\Delta q^1/\rho_i + c_i \Delta q^2)/(c_{i-1}/\rho_i + c_i/\rho_{i-1}) \\ \alpha_i^2 &= (-\Delta q^1/\rho_{i-1} + c_{i-1} \Delta q^2)/(c_{i-1}/\rho_i + c_i/\rho_{i-1}). \end{aligned}$$

We then define the ‘‘flux-difference splitting’’ by

$$\begin{aligned} \mathcal{A}^- \Delta q_i &= \lambda_i^1 \mathcal{W}_i^1 \\ \mathcal{A}^+ \Delta q_i &= \lambda_i^2 \mathcal{W}_i^2. \end{aligned} \quad (14)$$

Note that this is not really the splitting of any flux difference in this nonconservative problem. In particular,

$$\mathcal{A}^- \Delta q_i + \mathcal{A}^+ \Delta q_i \neq A_i q_i - A_{i-1} q_{i-1}.$$

Nonetheless, Godunov’s method, with the same physical interpretation of solving Riemann problems based on piecewise constant initial data and then computing cell averages to define \bar{q}_i , can be implemented in the form (12) and is effective for this problem. (Numerical results are presented in Section 2.6 after introducing the second-order corrections.)

2.3. Fluctuation Splitting

It seems desirable to have a term for $\mathcal{A}^- \Delta q$ and $\mathcal{A}^+ \Delta q$ other than ‘‘flux-difference splitting’’ in the general case where there may be no flux function. In early work of Roe (e.g., [43, 44]) the term *fluctuation* was often used for flux differences and was even used in the context of nonconservative formulations of the Euler equations in [44]. I propose reviving this term with a specific meaning that generalizes the notion of a flux difference. For a general Riemann problem we can define a *fluctuation* $\mathcal{A} \Delta q$ which will be split into a *left-going fluctuation* $\mathcal{A}^- \Delta q$ and a *right-going fluctuation* $\mathcal{A}^+ \Delta q$. The definition used here will be different from Roe’s use of the term in two space dimensions, however. Roe viewed the fluctuation as measuring the deviation from equilibrium in a steady-state problem, and so in two space dimensions he defined the fluctuation as an approximation to $f_x + g_y$ over the grid cell. Here the fluctuation will always refer to the total effect on the solution due to waves arising from a one-dimensional Riemann problem at a cell edge, in any number of space dimensions, and will be associated with a Riemann problem rather than a cell.

In the acoustics example given above, we first defined waves \mathcal{W}^p with speeds λ^p for each Riemann problem, and the fluctuation splitting was then naturally defined by (14). This generalizes to (7) for a problem with M_w waves in

the solution to the Riemann problem. The total fluctuation $\mathcal{A}\Delta q$ can then be defined *a posteriori* as

$$\mathcal{A}\Delta q = \mathcal{A}^-\Delta q + \mathcal{A}^+\Delta q = \sum_{p=1}^{M_w} \lambda_i^p \mathcal{W}_i^p.$$

Note that this has the physical interpretation of measuring the total effect of waves arising from this Riemann problem on the total integral of q (per unit time).

Alternatively, we can define the fluctuation without reference to the waves if we assume the Riemann problem between states q_l and q_r has a similarity solution of the form $Q(x/t)$, as it must have whenever we can define waves propagating at constant speeds, which is a basic assumption for the methods developed here. (For a variable coefficient problem such as the acoustics problem above, this requires using piecewise constant coefficients in the definition of the Riemann problem. This is consistent with the accuracy that can be expected for such problems, even with second-order corrections added to the algorithm as discussed in the next section.) Then we can define the fluctuation associated with this Riemann problem by

$$\mathcal{A}\Delta q = \int_{-\infty}^{\infty} (Q(\xi) - q_0(\xi)) d\xi, \quad (15)$$

where q_0 is defined by (3). Because of the finite propagation speeds, this integrand is nonzero over a bounded region and the integral is finite. The fluctuation splitting is then given by

$$\begin{aligned} \mathcal{A}^-\Delta q &= \int_{-\infty}^0 (Q(\xi) - q_0(\xi)) d\xi \\ &= \int_{-\infty}^0 (Q(\xi) - q_l) d\xi, \end{aligned} \quad (16)$$

$$\begin{aligned} \mathcal{A}^+\Delta q &= \int_0^{\infty} (Q(\xi) - q_0(\xi)) d\xi \\ &= \int_0^{\infty} (Q(\xi) - q_r) d\xi. \end{aligned} \quad (17)$$

Clearly these are exactly the integrals needed in defining Godunov's method via integration over the grid cell at time t_{n+1} , and again the method takes the form (12).

2.4. Second-Order Corrections

Godunov's method is extended to a high resolution method by adding an additional term. The form of the extended method is

$$\bar{q}_i = q_i - \frac{\Delta t}{\Delta x} (\mathcal{A}^+\Delta q_i + \mathcal{A}^-\Delta q_{i+1}) - \frac{\Delta t}{\Delta x} (\tilde{F}_{i+1} - \tilde{F}_i). \quad (18)$$

Note that the corrections are in a ‘‘flux-differencing’’ form, and in fact this form can be used even in the case of nonconservative equations. The correction flux \tilde{F}_i is defined in terms of the waves \mathcal{W}_i^p and speeds λ_i^p arising from the i th Riemann problem, and it is only at this point that the individual waves are needed, rather than the lumped fluctuations $\mathcal{A}^\pm\Delta q$. A decomposition into waves allows us to apply limiter functions to reduce oscillations near discontinuities by comparing the p th wave to the p th wave arising from the neighboring Riemann problem. This is discussed below.

In the absence of limiters, the second-order corrections take the form

$$\tilde{F}_i = \frac{1}{2} \sum_{p=1}^{M_w} |\lambda_i^p| \left(1 - \frac{\Delta t}{\Delta x} |\lambda_i^p| \right) \mathcal{W}_i^p. \quad (19)$$

This is a standard expression; see, e.g., [32, 58]. It can be interpreted as arising from propagating a piecewise linear ‘‘correction wave’’ as described in [30, 31, 35].

For the linear system (8), the method (18) with fluctuations (9) and corrections (19) reduces to the standard Lax–Wendroff method. For an autonomous nonlinear system of conservation laws of the form $q_t + f(q)_x = 0$, these correction terms give full second-order accuracy and the method is a variant of Lax–Wendroff in this case.

For nonautonomous problems of the form $q_t + f(q, x)_x = 0$ or $q_t + A(q, x)q_x = 0$ (e.g., the acoustics problem above), this form of the corrections does not give second-order accuracy formally. However, it does eliminate the dominant diffusive term in the first-order error and gives results that are as well resolved as we would normally expect from a high-resolution method. The remaining first-order error corresponds to a slight shift in the location of the solution, rather than in the excessive smearing seen with the first-order upwind method, for example.

To see this, consider the advection equation

$$q_t + u(x)q_x = 0 \quad (20)$$

with $u(x)$ a given smooth function. Taylor expansion shows that

$$\begin{aligned} q(x, t + \Delta t) &= q(x, t) + \Delta t q_t(x, t) + \frac{1}{2}(\Delta t)^2 q_{tt}(x, t) + \cdots \\ &= q - \Delta t u(x)q_x + \frac{1}{2}(\Delta t)^2 u(x)[u(x)q_x]_x + \cdots \end{aligned}$$

For second-order accuracy we need to match the first three terms, which can be rewritten as

$$q - \Delta t u(x)q_x + \frac{1}{2}(\Delta t)^2 [u^2(x)q_{xx} + u(x)u'(x)q_x]. \quad (21)$$

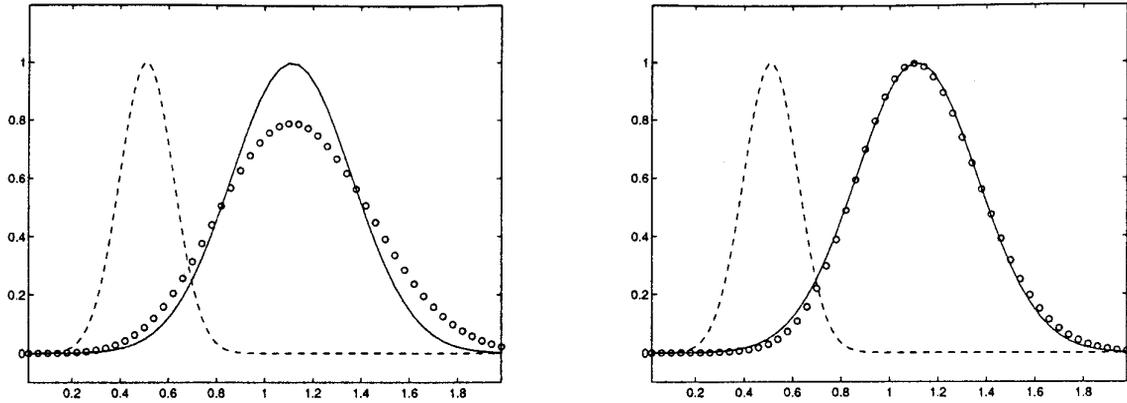


FIG. 2. Solution to $q_t + xq_x = 0$ with first-order upwind method (on left) and using “second-order” correction terms (on right). The dashed line shows the initial data.

The method proposed here can be applied to this problem by setting

$$\begin{aligned} \mathcal{W}_i^1 &= q_i - q_{i-1}, \quad \lambda_i^1 = u_{i-1/2}, \\ \mathcal{A} - \Delta q_i &= u_{i-1/2}^-(q_i - q_{i-1}), \\ \mathcal{A} + \Delta q_i &= u_{i+1/2}^+(q_i - q_{i-1}). \end{aligned}$$

Suppose, for example, that $u > 0$ everywhere. Then the above method, with “second-order” corrections, can be arranged to yield

$$\begin{aligned} q_i^{n+1} &= q_i^n - \frac{\Delta t}{2\Delta x} (u_{i-1/2}(q_i^n - q_{i-1}^n) + u_{i+1/2}(q_{i+1}^n - q_i^n)) \\ &\quad + \frac{1}{2} \left(\frac{\Delta t}{\Delta x} \right)^2 (u_{i+1/2}^2(q_{i+1}^n - q_i^n) - u_{i-1/2}^2(q_i^n - q_{i-1}^n)). \end{aligned}$$

This is clearly a second-order accurate approximation to

$$q - \Delta t u(x) q_x + \frac{1}{2} (\Delta t)^2 [u^2(x) q_{xx}]_x,$$

which can be rewritten as

$$q - \Delta t u(x) q_x + \frac{1}{2} (\Delta t)^2 [u^2(x) q_{xx} + 2u(x) u'(x) q_x].$$

Comparing this with (21) shows that we have correctly modeled the $u^2 q_{xx}$ term (failing to do so results in numerical diffusion as in upwind), but we have an error of $\frac{1}{2} (\Delta t)^2 u(x) u'(x) q_x$. This is equivalent to replacing $u(x) q_x$ in the second term by

$$(1 - \frac{1}{2} \Delta t u'(x)) u(x) q_x \approx u(x - \frac{1}{2} \Delta t u(x)) q_x.$$

In other words, there is an $O(\Delta t)$ error in the advection speed. The resulting error is much less dramatic than with a diffusive first-order method. This is illustrated in Fig. 2, where the equation $q_t + xq_x = 0$ has been solved both with and without the “second-order” correction terms. The equations are solved on a grid with $\Delta x = 0.04$ and $\Delta t = 0.009$.

Although it would be possible to modify the algorithm to obtain formal second-order accuracy on smooth solutions, it would no longer be in the unified framework that allows the limiters introduced in the next section to be applied in the standard manner on a wider variety of problems. The interest here is in problems where the use of such limiters is important (e.g., discontinuous solutions) and, once the limiters are introduced, formal second-order accuracy is typically lost anyway. The extension to nonautonomous problems introduced here gives algorithms that work essentially as well on such problems as on classical autonomous conservation laws. For further illustration of this, see the example in Section 2.6.

2.5. Wave Limiters

To reduce spurious oscillations and obtain a high resolution method it is necessary to introduce limiter functions that modify (19) near discontinuities. Depending on the interpretation these are generally called *flux limiters* or *slope limiters*. (See, e.g., [20, 58, 60].) In the present context these could also be called *wave limiters*, as it is the magnitude of the wave \mathcal{W}_i^p in (19) that will be modified.

We replace each \mathcal{W}_i^p by a limited version $\tilde{\mathcal{W}}_i^p$, which is obtained by comparing \mathcal{W}_i^p to the corresponding p -wave \mathcal{W}_{i-1}^p or \mathcal{W}_{i+1}^p arising from the solution to the Riemann problem at the adjacent grid point to the left or right. The direction is chosen to be the upwind direc-

tion; i.e., we look to the left if $\lambda_i^p > 0$ and to the right if $\lambda_i^p < 0$.

In the case of the linear system (8) we have

$$\mathcal{W}_i^p = \alpha_i^p r^p,$$

where α_i^p is a scalar and the vector r^p is independent of i . Then we can simply apply the limiter to the scalars α_i^p , setting $\tilde{\mathcal{W}}_i^p = \tilde{\alpha}_i^p r^p$, where $\tilde{\alpha}_i^p$ is the limited wave strength. This is calculated by applying some limiter function ϕ to the ratio of this wave strength to the strength of the neighboring wave in the same family, looking in the upwind direction,

$$\tilde{\alpha}_i^p = \phi(\theta_i^p) \alpha_i^p,$$

where

$$\theta_i^p = \frac{\alpha_i^p}{\alpha_{i-1}^p} \quad \text{with } I = \begin{cases} i-1 & \text{if } \lambda_i^p > 0 \\ i+1 & \text{if } \lambda_i^p \leq 0. \end{cases} \quad (22)$$

The ratio of wave strengths θ_i^p is used to measure the smoothness of the solution. Where the solution is smooth, this can be expected to be near 1. Near discontinuities in the p th family, θ_i^p may be far from 1. A wide variety of limiter functions have been studied. Some standard limiters used here are

$$\text{minmod: } \phi(\theta) = \max(0, \min(1, \theta))$$

$$\text{superbee: } \phi(\theta) = \max(0, \min(1, 2\theta), \min(2, \theta))$$

monotonized centered

$$\text{(MC): } \phi(\theta) = \max(0, \min((1 + \theta)/2, 2, 2\theta)).$$

For variable coefficient or nonlinear problems the wave \mathcal{W}_i^p will not be a scalar multiple of the waves \mathcal{W}_{i-1}^p or \mathcal{W}_{i+1}^p from the neighboring Riemann problem, and one must determine the manner in which these vectors are going to be compared and modified in applying the limiter.

For concreteness assume $\lambda_i^p > 0$ so that in the p th family we compare \mathcal{W}_i^p to \mathcal{W}_{i-1}^p . The approach used here (which is currently the default of CLAWPACK) is to project the neighboring wave \mathcal{W}_{i-1}^p onto the vector \mathcal{W}_i^p and compare the length of this projected vector with the length of \mathcal{W}_i^p itself, modifying the length of \mathcal{W}_i^p as needed, but preserving its direction. This is accomplished by setting

$$\theta_i^p = \frac{\mathcal{W}_{i-1}^p \cdot \mathcal{W}_i^p}{\mathcal{W}_i^p \cdot \mathcal{W}_i^p}$$

$$\tilde{\mathcal{W}}_i^p = \phi(\theta_i^p) \mathcal{W}_i^p,$$

where \cdot represents inner product. Note that this reduces to (22) for a linear system.

