

HIGH-RESOLUTION CONSERVATIVE ALGORITHMS FOR ADVECTION IN INCOMPRESSIBLE FLOW*

RANDALL J. LEVEQUE†

Abstract. A class of high-resolution algorithms is developed for advection of a scalar quantity in a given incompressible flow field in one, two, or three space dimensions. Multidimensional transport is modeled using a wave-propagation approach in which the flux at each cell interface is built up on the basis of information propagating in the direction of this interface from neighboring cells. A high-resolution second-order method using slope limiters is quite easy to implement. For constant flow, a minor modification gives a third-order accurate method. These methods are stable for Courant numbers up to 1. Fortran implementations are available by anonymous ftp.

Key words. multidimensional advection, incompressible flow, high-resolution methods, finite volume methods, flux-limiters, wave-propagation methods

AMS subject classifications. 65M06, 76M20

1. Introduction. We consider the advection of a scalar concentration or density function $q(\vec{x}, t)$ in a specified velocity field $\vec{u}(\vec{x}, t)$ in one, two, or three space dimensions. The evolution of q is governed by the conservation law

$$(1.1) \quad q_t + \nabla \cdot (\vec{u}