A simpler approach would be to apply a scalar limiter componentwise to each element of the vector \mathcal{W}_i^p and the corresponding element of \mathcal{W}_{i-1}^p . This also reduces to (22) for a linear system but typically would not preserve the direction of \mathcal{W}_i^p in a nonlinear problem. It is not clear at this point how important this preservation is. (One comparison is presented in [33].)

Another approach is taken in recent work of Liu and Lax [26, 40]. They use the Roe matrix A_i that determines $\mathcal{W}_i^p = \alpha_i^p r_i^p$, via an eigen-decomposition of $\Delta q_i = q_i - q_{i-1}$, to also decompose $\Delta q_{i-1} = \sum_s \beta^s r_i^s$. Then the limiter is applied to α_i^p and β^p to obtain $\tilde{\alpha}_i^p$. This requires additional work but has the advantage that their resulting scheme can be proved to be a positive scheme in the sense described in [40].

After defining the limited waves $\tilde{\mathcal{W}}_i^p$ by one of the approaches above, the \tilde{F}_i used in (18) are given by (19) with \mathcal{W}_i^p replaced by $\tilde{\mathcal{W}}_i^p$.

2.6. Numerical Results for Acoustics

As an example to demonstrate that this generalization of one-dimensional high resolution methods to nonconservative systems is effective, consider the acoustics equations (13) in the extreme case where the sound speed is discontinuous at an interface between two media. We take $K \equiv 1$ and

$$\rho = \begin{cases} 1 & \text{if } x < 0.6 \\ 4 & \text{if } x > 0.6, \end{cases}$$

giving a jump from $c = 1$ on the left to $c = 0.5$ on the right. As initial data we take a hump in pressure in the left region and $u \equiv 0$. The hump splits into equal left-going and right-going pieces and the right-going portion hits the interface, giving transmitted and reflected waves. The transmitted wave is narrower due to the lower sound speed.

As an initial hump we use

$$p(x, 0) = \begin{cases} \bar{p} \sqrt{1 - ((x - x_0)/\bar{x})^2} & \text{if } |x - x_0| < \bar{x}, \\ 0 & \text{otherwise,} \end{cases}$$

with $x_0 = 0.4$, $\bar{x} = 0.075$, and $\bar{p} = 0.2$. This is a half ellipse of the type used by Zalesak [62] in his comparison of advection algorithms. This is a nice test because it has an infinite slope at the corners as well as a region of smoothness.

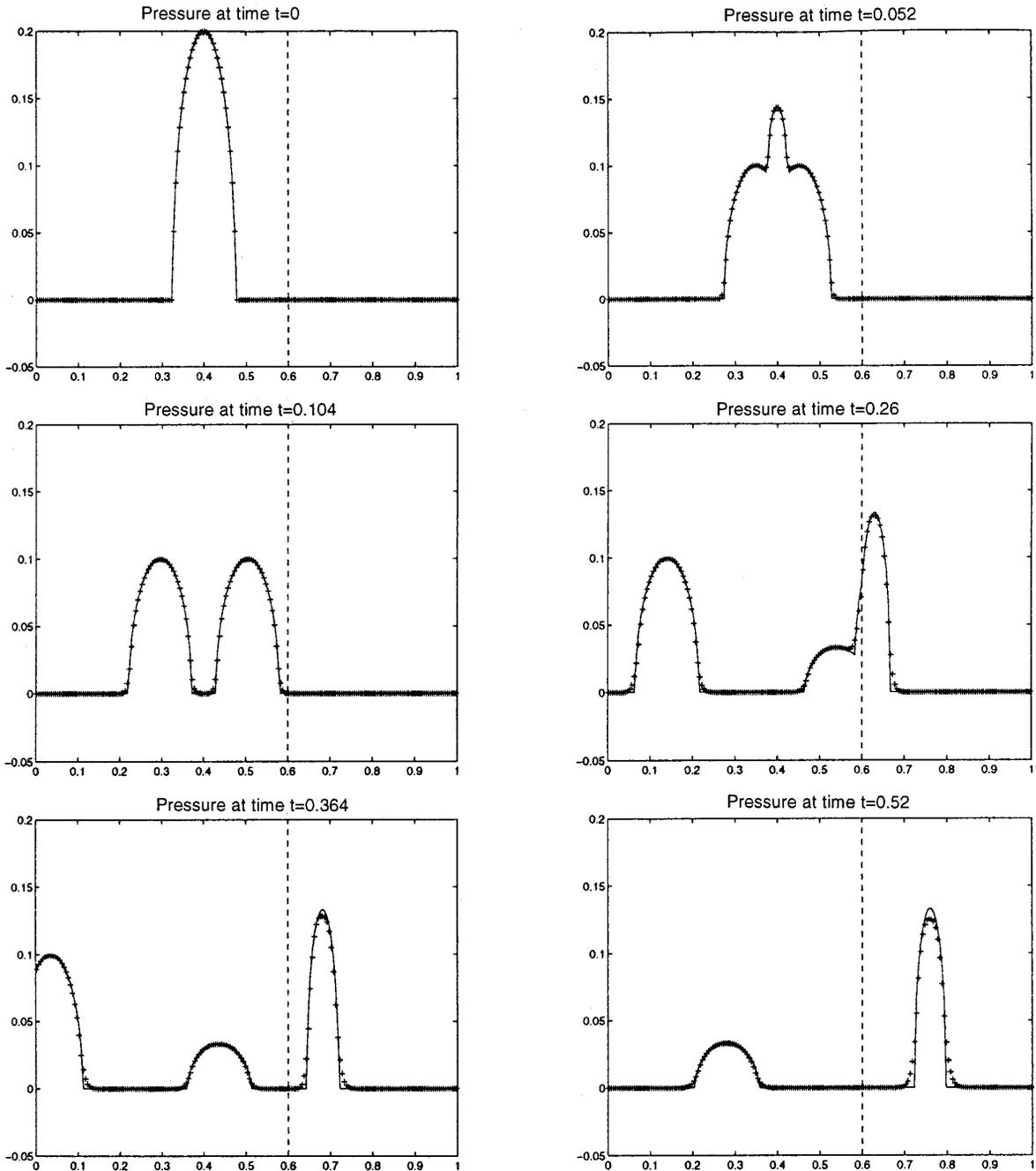


FIG. 3. Computed Results for One-Dimensional Acoustics with a Discontinuity in Sound Speed at $x = 0.6$.

Figure 3 shows the true solution and computed results for the pressure at six different times on a grid with 200 grid points, $\Delta x = 0.005$ and $\Delta t = 0.004$, corresponding to a Courant number of 0.8 in the left medium, and 0.4 in the right.

Note that the left-going hump leaves the domain cleanly. Zero-order extrapolation was used at the boundary, as discussed in [29]. It is also clear that there are no overshoots

or undershoots, even as the wave passes through the interface. In these tests the minmod limiter was used.

2.7. Capacity-Form Differencing

In many applications it is useful to allow a generalized form of the conservation law,

$$\kappa(x)q_t + f(q)_x = 0, \quad (23)$$

in one space dimension, or

$$\kappa(x, y)q_t + f(q)_x + g(q)_y = 0, \quad (24)$$

in two dimensions (or similar generalizations of the quasilinear equation (2)).

Here κ is a given function of space that I will refer to in general as a *capacity function*, since it generally represents, in some sense, the capacity of the medium at each point to hold the conserved quantity q . (In some applications κ may also vary with time.) In one dimension, the corresponding integral form is

$$\frac{\partial}{\partial t} \int_a^b q(x, t) \kappa(x) dx = f(q(a, t)) - f(q(b, t)).$$

Note that this integral can be viewed as the integral of q against the measure $\kappa(x) dx$.

EXAMPLE 2.7.1. In porous media flow κ could represent the *porosity*, the fraction of the volume that is available for the fluid to occupy. Consider, for example, a porous medium in which the porosity and permeability vary only in one direction, with x , say, and suppose we consider flow in the x -direction, with zero velocity in the other directions. If the porous medium is saturated with an incompressible fluid, then we have the advection equation

$$\kappa(x)q_t + u_0q_x = 0, \quad (25)$$

where u_0 is constant for an incompressible fluid. In this case, $\kappa(x) dx$ is the infinitesimal volume element available to the fluid at point x . The advection equation (25) would also be obtained by modeling incompressible flow in a “quasi one-dimensional” pipe with variable cross-sectional area $\kappa(x)$ and a velocity that is assumed to vary only with x .

In two space dimensions we would have an equation of the form

$$\kappa(x, y)q_t + u(x, y)q_x + v(x, y)q_y = 0, \quad (26)$$

where $u_x + v_y = 0$. Applications of these algorithms to saturated groundwater flow are discussed in [2, 29]. See also [1]. See Example 3.9.1 for a related example of density-stratified flow.

EXAMPLE 2.7.2. Suppose we wish to solve the equation $q_t + f(q)_x = 0$ on a stretched grid, with grid points

$$x_i = X(\xi_i),$$

where $X(\xi)$ is some smooth grid mapping function and $\xi_i = i\Delta\xi$ is a uniform computational grid. If we let

$\bar{q}(\xi, t) = q(X(\xi), t)$, then the conservation law can be rewritten as

$$X'(\xi)\bar{q}_t + f(\bar{q})_\xi = 0$$

and solved on the uniform grid in ξ -space. In this case $X'(\xi)$ is the capacity function. Note that

$$X'(\xi_i)\Delta\xi \approx x_{i+1} - x_i,$$

the length of the i th cell in physical space, so again the notion of capacity makes sense.

The ability to handle mapped grids is more interesting in two space dimensions, in which case the capacity function is the Jacobian of the transformation. General curvilinear grids in two dimensions are discussed in Section 3.10.

In each of the examples above, it is κq that is really the conserved quantity although q may be of more physical significance. One approach to solving (23) would be to manipulate it to the form

$$q_t + (f(q)/\kappa)_x = \left(\frac{\kappa'}{\kappa}\right)f(q), \quad (27)$$

which is a standard conservation law with a source term. Solving in this form may not guarantee conservation of κq , however, and has the additional problem of introducing an unnecessary source term.

Another approach might be to define the new variable $\psi(x, t) = \kappa(x)q(x, t)$ and then to solve the conservation law

$$\psi_t + f(\psi/\kappa)_x = 0$$

for ψ , dividing by κ at the end to recover q . This would guarantee conservation of κq but often has other difficulties such as the inability to preserve uniform states in q .

Instead of reducing Eq. (23) to a more familiar conservation law by one of the above devices, it is preferable to apply *capacity-form differencing*,

$$\bar{q}_i = q_i - \frac{\Delta t}{\kappa_i \Delta x} (\mathcal{A}^+ \Delta q_i + \mathcal{A}^- \Delta q_{i+1}) - \frac{\Delta t}{\kappa_i \Delta x} (\tilde{F}_{i+1} - \tilde{F}_i), \quad (28)$$

where κ_i is the capacity of the i th cell. This is a simple extension of (18) which ensures that $\sum \kappa_i q_i$ is conserved (except for fluxes through the boundaries) and yet allows the Riemann solution to be computed based on q as in the case $\kappa \equiv 1$. In particular, if $q_i \equiv \text{constant}$, then typically the fluctuations and also the waves used to define the \tilde{F} are all zero, so that $\bar{q}_i = q_i$ and constant data is preserved.

The formulas for the second-order correction terms also

need to be modified in a simple manner to take into account the function κ . The second-order correction term (19) is replaced by

$$\tilde{F}_i = \frac{1}{2} \sum_p^{M_w} |\lambda_i^p| \left(1 - \frac{\Delta t}{\kappa_{i-1/2} \Delta x} |\lambda_i^p| \right) \tilde{\mathcal{W}}_i^p, \quad (29)$$

where $\kappa_{i-1/2}$ is some average value of κ ; e.g., $\kappa_{i-1/2} = \frac{1}{2}(\kappa_{i-1} + \kappa_i)$.

Note that in the case of a mapped grid, Example 2.7.2, $\kappa_{i-1/2} \Delta \xi$ is roughly the distance between the centers of cells $i-1$ and i in physical space, and so $\sum_p \tilde{\mathcal{W}}_i^p / (\kappa_{i-1/2} \Delta \xi)$ is approximately q_x .

This form of differencing is frequently used in practice in various contexts, but it is not usually presented in the general framework used here, to the best of my knowledge. It is a simple yet powerful generalization that allows application of finite volume methods to a wider variety of problems and should be better known.

Sample calculations with capacity-form differencing are given for a more interesting two-dimensional problem in Sections 3.9 and 3.10.

3. TWO SPACE DIMENSIONS

In two space dimensions we have the standard conservation law

$$q_t + f(q)_x + g(q)_y = 0. \quad (30)$$

We first consider the extension of the wave-propagation algorithm to multiple dimensions for this case and then extend to nonconservative hyperbolic systems in Section 3.8. Capacity functions can also be introduced, as in one dimension (see Section 3.9).

We discretize using a Cartesian grid with uniform spacing Δx and Δy . The cell average over cell (i, j) is denoted by q_{ij} . The standard flux-differencing form of a conservative finite volume method is

$$\bar{q}_{ij} = q_{ij} - \frac{\Delta t}{\Delta x} (F_{i+1,j} - F_{ij}) - \frac{\Delta t}{\Delta y} (G_{i,j+1} - G_{ij}), \quad (31)$$

where F_{ij} is the numerical flux at the left edge of cell (i, j) and G_{ij} is the flux below this cell, e.g.,

$$F_{ij} \approx \frac{1}{\Delta y \Delta t} \int_{t_n}^{t_{n+1}} \int_{y_j}^{y_{j+1}} f(q(x_i, y, t)) dy dt, \quad (32)$$

where (x_i, y_j) is the coordinate of the lower left corner of cell (i, j) .

The multidimensional wave propagation algorithm developed here could be written in this form when applied

to a standard conservation law (30), but, again, it is described (and implemented) in a more general form which allows easier application to other hyperbolic problems that are not in conservation form. The method is implemented as

$$\begin{aligned} \bar{q}_{ij} = & q_{ij} + \Delta_{ij}^{\text{up}} - \frac{\Delta t}{\Delta x} (\tilde{F}_{i+1,j} - \tilde{F}_{ij}) \\ & - \frac{\Delta t}{\Delta y} (\tilde{G}_{i,j+1} - \tilde{G}_{ij}), \end{aligned} \quad (33)$$

where Δ_{ij}^{up} is the update for a first-order upwind (donor-cell or Godunov) method, of the form

$$\begin{aligned} \Delta_{ij}^{\text{up}} = & \frac{\Delta t}{\Delta x} (\mathcal{A}^- \Delta q_{i+1,j} + \mathcal{A}^+ \Delta q_{ij}) \\ & - \frac{\Delta t}{\Delta y} (\mathcal{B}^- \Delta q_{i,j+1} + \mathcal{B}^+ \Delta q_{ij}). \end{aligned} \quad (34)$$

The $\mathcal{A}^\pm \Delta q$ and $\mathcal{B}^\pm \Delta q$ terms represent fluctuations arising from Riemann problems in the x - and y -directions, respectively. The \tilde{F} and \tilde{G} fluxes are used to perform second-order corrections and, also, corrections for cross-derivative terms that arise in two dimensions which did not appear in one dimension.

The descriptions below will focus primarily on the solution of a Riemann problem in the x -direction, at an interface between cells $(i-1, j)$ and (i, j) , and the manner in which the waves from this Riemann problem contribute to Δ_{ij}^{up} , \tilde{F}_{ij} , and nearby \tilde{G} fluxes. An analogous procedure is followed at each interface in the y -direction between cells $(i, j-1)$ and (i, j) , with a switch in the roles of F and G and in $\mathcal{A} \Delta q$ and $\mathcal{B} \Delta q$. The symbol Δq_{ij} below thus refers to $q_{ij} - q_{i-1,j}$, the difference in the x -direction.

3.1. First-Order Godunov

We begin by solving a one-dimensional Riemann problem normal to each cell interface, exactly as in one space dimension. We solve the one-dimensional Riemann problem $q_t + f(q)_x = 0$ with data $q_{i-1,j}$ and q_{ij} . This results in a set of M_w waves and speeds, along with a splitting of the flux difference $f(q_{ij}) - f(q_{i-1,j})$ into two pieces $\mathcal{A}^- \Delta q_{ij}$ and $\mathcal{A}^+ \Delta q_{ij}$ moving to the left and right, respectively.

A basic first-order Godunov method is thus defined by simple extension from one-dimension via (33) and (34) with $\tilde{F} = \tilde{G} = 0$. This method is typically stable only for Courant numbers up to $\frac{1}{2}$ (see Section 3.7).

3.2. Transverse Propagation

The Godunov method described above is based on propagating waves normal to each cell interface. In reality the waves should propagate in a multidimensional manner and

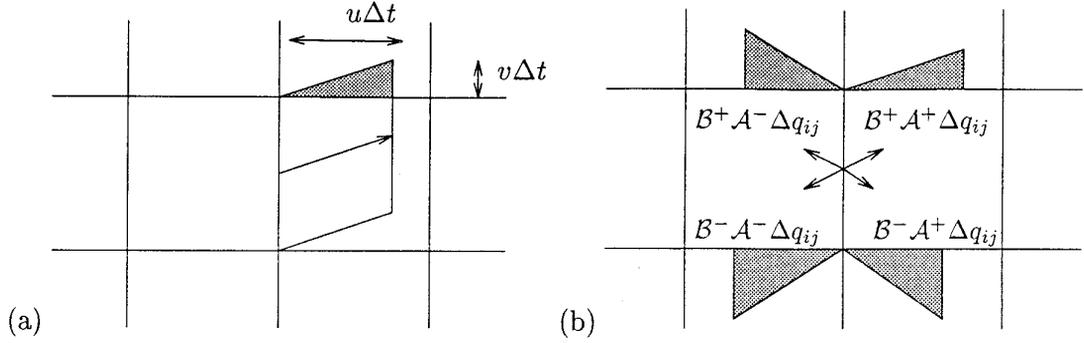


FIG. 4. (a) Transverse propagation in the advection equation. (b) The four transverse flux differences for a general system of equations.

affect other cell averages besides those adjacent to the interface. This is accomplished by splitting each fluctuation $\mathcal{A}^* \Delta q_{ij}$ for $*$ = + and - into two *transverse fluctuations* which will be called $\mathcal{B}^+ \mathcal{A}^* \Delta q_{ij}$ (the up-going transverse fluctuation) and $\mathcal{B}^- \mathcal{A}^* \Delta q_{ij}$ (the down-going transverse fluctuation). The notation is motivated by the linear system case mentioned further below.

Figure 4a shows an example for the advection equation

$$q_t + uq_x + vq_y = 0 \quad (35)$$

with velocities $u, v > 0$, in which case the single wave should propagate in the direction (u, v) . There is a triangular portion of the wave which should move into cell $(i, j + 1)$, rather than cell (i, j) in this figure. This can be accomplished by modifying the flux $G_{i,j+1}$ at the interface between these two cells by the appropriate amount,

$$\tilde{G}_{i,j+1} := \tilde{G}_{i,j+1} - \frac{1}{2} \frac{\Delta t}{\Delta x} uv (q_{ij} - q_{i-1,j}).$$

This is discussed in much greater detail for the advection equation in [35].

Introducing transverse propagation has two important effects. First, it provides the cross-derivative terms q_{xy} and q_{yx} required in a second-order accurate algorithm. Once the transverse flux differences have been included, second-order accuracy is easily achieved by including the second-derivative terms in each coordinate direction (q_{xx} and q_{yy}) using the same correction that is applied in one space dimension. See Section 3.3 for these terms, [35] for an analysis of the truncation error in the case of linear advection, and [33] for some discussion of systems. Second, the transverse correction terms improve the stability limit and allow full Courant number 1, relative to the maximum wave speed in any direction. This is discussed in Section 3.7.

To generalize the notion of transverse propagation to a system of equations, we begin by observing that, for advection, $u(q_{ij} - q_{i-1,j})$ is the right-going fluctuation

(which is the entire fluctuation in this scalar case) and that this should be propagated upward by the vertical velocity v . The quantity $vu(q_{ij} - q_{i-1,j})$, the product of this fluctuation and the vertical velocity, gives the up-going transverse fluctuation.

For a system of equations we typically will have both a left-going fluctuation $\mathcal{A}^- \Delta q_{ij}$ and a right-going fluctuation $\mathcal{A}^+ \Delta q_{ij}$. Each of these will be split into an up-going piece and a down-going piece, so there will be four transverse fluctuations modifying the four neighboring G fluxes as indicated in Fig. 4b.

Again the notation $\mathcal{B}^\pm \mathcal{A}^* \Delta q_{ij}$ is motivated by the case of a linear system of equations

$$q_t + Aq_x + Bq_y = 0 \quad (36)$$

in which case the matrices B^\pm are defined in an analogous way to A^\pm , based on the positive and negative eigenvalues of B . The transverse fluctuations are then given by

$$\mathcal{B}^\pm \mathcal{A}^* \Delta q_{ij} = B^\pm A^* (q_{ij} - q_{i-1,j}),$$

where $*$ = + or -.

In general, the transverse fluctuations are used to modify the four neighboring fluxes according to

$$\begin{aligned} \tilde{G}_{i,j+1} &:= \tilde{G}_{i,j+1} - \frac{1}{2} \frac{\Delta t}{\Delta x} \mathcal{B}^+ \mathcal{A}^+ \Delta q_{ij} \\ \tilde{G}_{ij} &:= \tilde{G}_{ij} - \frac{1}{2} \frac{\Delta t}{\Delta x} \mathcal{B}^- \mathcal{A}^+ \Delta q_{ij} \\ \tilde{G}_{i-1,j+1} &:= \tilde{G}_{i-1,j+1} - \frac{1}{2} \frac{\Delta t}{\Delta x} \mathcal{B}^+ \mathcal{A}^- \Delta q_{ij} \\ \tilde{G}_{i-1,j} &:= \tilde{G}_{i-1,j} - \frac{1}{2} \frac{\Delta t}{\Delta x} \mathcal{B}^- \mathcal{A}^- \Delta q_{ij}. \end{aligned} \quad (37)$$

For a linear system $q_t + Aq_x + Bq_y = 0$, the sum of all the transverse fluctuations is

$$(B^+A^+ + B^-A^+ + B^+A^- + B^-A^-)\Delta q = BA\Delta q. \quad (38)$$

Since Δq represents a difference in the x -direction, and the G -fluxes are then differenced in the y -direction in updating q , the modification (37) results in an approximation to $\frac{1}{2}\Delta t BAq_{xy}$, which is one of the cross-derivative terms needed in order to achieve second-order accuracy. The splitting of this term into four pieces based on the signs of the eigenvalues gives an upwinding of these terms that substantially improves stability over the centered Lax–Wendroff approach, which can be shown to correspond to using $\frac{1}{4}BA\Delta q$ in each of the updates in (37) instead of the splitting (38). Radvogin [41] defines methods for linear systems that are very similar.

For a linear system of equations, in the special case where A and B are simultaneously diagonalizable (i.e., have the same eigenvectors), the system can be transformed to a set of m independent scalar advection equations and the modifications (37) are equivalent to applying the advection algorithm to each of these independent equations.

Even when A and B are not simultaneously diagonalizable, these transverse fluctuations can also be interpreted in terms of multidimensional wave propagation. The flux difference $\mathcal{A}^+\Delta q$, for example, is the sum of $\lambda^p \mathcal{W}^p$ over all right-going waves. If we now decompose each wave \mathcal{W}^p as a linear combination of the eigenvectors of B , say

$$\mathcal{W}^p = \sum_{s=1}^m \beta^{ps} w^s,$$

where $Bw^s = \mu^s w^s$, then the subwave $\beta^{ps} w^s$ should be propagated upwards or downwards with speed μ^s , depending on whether μ^s is positive or negative. The up-going fluctuation, for example, is

$$\sum_s (\mu^s)^+ \beta^{ps} w^s.$$

Summing these over all right-going waves \mathcal{W}^p for $\lambda^p > 0$ gives

$$\sum_p (\lambda^p)^+ \sum_s (\mu^s)^+ \beta^{ps} w^s = \sum_s (\mu^s)^+ \left(\sum_p (\lambda^p)^+ \beta^{ps} \right) w^s$$

and it can be easily verified that this is precisely $B^+A^+\Delta q$. The other transverse flux differences can be interpreted similarly as combinations of waves in the other three directions.

Examples are given below to illustrate these transverse splittings more concretely. The two-dimensional acoustics equations are discussed in Section 3.5 and the acoustics equations with varying material parameters are discussed in Example 3.8.1. In the latter case the corrections due to transverse propagation can still be written in flux-differencing form, even though the system is not conservative.

For a nonlinear system of equations, we must still specify how the fluctuations $\mathcal{A}^*\Delta q$ ($*$ = + or -) defined by solving the Riemann problem normal to each interface will be split up into transverse fluctuations. This is typically done by splitting the vector $\mathcal{A}^*\Delta q$ into eigenvectors of an approximate Jacobian matrix in the transverse direction. If the Roe approximation is used, for example, then we have an approximate Jacobian $A \approx f'(q)$ defined at the interface between cells $(i-1, j)$ and (i, j) that is based on averaged values between these two states. The same averaged value can be used to define an approximate Jacobian $B \approx g'(q)$ at this interface. (See Example 3.6.1 for a concrete example.) The eigenvectors w^s of B can be used to split $\mathcal{A}^*\Delta q_{ij}$ just as in the linear case,

$$\mathcal{A}^*\Delta q_{ij} = \sum_s \beta^s w^s.$$

We then define

$$\mathcal{B}^- \mathcal{A}^*\Delta q_{ij} = \sum_s (\mu^s)^- \beta^s w^s, \quad \mathcal{B}^+ \mathcal{A}^*\Delta q_{ij} = \sum_s (\mu^s)^+ \beta^s w^s,$$

where μ^s are the corresponding eigenvalues of B . Note that, as for the linear system, this splitting can also be written as

$$\mathcal{B}^- \mathcal{A}^*\Delta q_{ij} = B^-(\mathcal{A}^*\Delta q_{ij}), \quad \mathcal{B}^+ \mathcal{A}^*\Delta q_{ij} = B^+(\mathcal{A}^*\Delta q_{ij}),$$

where, again, B is the approximate Jacobian.

Although still only first-order accurate, the inclusion of transverse flux differences typically improves stability and allows Courant numbers up to 1 (see Section 3.7). These terms are also a necessary component of the second-order accurate method, since they yield the cross derivative terms that arise in the $(\Delta t)^2$ term of the Taylor series expansion.

3.3. Second-Order Corrections

Once the transverse corrections described above have been implemented, it is possible to achieve second-order accuracy by simply making one-dimensional flux correc-

tions analogous to (19). It is at this point that the waves \mathcal{W}_{ij}^p and speeds λ_{ij}^p computed in solving the Riemann problem normal to each interface are used. We make the corrections

$$\tilde{F}_{ij} := \tilde{F}_{ij} + \frac{1}{2} \sum_p^{M_w} |\lambda_{ij}^p| \left(1 - \frac{\Delta t}{\Delta x} |\lambda_{ij}^p| \right) \tilde{\mathcal{W}}_{ij}^p, \quad (39)$$

where $\tilde{\mathcal{W}}_{ij}^p$ is a limited version of \mathcal{W}_{ij}^p . The limiter is applied exactly as in one dimension, so $\tilde{\mathcal{W}}_{ij}^p$ is compared to $\mathcal{W}_{i-1,j}^p$ or $\mathcal{W}_{i+1,j}^p$, depending on whether $\lambda_{ij}^p > 0$ or < 0 , just as described in Section 2.4.

This is a potential weak point in the algorithm, since the limiter does not take into account the behavior of the solution in the transverse direction. A more sophisticated multidimensional limiting procedure (e.g., [27, 46, 61]) might be able to control oscillations in multidimensional problems. However, in practice this one-dimensional approach to limiting seems to work very well for most problems and is much simpler to implement and less computationally intensive than other approaches.

3.4. Transverse Propagation of the Second-Order Corrections

The above method is already second-order accurate (for smooth solutions), but it is quite easy to also propagate the second-order corrections in the transverse direction. This is motivated by ‘‘Method 4’’ in [35] and, while this additional correction does not increase the order of accuracy, it has been found to improve stability properties and reduce spurious oscillations in many problems (an example is given in [35]).

The flux corrections (39) will affect the cell averages $q_{i-1,j}$ and q_{ij} in the cells to the left and right of this interface. Hence the transverse propagation of this correction should affect four G fluxes, those below and above these two cells. The corrections are split into up-going and down-going portions in exactly the same manner as $\mathcal{A}^* \Delta q$ is split into $\mathcal{B}^+ \mathcal{A}^* \Delta q_{ij}$ and $\mathcal{B}^- \mathcal{A}^* \Delta q_{ij}$. In fact, the algorithm with this transverse propagation is implemented by simply modifying $\mathcal{A}^* \Delta q$ by these second-order corrections before calling the routine that splits these vectors into transverse fluctuations. The proper modifications are

$$\begin{aligned} \mathcal{A}^+ \Delta q_{ij} &:= \mathcal{A}^+ \Delta q_{ij} - \sum_p^{M_w} |\lambda_{ij}^p| \left(1 - \frac{\Delta t}{\Delta x} |\lambda_{ij}^p| \right) \tilde{\mathcal{W}}_{ij}^p, \\ \mathcal{A}^- \Delta q_{ij} &:= \mathcal{A}^- \Delta q_{ij} + \sum_p^{M_w} |\lambda_{ij}^p| \left(1 - \frac{\Delta t}{\Delta x} |\lambda_{ij}^p| \right) \tilde{\mathcal{W}}_{ij}^p. \end{aligned}$$

Note that the correction term in these updates is exactly the same as the term that modifies \tilde{F}_{ij} in (39), so this requires

virtually no additional work. This has been used in all the numerical results presented below.

3.5. Acoustics

In two space dimensions the acoustics equations for the pressure perturbation p and velocities u and v can be written

$$q_t + Aq_x + Bq_y = 0, \quad (40)$$

where

$$q = \begin{bmatrix} p \\ u \\ v \end{bmatrix}, \quad A = \begin{bmatrix} 0 & K & 0 \\ 1/\rho & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & K \\ 0 & 0 & 0 \\ 1/\rho & 0 & 0 \end{bmatrix}.$$

The algorithm will be described for the x direction. Analogous formulas hold for y sweeps with the role of u and v switched.

The Riemann solution for $q_t + Aq_x = 0$ between states $q_{i-1,j}$ and q_{ij} consists of three waves, but one always has speed zero and can be ignored. So we can take $M_w = 2$ and only use two waves, $\alpha^1 r^1$ with speed $\lambda^1 = -c$ and $\alpha^2 r^2$ with speed $\lambda^2 = +c$. The eigenvectors are

$$r^1 = \begin{bmatrix} c \\ 1/\rho \\ 0 \end{bmatrix}, \quad r^2 = \begin{bmatrix} -c \\ 1/\rho \\ 0 \end{bmatrix}. \quad (41)$$

The coefficients are the same as in one dimension,

$$\alpha^1 = \frac{1}{2}(-\Delta q^1/c + \rho \Delta q^2), \quad \alpha^2 = \frac{1}{2}(\Delta q^1/c + \rho \Delta q^2). \quad (42)$$

Again the fluctuations are

$$\mathcal{A}^- \Delta q = \lambda^1 \alpha^1 r^1, \quad \mathcal{A}^+ \Delta q = \lambda^2 \alpha^2 r^2. \quad (43)$$

To obtain the transverse fluctuations we split $\mathcal{A}^* \Delta q_{ij}$ into eigenvectors of the matrix B . For the acoustics equations these are particularly simple. For example, $\mathcal{A}^- \Delta q$ would be split as

$$\mathcal{A}^- \Delta q = -c \alpha^1 \begin{bmatrix} -c \\ 1/\rho \\ 0 \end{bmatrix} = \beta^1 \begin{bmatrix} -c \\ 0 \\ 1/\rho \end{bmatrix} + \beta^2 \begin{bmatrix} c \\ 0 \\ 1/\rho \end{bmatrix} + \beta^3 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$

where

$$\beta^1 = -c \alpha^1/2, \quad \beta^2 = c \alpha^1/2, \quad \beta^3 = -c \alpha^1/\rho.$$

The three eigenvectors of B displayed above correspond to eigenvalues $\mu^1 = -c$, $\mu^2 = c$, $\mu^3 = 0$, and the third one plays no role in the transverse propagation. The transverse fluctuations are then

$$\mathcal{B}^- \mathcal{A}^- \Delta q_{ij} = -c\beta^1 \begin{bmatrix} -c \\ 0 \\ 1/\rho \end{bmatrix}, \quad \mathcal{B}^+ \mathcal{A}^- \Delta q_{ij} = c\beta^2 \begin{bmatrix} c \\ 0 \\ 1/\rho \end{bmatrix}.$$

Some numerical examples for two-dimensional acoustics (in a heterogeneous material) are given in Section 3.8 below.

3.6. Gas Dynamics

First we summarize the way in which the waves, fluctuations, and transverse fluctuations are computed for the case of isothermal flow. Numerical results will then be presented for the full Euler equations.

EXAMPLE 3.6.1. Isothermal flow is governed by a nonlinear system of equations quite similar to the full Euler equations of compressible flow, but simpler, as it involves only three equations instead of four and it involves a very simple equation of state, $p = c^2\rho$, where c is the sound speed. See [32] for more discussion. Considering this case should be sufficient to illuminate the main features of the splittings. (The splittings for the full Euler equations can be found in [42] and in examples included with the CLAWPACK software.)

In two space dimensions the isothermal equations take the form

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u \\ \rho v \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho u \\ \rho uv + c^2 u \\ \rho uv \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + c^2 v \end{bmatrix} = 0. \quad (44)$$

Here c is the constant sound speed. Suppose that we use the Roe approximate Riemann solver to solve the Riemann problem normal to each cell face. The Roe solver between states $q_{i-1,j}$ and q_{ij} , for example, is based on averaged states

$$\bar{u} = \frac{\sqrt{\rho_{i-1,j}} u_{i-1,j} + \sqrt{\rho_{i,j}} u_{i,j}}{\sqrt{\rho_{i-1,j}} + \sqrt{\rho_{i,j}}}, \quad \bar{v} = \frac{\sqrt{\rho_{i-1,j}} v_{i-1,j} + \sqrt{\rho_{i,j}} v_{i,j}}{\sqrt{\rho_{i-1,j}} + \sqrt{\rho_{i,j}}}. \quad (45)$$

The Roe matrix $A = A_{ij}$ is simply the Jacobian matrix $f'(\bar{q})$ based on these averaged states:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -\bar{u}^2 + c & 2\bar{u} & 0 \\ -\bar{u}\bar{v} & \bar{v} & \bar{u} \end{bmatrix}.$$

(Note that the superscripts are exponents here.) We can write $A = R\Lambda R^{-1}$, where

$$R = \begin{bmatrix} 1 & 0 & 1 \\ \bar{u} - c & 0 & \bar{u} + c \\ \bar{v} & 1 & \bar{v} \end{bmatrix}, \quad R^{-1} = \frac{1}{2c} \begin{bmatrix} \bar{u} + c & -1 & 0 \\ -2c\bar{v} & 0 & 2c \\ -\bar{u} + c & 1 & 0 \end{bmatrix}.$$

The waves required for the second-order corrections are then

$$\mathcal{W}_i^1 = \alpha_i^1 \begin{bmatrix} 1 \\ \bar{u} - c \\ \bar{v} \end{bmatrix}, \quad \mathcal{W}_i^2 = \alpha_i^2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathcal{W}_i^3 = \alpha_i^3 \begin{bmatrix} 1 \\ \bar{u} - c \\ \bar{v} \end{bmatrix}, \quad (46)$$

where the wave strengths α_i are given by

$$\alpha_i = R^{-1} \Delta q_{ij}. \quad (47)$$

The wave speeds are the eigenvalues of A :

$$\lambda_i^1 = \bar{u} - c, \quad \lambda_i^2 = \bar{u}, \quad \lambda_i^3 = \bar{u} + c. \quad (48)$$

The fluctuations are

$$\mathcal{A}^\pm \Delta q_{ij} = \sum_{p=1}^3 (\lambda_i^p)^\pm \mathcal{W}_i^p,$$

perhaps modified by an entropy fix as discussed in [29, 32].

To define the transverse fluctuations, we use the same average states from $q_{i-1,j}$ and q_{ij} to define the matrix $B = g'(\bar{q})$:

$$B = \begin{bmatrix} 0 & 0 & 1 \\ -\bar{u}\bar{v} & \bar{v} & \bar{u} \\ -\bar{v}^2 + c^2 & 0 & 2\bar{v} \end{bmatrix}.$$

We have the eigen-decomposition $B = TMT^{-1}$, where

$$T = \begin{bmatrix} 1 & 0 & 1 \\ \bar{u} & 1 & \bar{u} \\ \bar{v} - c & 0 & \bar{v} + c \end{bmatrix}, \quad T^{-1} = \frac{1}{2c} \begin{bmatrix} \bar{v} + c & 0 & -1 \\ -2c\bar{u} & 2c & 0 \\ -\bar{v} + c & 0 & 1 \end{bmatrix}$$

and the eigenvalues (transverse wave speeds) are

$$\mu_i^1 = \bar{v} - c, \quad \mu_i^2 = \bar{v}, \quad \mu_i^3 = \bar{v} + c.$$

We then decompose $\mathcal{A}^-\Delta q_{ij}$ and $\mathcal{A}^+\Delta q_{ij}$ into eigenvectors of this matrix, which gives transverse waves. The up-going transverse fluctuations are obtained by splitting $\mathcal{A}^+\Delta q_{ij}$ into eigenvectors of B :

$$\mathcal{F}^1 = \beta^1 \begin{bmatrix} 1 \\ \bar{u} \\ \bar{v} - c \end{bmatrix}, \quad \mathcal{F}^2 = \beta^2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathcal{F}^3 = \beta^3 \begin{bmatrix} 1 \\ \bar{u} \\ \bar{v} + c \end{bmatrix}, \quad (49)$$

where $\beta = T^{-1}(\mathcal{A}^+\Delta q_{ij})$ and then setting

$$\mathcal{B}^+ \mathcal{A}^* \Delta q_{ij} = \sum_{s=1}^3 (\mu_{ij}^s)^+ \mathcal{F}^s.$$

Similarly, to obtain the down-going transverse fluctuations we split $\mathcal{A}^-\Delta q_{ij}$ as in (49) but now with $\beta = T^{-1}(\mathcal{A}^-\Delta q_{ij})$ and then set

$$\mathcal{B}^- \mathcal{A}^* \Delta q_{ij} = \sum_{s=1}^3 (\mu_{ij}^s)^- \mathcal{F}^s.$$

Since much of the computational work of the Roe approximate Riemann solver is in the calculation of the averages (45) (even more so for the full Euler equations), the solution of the transverse Riemann problem is less expensive than the initial Riemann solution normal to the interface. Moreover, it does not appear necessary to apply any entropy fix in this step, further reducing the cost of the transverse splitting.

Note that in the two-dimensional algorithm, at each cell interface we must solve one normal Riemann problem and then do two transverse splittings, one of $\mathcal{A}^-\Delta q_{ij}$ and then one of $\mathcal{A}^+\Delta q_{ij}$. More implementation details can be found by looking at the examples in CLAWPACK.

EXAMPLE 3.6.2. We next verify second-order accuracy of the algorithm by solving the two-dimensional Euler equations for a gamma-law gas with radially symmetric smooth initial data. The results on a sequence of grids are compared to the “exact” solution obtained by solving the one-dimensional Euler equations with a source term for radial symmetry on a much finer grid. The initial data has zero velocity and

$$\rho(x, y, 0) = E(x, y, 0) = \begin{cases} 1 - 0.1(\cos(4\pi r) - 1) & \text{if } 0 < r < 0.5 \\ 1 & \text{if } r \geq 0.5, \end{cases}$$

where $r = \sqrt{x^2 + y^2}$. This data gives a single hump centered at $r = 0.25$. The problem is solved on the quarter plane section $[0, 1] \times [0, 1]$, with solid wall boundary conditions [29] along $x = 0$ and $y = 0$, which gives the same radial solution as would be obtained over a larger domain.

Figure 5 shows scatter plots of the two-dimensional solutions at time $t = 0.5$ on two different grids. The solid line is the “exact” density as a function of r , as computed by the one-dimensional code. The points show the N^2 values ρ_{ij} ($N = 20, 40$) on the two-dimensional grid, plotted against the distance each point is from the origin. This way of viewing the solution shows not only the pointwise error at each point, but also the radial symmetry of the computed solution, as seen by the lack of scatter from the true solution.

The table in Fig. 5 shows the computed error in each component in both the 1-norm and max-norm. The order of accuracy is estimated from the two finest grids. In these computations no limiters were used. Similar results in three space dimensions can be found in [23].

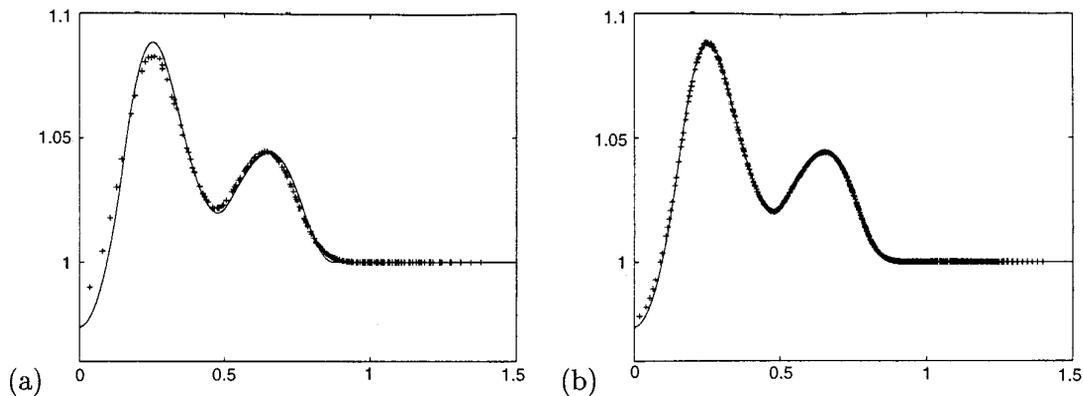
EXAMPLE 3.6.3. As an example to show how these algorithms perform on shock waves, we consider a two-dimensional Riemann problem of the type studied in [48, 49]. The data consists of four constant values in four quadrants chosen so that each pair of data gives a single shock wave in its solution. The interaction at the corner leads to a more complicated wave structure. This is a nice sample problem since the geometry and boundary conditions (extrapolation) are very simple. For other computations (see, e.g., [49, 26, 33]).

Some results on a 200×200 grid are shown in Fig. 6 for the case corresponding to Fig. 4b in [49]. The two computations shown here were identical, except for the choice of limiter functions. On the left the superbee limiter was used, whereas on the right the “monotonized centered” limiter was used. Both give equally sharp resolution of the primary shock waves, but have different behavior on the unstable slip lines. The superbee limiter, which is more compressive, gives sharper slip lines and, as a result, exhibits the rollup behavior expected from a Kelvin–Helmholz instability.

3.7. Stability

The multidimensional wave propagation algorithms described above appear to have very good stability properties for general systems of conservation laws, allowing Courant numbers up to 1. Here the Courant number CFL is defined by

$$CFL = \max(\lambda \Delta t / \Delta x, \mu \Delta t / \Delta y),$$



N	1-norm errors			max-norm errors		
	ρ	ρu	E	ρ	ρu	E
20	0.1472D-02	0.6395D-03	0.2099D-02	0.1224D-01	0.5259D-02	0.1726D-01
40	0.4152D-03	0.1989D-03	0.5872D-03	0.3337D-02	0.2642D-02	0.4646D-02
80	0.1155D-03	0.5252D-04	0.1624D-03	0.9391D-03	0.1060D-02	0.1227D-02
order	1.85	1.92	1.85	1.83	1.32	1.92

FIG. 5. Scatter plot of the computed density vs distance from origin for two-dimensional computations with smooth radially symmetric solution. Solution at $t = 0.5$, before solution has hit the boundary. (a) 20×20 grid. (b) 40×40 grid. The table shows the errors computed on three different grids, by comparing with the one-dimensional fine grid solution. The order of accuracy is estimated from the two finest grids. Left: scatter plot of density vs r on 20×20 grid. Right: scatter plot of density vs r on 40×40 grid.

with λ and μ being the maximum wave speeds in the x - and y -directions, respectively.

For the scalar advection equation this gives an improvement over the standard Lax–Wendroff method, for example, as illustrated in [35]. The improvement

arises from the upwind handling of the transverse fluxes, whereas Lax–Wendroff uses a centered average for these terms.

Similar improvement can be shown for linear hyperbolic systems using von Neumann analysis, as we now show. For

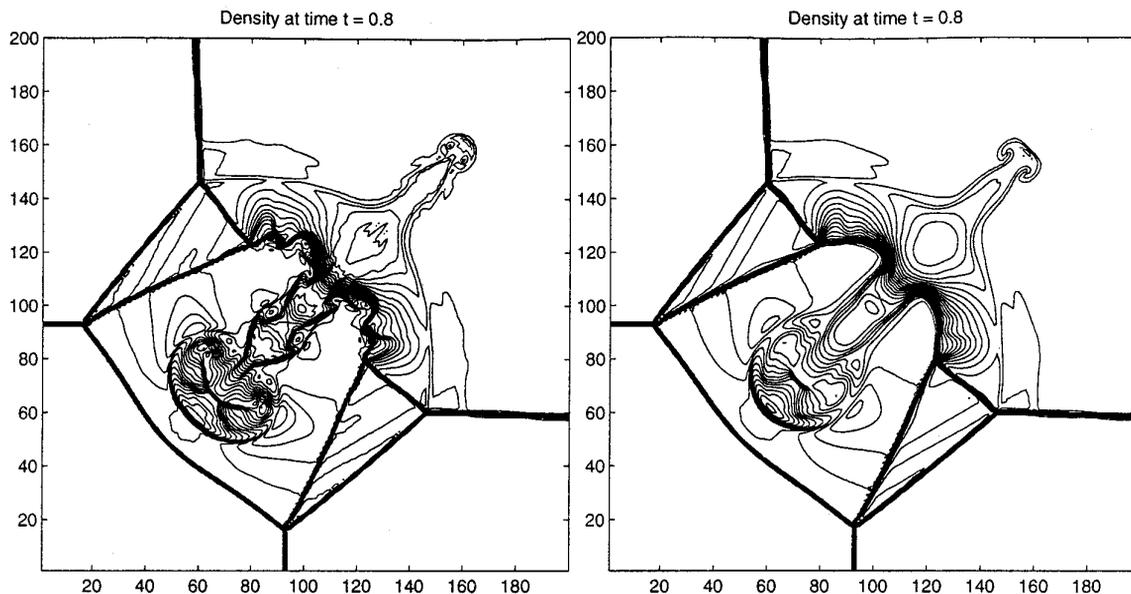


FIG. 6. Density contours for a two-dimensional Riemann problem on a 200×200 grid. Left: superbee limiter. Right: monotized centered limiter.

the linear system $q_t + Aq_x + Bq_y = 0$, consider data of the form

$$q_{IJ} = \exp(i(\xi x_I + \eta y_J)),$$

where $i = \sqrt{-1}$ in this section. A linear scheme then gives

$$\bar{q}_{IJ} = T(\xi, \eta)q_{IJ},$$

where the ‘‘amplification matrix’’ $T(\xi, \eta)$ depends on the wave numbers ξ and η (and also, of course, on the particular A and B , as well as the mesh ratios $\Delta t/\Delta x$ and $\Delta t/\Delta y$). The method is stable in the 2-norm on a particular grid if $\rho(T(\xi, \eta)) \leq 1$ for all ξ and η , where ρ is the spectral radius. (see, e.g., [56].)

In general we can write

$$\begin{aligned} T(\xi, \eta) = & I - \frac{\Delta t}{\Delta x} (e^{i\xi\Delta x} - 1) T_A(\xi, \eta) \\ & - \frac{\Delta t}{\Delta y} (e^{i\eta\Delta y} - 1) T_B(\xi, \eta), \end{aligned}$$

where the T_A and T_B terms arise from considering the fluxes F and G , respectively.

The form of T_A and T_B is given below for various forms of the algorithm. The superscripts (m_1, m_2) on the matrices indicate exactly which method is being studied, as follows:

$$\begin{aligned} m_1 = & \begin{cases} 1 & \text{if only first-order fluctuation terms are used} \\ 2 & \text{if second-order corrections are also used} \end{cases} \\ m_2 = & \begin{cases} 0 & \text{if no transverse wave propagation is used} \\ 1 & \text{if transverse fluctuations as in Section 3.3 are used} \\ 2 & \text{if second-order corrections are also} \\ & \text{propagated transversely, as in Section 3.3.} \end{cases} \end{aligned}$$

For upwind differencing with no transverse propagation,

$$\begin{aligned} T_A^{1,0} &= e^{-i\xi\Delta x} A^+ + A^- \\ T_B^{1,0} &= e^{-i\eta\Delta y} B^+ + B^-. \end{aligned}$$

For the other algorithms we will only display T_A to save space. The matrix T_B can be obtained by replacing $A, \xi, \Delta x$ by $B, \eta, \Delta y$, respectively, and vice versa.

When transverse propagation of the first-order waves is included, we find that

$$\begin{aligned} T_A^{1,1} = & T_A^{1,0} - \frac{1}{2} \frac{\Delta t}{\Delta y} (A^+ B^+ (1 - e^{-i\eta\Delta y}) \\ & + A^+ B^- (e^{i\eta\Delta y} - 1)) e^{-i\xi\Delta x} \\ & - \frac{1}{2} \frac{\Delta t}{\Delta y} (A^- B^+ (1 - e^{-i\eta\Delta y}) \\ & + A^- B^- (e^{i\eta\Delta y} - 1)). \end{aligned}$$

When the second-order corrections (39) are also included (but not propagated transversely), we obtain

$$T_A^{2,1} = T_A^{1,1} + \frac{1}{2} (A^+ - A^-) \left(I - \frac{\Delta t}{\Delta x} (A^+ - A^-) \right) (1 - e^{-i\xi\Delta x}).$$

Finally, if we also propagate the second-order corrections transversely,

$$\begin{aligned} T_A^{2,2} = & T_A^{2,1} - \frac{1}{2} \frac{\Delta t}{\Delta y} (B^+ - B^-) \left(I - \frac{\Delta t}{\Delta y} (B^+ - B^-) \right) \\ & \times (1 - e^{-i\eta\Delta y}) (e^{i\eta\Delta y} - 1) (A^- + A^+ e^{-i\xi\Delta x}). \end{aligned}$$

We can test the stability of the methods on any given system of linear equations by numerically computing $\max_{\xi, \eta} \rho(T(\xi, \eta))$ over a discrete set of points ξ, η in $[0, 2\pi] \times [0, 2\pi]$, where $\rho(T)$ is the spectral radius of the matrix T . By doing this for different values of the mesh ratio $\Delta t/\Delta x = \Delta t/\Delta y$ and observing at what point this value exceeds 1, it is possible to estimate the stability limit. Note that the maximum value of the spectral radius will never be less than 1 since $T(0, 0) = I$.

This has been implemented in a `matlab` script that is available in `CLAWPACK`. As a typical example we consider the two-dimensional acoustics equations (40) with wave speed $c = 1$, so that

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

The stability limit for this case gives the stability limit on $c\Delta t/\Delta x$ for general wave speed c , since c simply enters as a scalar factor.

Table I shows the results for each method, and also for the standard Lax–Wendroff method for comparison. For Lax–Wendroff,

$$\begin{aligned} T_A^{LW} = & (A^+ e^{-i\xi\Delta x} + A^-) - \frac{1}{8} \frac{\Delta t}{\Delta x} AB (e^{i\eta\Delta y} - e^{-i\eta\Delta y}) (1 + e^{-i\xi\Delta x}) \\ & + \frac{1}{2} (A^+ - A^-) \left(I - \frac{\Delta t}{\Delta x} (A^+ - A^-) \right) (1 - e^{-i\xi\Delta x}). \end{aligned}$$

The Lax–Wendroff method is stable only for $c\Delta t/\Delta x < 0.7$ approximately. The pure upwind method denoted (1,0) is stable only for $c\Delta t/\Delta x \leq \frac{1}{2}$. Both of these results are seen in the table. By contrast, all of the wave propagation methods which include the correct transverse propagation of the first-order waves, methods (1, 1), (2, 1), and (2, 2) are seen to be stable for $c\Delta t/\Delta x \leq 1$, the best that can be expected. This is also confirmed by computations with `CLAWPACK`.

TABLE I

Amplification Factors for the Various Methods Applied to Two-Dimensional Linear Acoustics

$\frac{c\Delta t}{\Delta x}$	$\max_{\xi,\eta} \rho(T(\xi, \eta))$ for				
	$T^{1,0}$	$T^{1,1}$	$T^{2,1}$	$T^{2,2}$	T^{LW}
0.10	1.00	1.00	1.00	1.00	1.00
0.20	1.00	1.00	1.00	1.00	1.00
0.30	1.00	1.00	1.00	1.00	1.00
0.40	1.00	1.00	1.00	1.00	1.00
0.50	1.00	1.00	1.00	1.00	1.00
0.51	1.03	1.00	1.00	1.00	1.00
0.60	1.40	1.00	1.00	1.00	1.00
0.70	1.80	1.00	1.00	1.00	1.01
0.80	2.20	1.00	1.00	1.00	1.53
0.90	2.60	1.00	1.00	1.00	2.21
1.00	3.00	1.00	1.00	1.00	2.97
1.01	3.04	1.04	1.04	1.08	3.04

3.8. *Nonconservative Systems*

Nonconservative equations are handled in two dimensions in exactly the same way as in one space dimension, with the addition of a transverse splitting. The “flux differences” $\mathcal{A}^-\Delta q_{ij}$ and $\mathcal{A}^+\Delta q_{ij}$ are defined by the Riemann solution normal to each interface, and these are then split in the transverse direction using the eigenstructure in that direction. This is best illustrated with an example.

EXAMPLE 3.8.1. As an example we consider two-dimensional acoustics in a heterogeneous material. The equations are

$$q_t + A(x, y)q_x + B(x, y)q_y = 0,$$

where

$$q = \begin{bmatrix} p \\ u \\ v \end{bmatrix}, \quad A = \begin{bmatrix} 0 & K(x, y) & 0 \\ 1/\rho(x, y) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & K(x, y) \\ 0 & 0 & 0 \\ 1/\rho(x, y) & 0 & 0 \end{bmatrix}.$$

The solution to the Riemann problem normal to each cell interface is done exactly as in the one-dimensional solution of Section 2.6. Let ρ_{ij} be the density and sound speed in the (i, j) cell, where $c = \sqrt{K/\rho}$. Then the Riemann problem

at the interface between $(i - 1, j)$ and (i, j) , for example, gives

$$\mathcal{W}^1 = \alpha^1 \begin{bmatrix} -c_{i-1,j} \\ 1/\rho_{i-1,j} \\ 0 \end{bmatrix}, \quad \mathcal{W}^2 = \alpha^2 \begin{bmatrix} c_{i,j} \\ 1/\rho_{ij} \\ 0 \end{bmatrix},$$

where

$$\alpha^1 = (-\Delta q^1/\rho_{ij} + c_{ij}\Delta q^2)/(c_{i-1,j}/\rho_{ij} + c_{ij}/\rho_{i-1,j})$$

$$\alpha^2 = (\Delta q^1/\rho_{i-1,j} + c_{i-1,j}\Delta q^2)/(c_{i-1,j}/\rho_{ij} + c_{ij}/\rho_{i-1,j}).$$

Again the fluctuations $\mathcal{A}^-\Delta q$ and $\mathcal{A}^+\Delta q$ are given by the product of the waves and wave speeds,

$$\mathcal{A}^-\Delta q = \lambda_i^1 \mathcal{W}^1, \quad \mathcal{A}^+\Delta q = \lambda_i^2 \mathcal{W}^2,$$

where $\lambda_i^1 = -c_{i-1,j}$ and $\lambda_i^2 = c_{ij}$.

Transverse propagation. The rightgoing fluctuation $\mathcal{A}^+\Delta q$ is split into up-going and down-going transverse fluctuations $\mathcal{B}^+\mathcal{A}^+\Delta q_{ij}$ and $\mathcal{B}^-\mathcal{A}^+\Delta q_{ij}$ that modify the fluxes $G_{i,j+1}$ and G_{ij} above and below the cell (i, j) . We can think of first splitting this fluctuation into waves

$$\beta^1 \begin{bmatrix} -c_{ij} \\ 0 \\ 1/\rho_{ij} \end{bmatrix}, \quad \beta^2 \begin{bmatrix} c_{ij} \\ 0 \\ 1/\rho_{ij} \end{bmatrix},$$

moving vertically with speeds $-c_{ij}$ and $+c_{ij}$, respectively. But now these waves must be split at the interface into transmitted and reflected pieces. Only the transmitted parts are used in defining the transverse fluctuations which modify the fluxes above or below this cell. The downward transmitted wave should be of the form

$$\tilde{\beta}^1 \begin{bmatrix} -c_{i,j-1} \\ 0 \\ 1/\rho_{i,j-1} \end{bmatrix} \quad \text{with speed } -c_{i,j-1}$$

and we set

$$\mathcal{B}^+\mathcal{A}^+\Delta q_{ij} = -c_{i,j-1}\tilde{\beta}^1 \begin{bmatrix} -c_{i,j-1} \\ 0 \\ 1/\rho_{i,j-1} \end{bmatrix}.$$

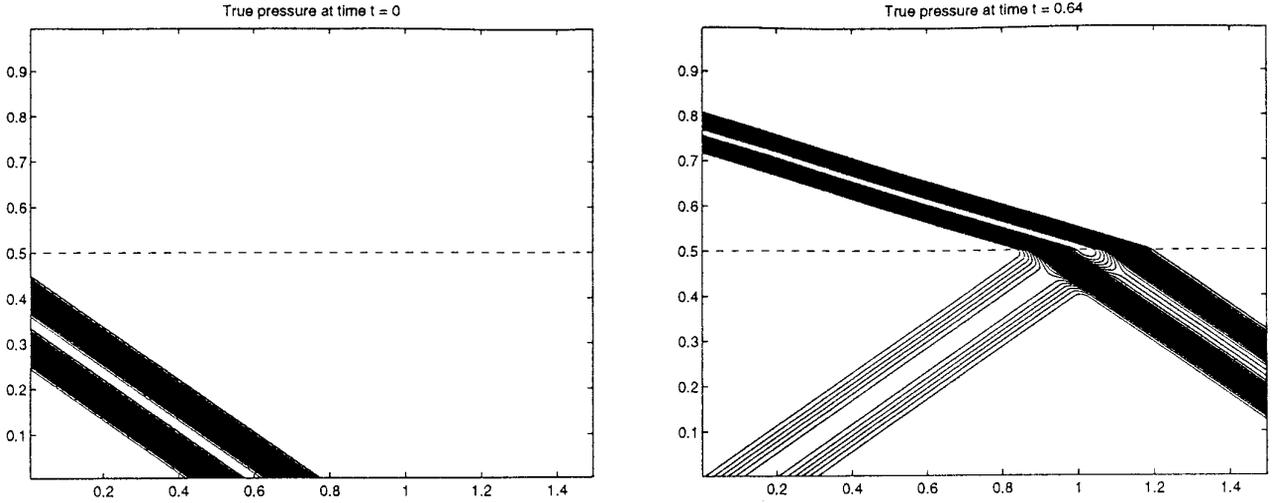


FIG. 7. Plane wave hitting an interface in sound speed with the acoustics equations. The initial data and true solution at time $t = 0.64$.

The upward transmitted wave is of the form

$$\tilde{\beta}^2 \begin{bmatrix} -c_{i,j+1} \\ 0 \\ 1/\rho_{i,j+1} \end{bmatrix} \text{ with speed } +c_{i,j+1}$$

and we set

$$\mathcal{B}^- \mathcal{A}^+ \Delta q_{ij} = +c_{i,j+1} \tilde{\beta}^2 \begin{bmatrix} -c_{i,j+1} \\ 0 \\ 1/\rho_{i,j+1} \end{bmatrix}.$$

Solving first for β^1 , β^2 and then for $\tilde{\beta}^1$, $\tilde{\beta}^2$, we find that

$$\tilde{\beta}^1 = (-(\mathcal{A}^+ \Delta q)^1 / \rho_{ij} + (\mathcal{A}^+ \Delta q)^3 c_{ij}) / (c_{i,j-1} / \rho_{ij} + c_{ij} / \rho_{i,j-1}) \quad (50)$$

$$\tilde{\beta}^2 = (-(\mathcal{A}^+ \Delta q)^1 / \rho_{ij} + (\mathcal{A}^+ \Delta q)^3 c_{ij}) / (c_{ij} / \rho_{i,j+1} + c_{i,j+1} / \rho_{ij}), \quad (51)$$

where $(\mathcal{A}^+ \Delta q)^1$ and $(\mathcal{A}^+ \Delta q)^3$ are the first and third components of the vector $\mathcal{A}^+ \Delta q$, respectively. The up-going and down-going transverse fluctuations $\mathcal{B}^+ \mathcal{A}^+ \Delta q_{ij}$ and $\mathcal{B}^- \mathcal{A}^+ \Delta q_{ij}$ are given by the product of the respective wave speed and wave, as indicated above.

EXAMPLE 3.8.2. This example shows a plane wave striking an interface in density. We first consider the case where the interface is aligned with the grid, with

$$\rho(x, y) = \begin{cases} \rho_1 = 1 & \text{if } y < y_0 \\ \rho_2 = 4 & \text{if } y > y_0 \end{cases}$$

and $\beta \equiv 1$. As initial data we take a plane wave with a single hump propagating in some direction at an angle to the grid and interface. We can easily compute the reflected and transmitted waves (see the CLAWPACK code and documentation for details).

Figure 7 shows the initial data and true solution at time $t = 0.64$ for the case studied. Figure 8 shows computed results at this same time on two different grids. Along with the contour plot, two cross sections of the solution along $x = 0.6$ and $x = 1.0$ are shown. We observe good accuracy of both the transmitted and reflected waves.

In these computations the superbee limiter was used. Without a limiter, dispersive wave behavior would be seen with the second-order method, leading to spurious oscillations and phase errors. With the limiter there are no spurious oscillations and the limiters are apparently effective even as the wave interacts with the interface.

Boundary conditions for this two-dimensional computation are set in a subroutine that extends the solution to two rows of ghost cells along each side of the physical domain. For this computation the exact solution is specified in all ghost cells at each step of the computation. This minimizes the effects of numerical boundary conditions on the computed results. This appears to be an over-specification of data since there is only one incoming characteristic at each boundary. However, with a wave propagation approach based on Riemann solvers it is possible to over-specify in this manner without exciting instabilities since solving the Riemann problem properly selects the incoming characteristic automatically.

EXAMPLE 3.8.3. Next we consider the same example but with the interface (and plane waves) rotated at an angle to the grid, so that the interface cuts through grid cells. In the previous example we had $\rho_{ij} = \rho_1$ or ρ_2 in

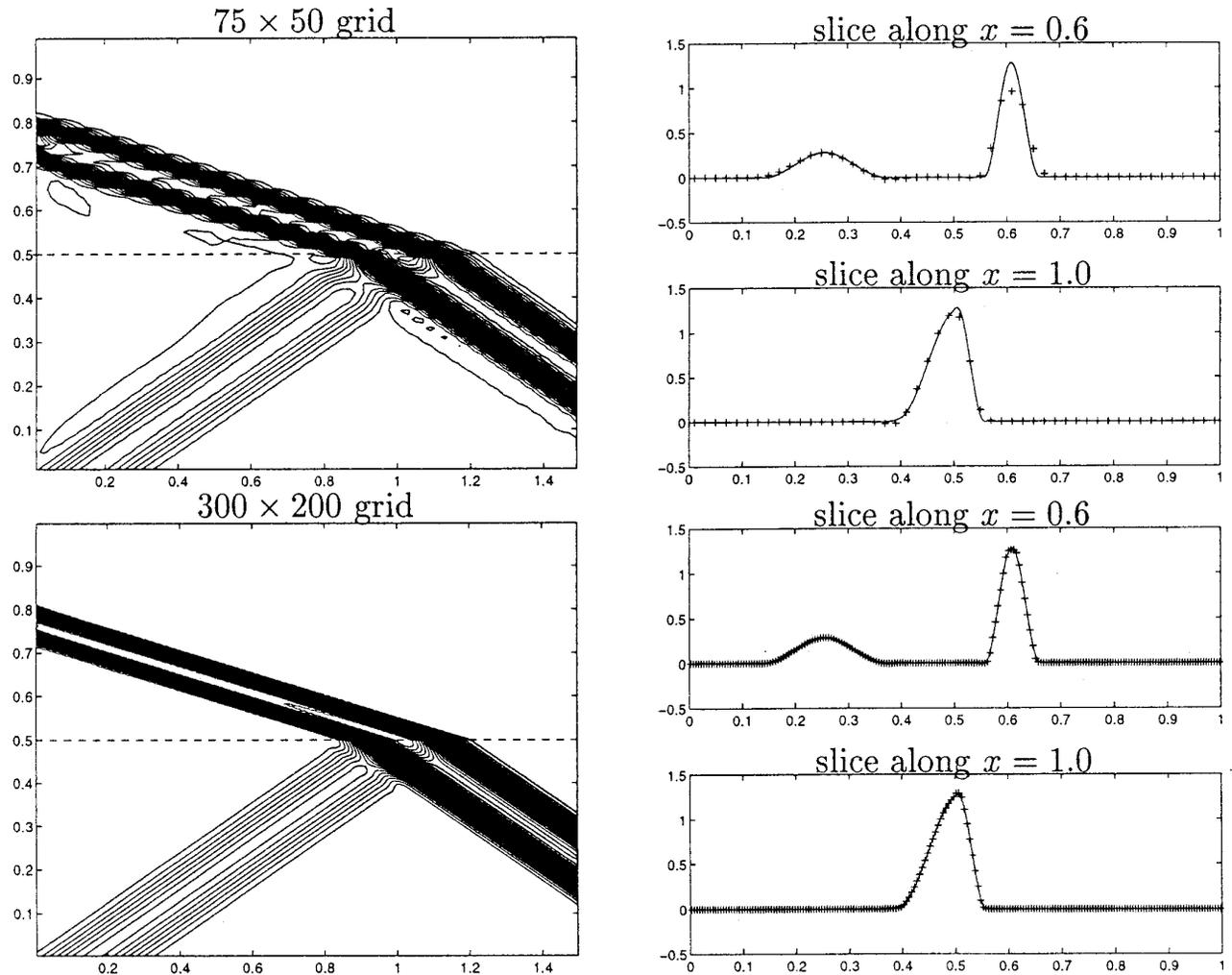


FIG. 8. Plane wave hitting an interface that is aligned with the grid, as computed by CLAWPACK on a 75×50 grid (top figures) and 300×200 grid (bottom figures).

every cell, with a sharp interface aligned with the grid. Now the grid cells that are cut by the grid are assigned a value ρ_{ij} obtained by averaging $\rho(x, y)$ over the grid cell, yielding a convex combination of ρ_1 and ρ_2 . With many algorithms this type of averaging would lead to considerable loss of resolution and the generation of numerical noise as the wave interacts with the smeared interface.

The nonconservative wave propagation methods developed here seem to be very robust. Figure 9 shows results at the same time as before. The solution is nearly as good as when the interface was aligned with the grid.

Analysis of the error on slices away from the boundaries shows that in both cases the pressure is only first-order accurate, when measured in the 1-norm or max-norm. This is not surprising since the solution is not smooth as it passes through the interface. Moreover, the velocities, which are discontinuous across the interface, are smeared in much

the same way as discontinuities are smeared in a shock computation.

Better accuracy at the interface can be achieved by combining the CLAWPACK algorithms with an *immersed interface method*, similar to methods developed in [36, 37, 39], which can give second-order accurate results on a uniform grid even when there are discontinuities that are not aligned with the grid. This was studied in [38] for the acoustics equations and has also been extended to elasticity [63]. A discussion of this technique is beyond the scope of the present paper.

3.9. Capacity-Form Differencing

In two space dimensions, capacity-form differencing takes essentially the same form as in one dimension, with the obvious extension of (33) to a form analogous to (28).

EXAMPLE 3.9.1. As an example we consider advection of a tracer in a density-stratified flow. The density $\rho(x, y)$ is taken to be constant in time, and in this test problem varies only with y , e.g.,

$$\rho(x, y) = \rho(y) = e^{-\gamma y} \quad \text{for some } \gamma, \quad (52)$$

as one would expect in the atmosphere. Now $q(x, y, t)$ represents the mass fraction of some tracer, and so ρq is the mass per unit volume of the tracer, which is the conserved quantity. The conservation law is

$$\rho q_t + (\rho u q)_x + (\rho v q)_y = 0. \quad (53)$$

We take $u(x, y), v(x, y)$ to be a fixed velocity field, chosen so that

$$(\rho u)_x + (\rho v)_y = 0. \quad (54)$$

This is a consequence of conservation of mass and the fact that $\rho_t \equiv 0$. Then Eq. (53) can be rewritten as

$$\rho q_t + \rho u q_x + \rho v q_y = 0. \quad (55)$$

We could eliminate ρ from this equation, but solving the resulting advection equation $q_t + u q_x + v q_y = 0$ would not guarantee conservation of ρq . Instead we solve in the form (55) using capacity-form differencing with $\kappa(x, y) = \rho(x, y)$.

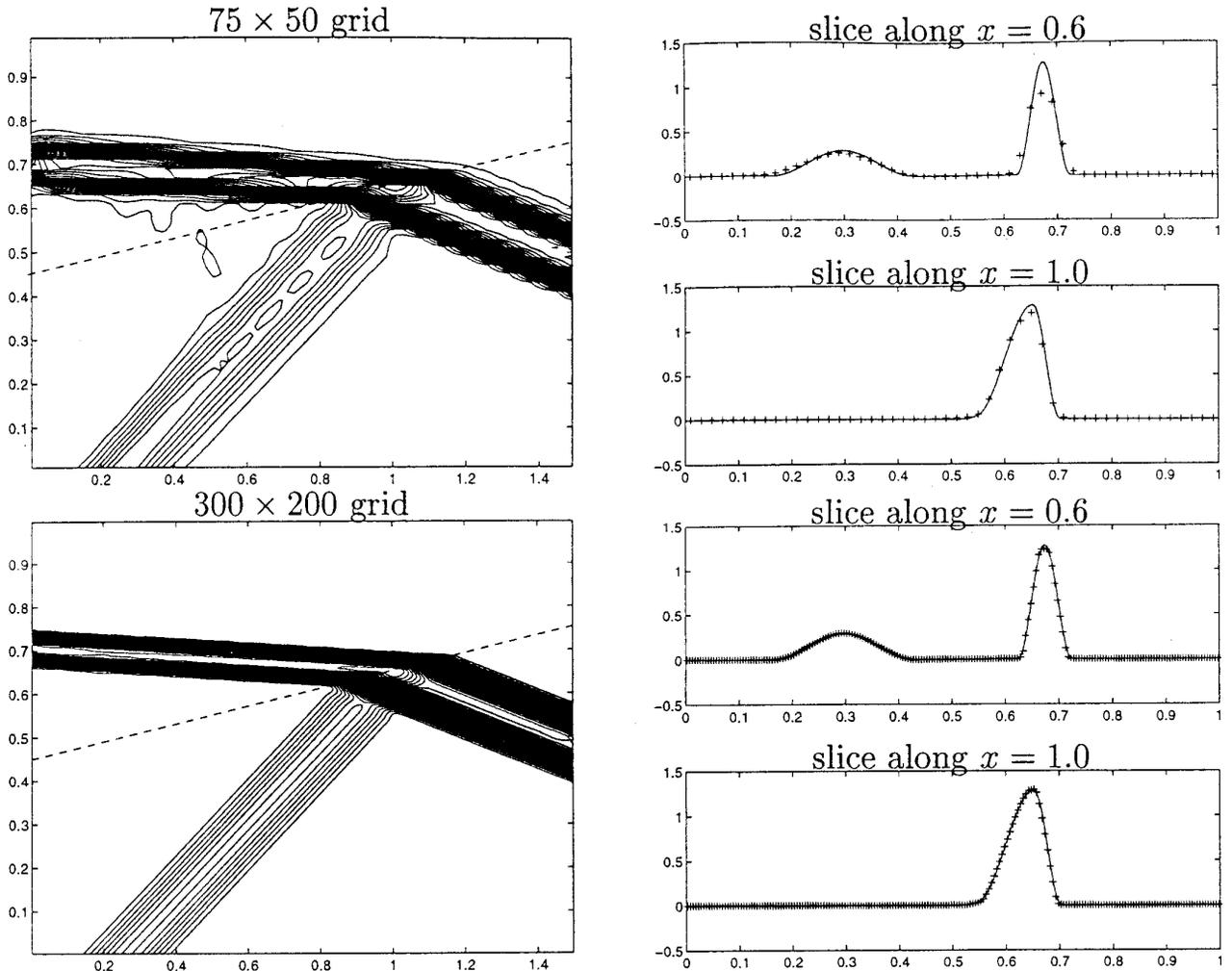


FIG. 9. Plane wave hitting an interface that is not aligned with the grid, as computed by CLAWPACK on a 75×50 grid (top figures) and 300×200 grid (bottom figures).

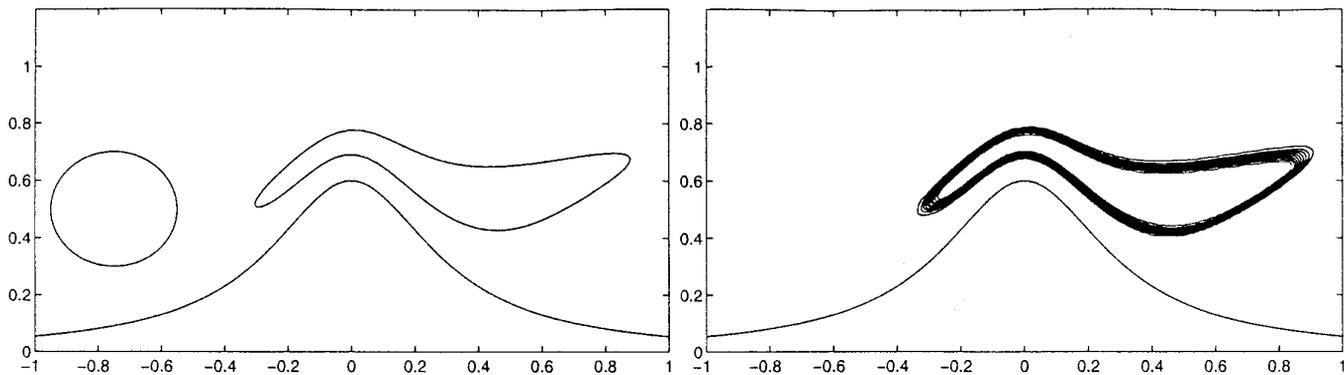


FIG. 10. Density-stratified flow over a hump. Left: initial data is 1 inside the circular region and 0 elsewhere. The region where the solution is 1 at time $t = 0.18$ is also shown. Right: Computed results on a 200×100 Cartesian grid.

As a specific test, consider flow over a hump with the bottom topography given by

$$B(x) = \frac{\alpha}{1 + \beta x^2} \quad (56)$$

in the domain $-1 \leq x \leq 1$, $B(x) \leq y \leq 1$ (with $\alpha < 1$). The velocity field is chosen by using the “stream function”

$$\psi(x, y) = \frac{y - B(x)}{1 - B(x)}$$

to define

$$\begin{aligned} \rho u &= \psi_y = \frac{1}{1 - B(x)}, \\ \rho v &= -\psi_x = \frac{B'(x)(1 - y)}{(1 - B(x))^2}, \end{aligned}$$

so that condition (54) is satisfied. Dividing by ρ gives the velocity field. Note that (u, v) is not divergence free and ψ is not a stream function for this velocity, although it is true that contours of constant ψ give streamlines of the flow. Note also that with the density profile (52), the velocities increase exponentially with y . This may not be reasonable physically, but it does give a challenging test problem that is easy to set up.

In the test below we use $\alpha = 0.6$, $\beta = 10$, and $\gamma = 2.5$. Figure 10 shows the initial data and exact solution at time $t = 0.18$ for data consisting of a circular blob of tracer:

$$q(x, y, 0) = \begin{cases} 1 & \text{if } (x + 0.75)^2 + (y - 0.5)^2 < (0.2)^2 \\ 0 & \text{otherwise.} \end{cases}$$

Note that the area of the blob increases as it rises and decreases again as it falls. In this problem the area is not conserved since it is the integral of ρq that is conserved, not the integral of q . The value of q remains 1 inside the blob since q remains constant along particle paths for the color equation. The severe stretching of the blob is due to the fact that the velocity increases exponentially with y , so that the top of the blob moves more quickly than the bottom.

Figure 10 also shows computed results on a 200×100 grid using the superbee limiter. Using capacity-form differencing the sum of $\rho_{ij} q_{ij}$ over all grid points is exactly conserved (to machine precision) until the time at which tracer begins to pass through the right boundary.

3.10. Curvilinear Grids and Coordinate Mappings

In principle the capacity-form differencing introduced in Section 3.9 can be used to solve any conservation law on a curvilinear grid that can be smoothly mapped to a uniform rectangular grid. This will be demonstrated here for advection and applied to the example of stratified flow from the previous section.

Consider the advection equation

$$q_t + (u(x, y, t)q)_x + (v(x, y, t)q)_y = 0 \quad (57)$$

in an irregular region of the x - y plane that can be mapped smoothly to a rectangle. Then equation (57) can be transformed to an advection equation on the rectangle and solved on a uniform Cartesian grid. The computational points will be denoted by (ξ_i, η_j) with $\xi_i = i\Delta\xi$, $\eta_j = j\Delta\eta$. A grid mapping defines the relation between the point (ξ_i, η_j) and the corresponding physical point (x_{ij}, y_{ij}) . We assume that this mapping is defined by a differentiable function

$$x_{ij} = X(\xi_i, \eta_j), \quad y_{ij} = Y(\xi_i, \eta_j).$$

Then the advection equation (57) can be transformed to

$$J(\xi, \eta)q_t + (\hat{u}(\xi, \eta, t)q)_\xi + (\hat{v}(\xi, \eta, t)q)_\eta = 0, \quad (58)$$

where

$$\begin{aligned} \hat{u}(\xi, \eta, t) &= Y_\eta(\xi, \eta)u(X(\xi, \eta), Y(\xi, \eta)) \\ &\quad - X_\eta(\xi, \eta)v(X(\xi, \eta), Y(\xi, \eta)) \\ \hat{v}(\xi, \eta, t) &= -Y_\xi(\xi, \eta)u(X(\xi, \eta), Y(\xi, \eta)) \\ &\quad + X_\xi(\xi, \eta)v(X(\xi, \eta), Y(\xi, \eta)) \\ J(\xi, \eta) &= X_\xi(\xi, \eta)Y_\eta(\xi, \eta) - X_\eta(\xi, \eta)Y_\xi(\xi, \eta). \end{aligned} \quad (59)$$

Here J is the *Jacobian* of the grid transformation.

If the flow is divergence free, $u_x + v_y = 0$, then it is easy to verify that $\hat{u}_\xi + \hat{v}_\eta = 0$ as well. Then (58) can also be written in the advective form

$$Jq_t + \hat{u}q_\xi + \hat{v}q_\eta = 0. \quad (60)$$

EXAMPLE 3.10.1. The stratified-flow advection problem of Example 3.9.1 can be solved on a curvilinear grid that conforms to the bottom topography. One choice might be

Grid 1: $X(\xi, \eta) = \xi, Y(\xi, \eta) = B(\xi) + \eta,$

as shown in Fig. 11. We now solve the advection equation

$$\begin{aligned} \rho(Y(\xi, \eta))J(\xi, \eta)q_t + \rho(Y(\xi, \eta))\hat{u}(\xi, \eta)q_\xi \\ + \rho(Y(\xi, \eta))\hat{v}(\xi, \eta)q_\eta = 0, \end{aligned} \quad (61)$$

using capacity-form differencing with

$$\kappa(\xi, \eta) = \rho(Y(\xi, \eta))J(\xi, \eta).$$

Computational results are shown in Fig. 11, again on a 200×100 grid as in Example 3.9.1. The results are quite similar and in the computation $\sum \rho_{ij}J_{ij}q_{ij}$ is exactly conserved (to machine precision). This is the proper quantity to conserve since it approximates the integral of ρq .

Even better results can be obtained by choosing the grid so that grid lines are streamlines of the flow. For this test problem with constant velocities this is easy to accomplish:

Grid 2: $X(\xi, \eta) = \xi, Y(\xi, \eta) = B(\xi) + \eta(1 - B(\xi)).$

Figure 11 also shows the grid and computed results, which are considerably sharper since the velocity is now zero in the η -direction, minimizing numerical smearing.

In all of these tests (the Cartesian grid, Grid 1, and Grid 2) the superbee limiter was used and mild overshoots

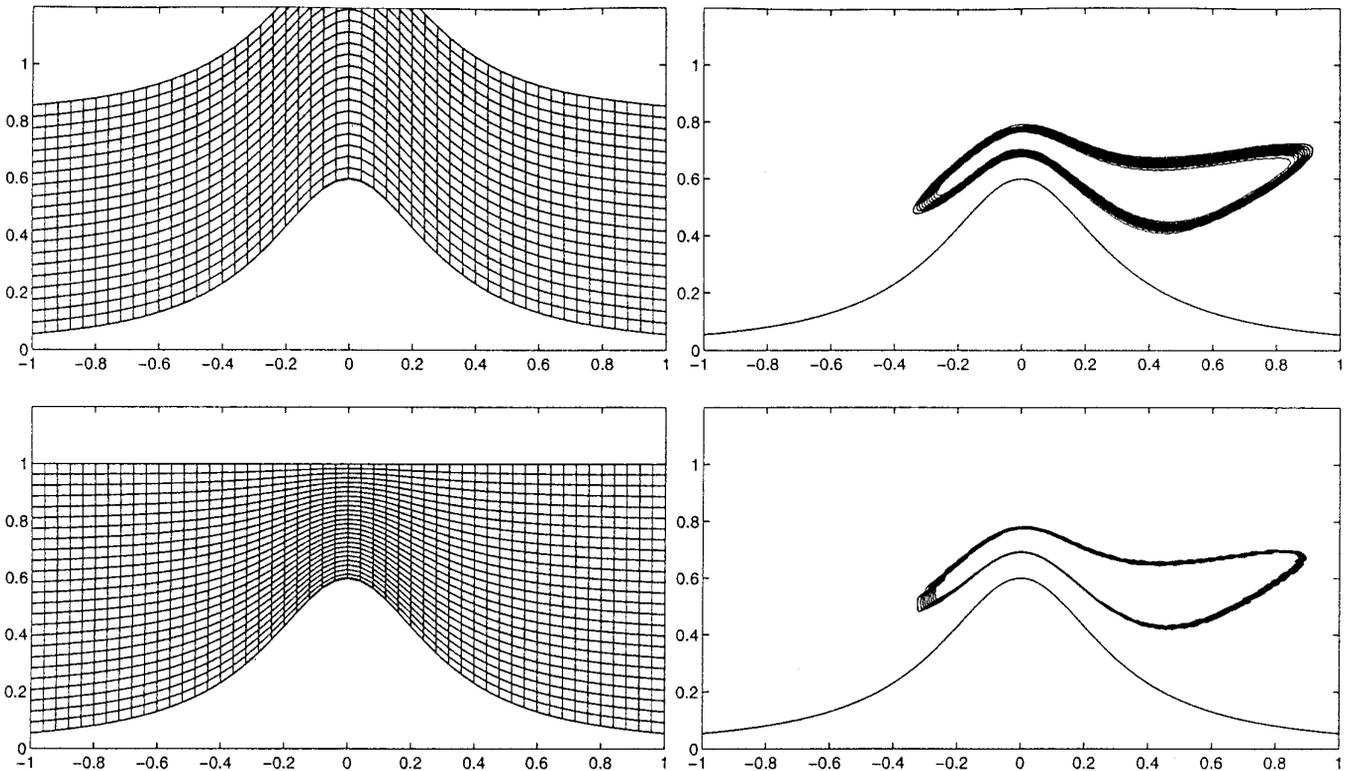


FIG. 11. Density-stratified flow over a hump. Computational results on two different 200×100 curvilinear grids. The grids are shown for a coarser grid size, 50×25 .

TABLE II

Errors in Stratified Flow over a Hump with Smooth Initial Data on Three Different Grids

ξ	1-norm errors			Max-norm errors		
	Cartesian	Grid 1	Grid 2	Cartesian	Grid 1	Grid 2
0.04	5.06D-03	6.9D-03	5.71D-03	1.20D-01	1.34D-01	7.55D-02
0.02	1.28D-03	1.9D-03	1.46D-03	3.04D-02	4.55D-02	1.98D-02
0.01	4.62D-04	6.04D-04	3.89D-04	9.32D-03	1.33D-02	4.90D-03
Order	1.47	1.66	1.91	1.71	1.77	2.01

and/or undershoots were observed at the level of about 2%. The contour levels plotted in each case are at $q = 0.05, 0.10, 0.15, \dots, 0.95$. Use of the minmod limiter gives strictly monotonic results but slightly more smearing.

EXAMPLE 3.10.2. We now check the order of accuracy for the stratified flow problem on each of the grids used above, as well as the Cartesian grid of Example 3.9.1. Smooth initial data is now used, given by

$$q(x, y, 0) = \exp(-50((x + 0.5)^2 + (y - 0.5)^2))$$

and flow is over a hump with $\alpha = 0.4$ and $\beta = 10$ in (56) and the equation was solved up to time $\bar{t} = 0.1$. The exact solution can be computed by integrating backwards along streamlines using an ODE solver. To compute $q(\bar{x}, \bar{y}, \bar{t})$ at any arbitrary point we solve

$$X'(t) = -u(X(t), Y(t)), \quad X(0) = \bar{x},$$

$$Y'(t) = -v(X(t), Y(t)), \quad X(0) = \bar{y},$$

up to time $t = \bar{t}$ and then

$$q(\bar{x}, \bar{y}, \bar{t}) = q(X(\bar{t}), Y(\bar{t}), 0).$$

This has been done using the code LSODE from `netlib`, with a tolerance of 10^{-6} .

Table II shows the 1-norm of the error in the computed solution as a function of $\Delta\xi = \Delta\eta$ as each grid is refined. Good accuracy is observed on each grid. Note that the curvilinear grids used here are not orthogonal.

4. THE CLAWPACK SOFTWARE

All of the numerical results presented in this paper were computed with the CLAWPACK software, and the driver programs are available within CLAWPACK.

Details of the implementation and the use of CLAWPACK are given in [29] and within the software, and they will not be discussed extensively here. However, it may be useful

to briefly outline what the user must provide and how it relates to the algorithms as presented here.

In addition to a driver program, which sets the initial conditions and calls CLAWPACK, the user must provide sub-routines to solve the Riemann problems. In one space dimension a single routine `rp1` is needed, which takes data q_l and q_r and returns the fluctuations $\mathcal{A}^- \Delta q$ and $\mathcal{A}^+ \Delta q$ and, also, the waves \mathcal{W}^p and speeds λ^p .

In two dimensions the user must provide two Riemann solvers. The first, `rp2`, solves Riemann problems normal to each cell interface and has the same outputs as the one-dimensional routine `rp1`. This same routine is called in both the x - and y -directions and a parameter `ixy` indicates which is the normal direction. Often the formulas for solving the Riemann problem are nearly identical in the two directions, as in the acoustics or Euler equations, once `ixy` has been used to determine which velocity components is normal to the interface.

The second Riemann solver `rpt2` takes a fluctuation $\mathcal{A}^* \Delta q$ and returns the transverse fluctuation splitting $\mathcal{B}^- \mathcal{A}^* \Delta q_{ij}$ and $\mathcal{B}^+ \mathcal{A}^* \Delta q_{ij}$. This routine is called twice at each interface once applied to $\mathcal{A}^- \Delta q$ and once to $\mathcal{A}^+ \Delta q$. This routine typically involves similar formulas as in `rp2`, but it may be much simpler. For example, in the Euler equations the Roe averages are computed in `rp2` and then passed in a common block to `rpt2`. Again the same routine `rpt2` is also used in the y -direction, in which case the input would be a fluctuation $\mathcal{B}^* \Delta q$ resulting from calling `rp2` in this direction and the outputs of `rpt2` would be interpreted as $\mathcal{A}^- \mathcal{B}^* \Delta q_{ij}$ and $\mathcal{A}^+ \mathcal{B}^* \Delta q_{ij}$.

The user must also provide a routine that specifies the boundary conditions. This routine extends the data at the start of each time step from the computational domain to a border of two ghost cells along each side.

A capacity function $\kappa(x, y)$ can be specified as an array of values on the grid. In addition, a source term can be included, so that the equations being solved have the form

$$\kappa(x, y)q_t + A(q, x, y, t)q_x + B(q, x, y, t)q_y = \psi(q, \kappa, x, y, t).$$

The source term is handled by a fractional step (splitting) method. In each time step the homogeneous hyperbolic system is first advanced over time Δt , and then the source terms are advanced in each grid cell over the same time increment. An option allows the Strang splitting [55] to be used, in which the source terms are split into two half steps, one before and one after the hyperbolic step. Formally this gives second-order accuracy, but in practice an effect similar to what was analyzed in Section 2.4 for nonautonomous systems is seen; the “first-order” splitting gives essentially as good results in the high-resolution context and requires less work per time step. In principle one could combine the second half step on the source terms from one time step with the first half step on the next time step,

which shows that the Strang splitting and “first-order” splitting differ only in how the first and last time steps are handled (which is why no loss in resolution is observed). This would be more difficult to implement, however, since in `CLAWPACK` variable time steps can be automatically chosen based on a desired Courant number, which will change from step to step.

A variety of Riemann solvers and boundary conditions routines are provided in `CLAWPACK`, not only for the examples presented here but also for other problems, including Burgers’ equation, the isothermal equations, and shallow water equations.

5. EFFICIENCY AND COMPARISON WITH OTHER APPROACHES

The solution of Riemann problems is typically a very expensive part of the procedure. These Riemann-based high-resolution methods are intended primarily for a certain class of problems where discontinuities in the problem or its solution (or at least steep gradients) lead to difficulties with more standard finite-difference methods. For problems with smooth solutions it may be possible to obtain better accuracy much more efficiently by using high-order methods that do not require Riemann solutions.

The wave-propagation algorithms in two dimensions require solving transverse Riemann problems as well as Riemann problems in the normal direction. This Riemann solver may be much simpler than the normal solver, as indicated in Section 3.2 for the Euler equations, but still this requires additional work. For many problems a viable alternative is dimensional splitting, in which the one-dimensional algorithm is applied in alternating sweeps in the x - and y -directions. This is included as an option in `CLAWPACK`, and for many problems this is nearly as effective as the full multidimensional algorithm at reduced cost. For other problems it appears that the multidimensional algorithms are better. In some applications it may also be inconvenient to use a splitting technique. For example, in an incompressible velocity field where x - and y -effects must properly cancel for conservation the use of the multidimensional method allows the advective form of the equations to be used to good advantage (as discussed in [35]). Boundary conditions may also be more difficult to impose with splitting methods. Unsplit methods are also most convenient in conjunction with adaptive mesh refinement or Cartesian grid treatments of irregular boundaries (e.g., [6, 7, 11]).

The methods proposed here are related to other multidimensional methods proposed in the literature, some of which are mentioned in the introduction. In particular, it is interesting to compare this method with the approach of Collella [13] and the three-dimensional generalization of Saltzman [47], which is similar in that one-dimensional Riemann problems are solved at grid interfaces, transverse

derivatives are approximated, and limiter functions are used to achieve high resolution. That approach is fundamentally different from what is used here in that it can be summarized as “first interpolate to the interface and then compute the flux by solving the Riemann problem.” The wave-propagation approach is the other way around: “first solve the Riemann problem and then distribute the information.”

Consider the interface between cells $(i - 1, j)$ and (i, j) . In the “interpolate first” approach, two values q^- and q^+ to $q(x_i, y_j + \Delta y/2, t_n + \Delta t/2)$ are first obtained as approximations to the value at the interface a half time step forward in time. A Riemann problem is then solved based on these states to obtain the numerical flux. If $q^- = q_{i-1,j}$ and $q^+ = q_{ij}$ then this would be just Godunov’s method. The high-resolution method is obtained by defining q^- , say, as an approximation to

$$q_{i-1,j} + \frac{\Delta x}{2} q_x + \frac{\Delta t}{2} q_t,$$

where q_x is then approximated in cell $(i - 1, j)$ by defining a slope in the x -direction based on the data in this cell and neighboring cells, using some form of limiter to avoid oscillations. The q_t term is replaced by $-(f(q)_x + g(q)_y)$ and then $f(q)_x = f'(q)q_x$ is approximated again using the x -slope, together with the Jacobian matrix and a characteristic extrapolation scheme while the $g(q)_y$ term is approximated by differencing Godunov fluxes in the y -direction. This is just a brief sketch of the full algorithm; for details consult [13].

In the wave-propagation approach we start by solving a Riemann problem between states $q_{i-1,j}$ and q_{ij} . The information obtained is used to generate both the first-order updates and also the second-order corrections, since the jump across each wave, divided by Δx , gives a characteristic splitting of an approximation of q_x quite naturally. Transverse derivatives are defined by characteristic splitting of these waves in the transverse direction.

The total amount of work appears to be similar between the two methods. With the “interpolate first” approach, at each grid interface one must do a characteristic decomposition of the slope q_x (analogous to a Riemann solve), solve a Riemann problem in the transverse direction to approximate $g(q)_y$, and finally solve the Riemann problem between the states q^- and q^+ . With the wave-propagation algorithm one first solves a normal Riemann problem and then two transverse Riemann problems.

The wave-propagation algorithms appear to be a more direct generalization of one-dimensional algorithms. They generalize easily to other hyperbolic conservation laws, and also allows extension to nonconservative equations and other problems as presented here.

6. CONCLUSIONS

A class of multidimensional wave propagation algorithms has been developed that incorporates extensions of standard flux-limiter methods in several ways:

- The notion of fluctuation splitting was introduced to generalize flux-difference splitting to hyperbolic systems of equations that are not in conservation form.

- Capacity-form differencing was discussed as a general framework for handling equations of the form (23) or (24). Applications to stratified flow and curvilinear grids were discussed.

- Transverse fluctuation splitting was introduced as a general procedure to give up-winding of the cross-derivative terms needed for second-order accuracy in a simple manner that is broadly applicable. This treatment results in methods that are stable up to Courant number 1.

- The resulting algorithms can be implemented in a very general form that allows their application on a wide variety of hyperbolic problems. This has been done in the CLAWPACK software.

ACKNOWLEDGMENTS

My work on multidimensional wave propagation methods has been influenced by many people over the years, too numerous to mention here. I am particularly indebted to Marsha Berger and Jan Olav Langseth for their interest and advice and to Piotr Smolarkeiwicz for encouraging me to develop a general “capacity-form” differencing approach by pointing out its utility in stratified flow and mapped grids (see [53, 54]).

This work was supported in part by NSF Grants DMS-9204329, DMS-9303404 and DOE Grant DE-FG06-93ER25181. Much of this work was performed while visiting the Scientific Computing Division of the National Center for Atmospheric Research, supported by the National Science Foundation. Special thanks to Paul Swartrauber and Dick Sato for arranging this visit, without which I would never have found time to explore new applications or develop software.

REFERENCES

1. L. Adams, A multigrid algorithm for immersed interface problems, in *Seventh Copper Mountain Conference on Multigrid Methods*, NASA Conference Publication 3339, edited by N. D. Melson, et al. (National Aeronautic and Space Administrator, Washington, DC, 1995), p. 1.
2. L. M. Adams, M. J. Berger, R. J. LeVeque, Z. Li, and K. M. Shyue, in preparation.
3. J. B. Bell, C. N. Dawson, and G. R. Shubin, An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions, *J. Comput. Phys.* **74**, 1 (1988).
4. M. Berger, Ph.D. thesis, Computer Science Department, Stanford University, 1982.
5. M. Berger, *Adaptive Finite Difference Methods In Fluid Dynamics*, Von Karman Institute for Fluid Dynamics Lecture Series, 1987.
6. M. Berger and R. J. LeVeque, Cartesian meshes and adaptive mesh refinement for hyperbolic partial differential equations, in *Proc. Third Int'l Conf. Hyperbolic Problems, Uppsala, 1990*.
7. M. Berger and R. J. LeVeque, Stable boundary conditions for Cartesian grid calculations, *Comput. Systems Eng.* **1**, 305 (1990).
8. M. Berger and J. Olinger, Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* **53**, 484 (1984).
9. M. J. Berger, On conservation at grid interfaces, *SIAM J. Num. Anal.* **24**, 967 (1987).
10. M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* **82**, 64 (1989).
11. M. J. Berger and R. J. LeVeque, Adaptive mesh refinement for two-dimensional hyperbolic systems and the AMRCLAW software, in preparation.
12. M. J. Berger and R. J. LeVeque, *AMRCLAW software*, <http://www.amath.washington.edu/~rjl/amrclaw/>.
13. P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, *J. Comput. Phys.* **87**, 171 (1990).
14. R. Courant and K. O. Friedrichs, *Supersonic Flow and Shock Waves* (Springer-Verlag, New York/Berlin, 1948).
15. H. Deconinck, C. Hirsch, and J. Peuteman, Characteristic decomposition methods for the multidimensional Euler equations, in *10th Int. Conf. on Num. Meth. in Fluid Dyn.*, Lecture Notes in Phys. Vol. 264 (Springer-Verlag, New York/Berlin, 1986), 216.
16. H. Deconinck, R. Struijs, and P. L. Roe, Fluctuation splitting for multidimensional convection problems: An alternative to finite volume and finite element methods, in *VKI Lecture Series 1990-3* (Von Karman Institute, Brussels, 1990).
17. M. Fey, *The Method of Transport for Solving the Euler Equations*, SAM Report 1995-15, ETH-Zürich, 1995.
18. M. Fey, R. Jeltsch, and A.-T. Morel, Multidimensional schemes for nonlinear systems of hyperbolic conservation laws, in *16th Biannual Dundee Conference on Numerical Analysis, 1995*, edited by D. F. Griffiths and G. A. Watson (Longman, Harlow/New York, 1996).
19. M. Fey and A.-T. Morel, *Multidimensional Method of Transport for the Shallow Water Equations*, SAM Report 1995-05, ETH-Zürich, 1995.
20. J. B. Goodman and R. J. LeVeque, A geometric approach to high resolution TVD schemes, *SIAM J. Num. Anal.* **25**, 268 (1988).
21. A. Harten and J. M. Hyman, Self-adjusting grid methods for one-dimensional hyperbolic conservation laws, *J. Comput. Phys.* **50**, 235 (1983).
22. C. Hirsch, C. Lacor, and H. Deconinck, AIAA Paper 87-1163, 1987.
23. J. O. Langseth and R. J. LeVeque, Three-dimensional Euler computations using CLAWPACK, in *Conf. on Numer. Meth. for Euler and Navier-Stokes Eq., Montreal, 1995*. edited by P. Arminjon, to appear.
24. J. O. Langseth and R. J. LeVeque, in preparation.
25. P. D. Lax, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, SIAM Regional Conference Series in Applied Mathematics, Vol. 11 (SIAM, Philadelphia, 1972).
26. P. D. Lax and X. D. Liu, Solution of two dimensional Riemann problem of gas dynamics by positive schemes, *SIAM J. Sci. Comput.*, to appear.
27. B. P. Leonard, M. K. MacVean, and A. P. Lock, NASA Technical Memorandum 106055, ICOMP-93-05, 1993.
28. R. J. LeVeque, *CLAWPACK software*, available from [netlib.att.com](http://netlib.att.com/netlib/pdes/claw/) in netlib/pdes/claw/, or on the Web at the URL <http://www.amath.washington.edu/~rjl/clawpack.html>.
29. R. J. LeVeque, *CLAWPACK User Notes*, available from netlib.att.com in netlib/pdes/claw/doc/, or on the Web at the URL <http://www.amath.washington.edu/~rjl/clawpack.html>.

30. R. J. LeVeque, High resolution finite volume methods on arbitrary grids via wave propagation, *J. Comput. Phys.* **78**, 36 (1988).
31. R. J. LeVeque, *Hyperbolic Conservation Laws and Numerical Methods*, Lecture Series 90-03, 1990.
32. R. J. LeVeque, *Numerical Methods for Conservation Laws* (Birkhäuser, Basel, 1990).
33. R. J. LeVeque, Simplified multi-dimensional flux limiter methods, in *Numerical Methods for Fluid Dynamics 4*, edited by M. J. Baines and K. W. Morton (Oxford Univ. Press, Oxford, 1993), 175.
34. R. J. LeVeque, CLAWPACK—A software package for solving multi-dimensional conservation laws, in *Proceedings of the Fifth International Conference on Hyperbolic Problems: Theory, Numerics, Applications*, edited by J. Glimm *et al.* (World Scientific, Singapore, 1994), p. 188.
35. R. J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, *SIAM J. Numer. Anal.* **33**, 627 (1996).
36. R. J. LeVeque and Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, *SIAM J. Sci. Comput.*, to appear (1997).
37. R. J. LeVeque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* **31**, 1019 (1994).
38. R. J. LeVeque and C. Zhang, Immersed interface methods for wave equations with discontinuous coefficients, *Wave Motion*, to appear (1997).
39. Z. Li, *The Immersed Interface Method—A Numerical Approach for Partial Differential Equations with Interfaces*, Ph.D. thesis, University of Washington, 1994.
40. X. D. Liu and P. D. Lax, Positive schemes for solving multi-dimensional hyperbolic systems of conservation laws, *Comput. Fluid Dynamics J.* **5**(2), 133 (July 1996).
41. Y. B. Radvogin, Soviet Academy of Sciences preprint, 1991. [Russian]
42. P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* **43**, 357 (1981).
43. P. L. Roe, Fluctuations and signals—A framework for numerical evolution problems, in *Numerical Methods for Fluid Dynamics*, edited by K. W. Morton and M. J. Baines (Academic Press, New York, 1982), p. 219.
44. P. L. Roe, Upwind schemes using various formulations of the Euler equations, in *Numerical Methods for the Euler Equations of Fluid Dynamics*, edited by F. Angrand *et al.*, INRIA Workshop (SIAM, Philadelphia, 1985), 14.
45. P. L. Roe, Sonic flux formulae, *SIAM J. Sci. Stat. Comput.* **13**, 611 (1992).
46. J. Saltzman, Los Alamos Report LA-UR-87-2479, 1987.
47. J. Saltzman, An unsplit 3-D upwind method for hyperbolic conservation laws, *J. Comput. Phys.* **115**, 153 (1994).
48. C. W. Schultz-Rinne, Classification of the Riemann problem for two-dimensional gas dynamics, *SIAM J. Sci. Comput.* **14**, 1394 (1993).
49. C. W. Schulz-Rinne, J. P. Collins, and H. M. Glaz, Numerical solution of the Riemann problem for two-dimensional gas dynamics, *SIAM J. Sci. Comput.* **14**, 1394 (1993).
50. D. Sidilkover, A genuinely multidimensional upwind scheme for the compressible Euler equations, in *Proceedings of the Fifth International Conference on Hyperbolic Problems: Theory, Numerics, Applications*, edited by J. Glimm *et al.* (World Scientific, Singapore, 1994), p. 447.
51. D. Sidilkover, ICASE Report No. 94-84, NASA Langley Research Center, 1994.
52. D. Sidilkover, Multidimensional upwinding and multigrid, in *AIAA 95-1759, June 19–22 1995, 12th AIAA CFD meeting, San Diego*.
53. P. K. Smolarkiewicz and T. L. Clark, The multidimensional positive definite advection transport algorithm: further development and applications, *J. Comput. Phys.* **67**, 396 (1986).
54. P. K. Smolarkiewicz and L. G. Margolin, On forward-in-time differencing for fluids: extension to a curvilinear framework, *Mon. Weather Rev.* **121**, 1847 (1993).
55. G. Strang, On the construction and comparison of difference schemes, *SIAM J. Num. Anal.* **5**, 506 (1968).
56. J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations* (Wadsworth & Brooks/Cole, 1989).
57. R. Struijs, H. Deconinck, P. de Palma, P. L. Roe, and K. G. Powell, Progress on multidimensional upwind Euler solvers for unstructured grids, in *AIAA Conference on Computational Fluid Dynamics*, CP-91-1550, 1991.
58. P. K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM J. Num. Anal.* **21**, 995 (1984).
59. J. A. Trangenstein, preprint, 1993.
60. B. van Leer, Towards the ultimate conservative difference scheme V. A second-order sequel to Godunov's method, *J. Comput. Phys.* **32**, 101 (1979).
61. S. T. Zalesak, Fully multidimensional flux corrected transport algorithms for fluids. *J. Comput. Phys.* **31**, 335 (1979).
62. S. T. Zalesak, A preliminary comparison of modern shock-capturing schemes: linear advection, in *Advances in Computer Methods for Partial Differential Equations, VI*, edited by R. Vichnevetsky and R. S. Stepleman (IMACS, Baltzer, Basel, 1987), p. 15.
63. C. Zhang, *Immersed Interface Methods for Wave Equations*, Ph.D. thesis. University of Washington, 1996.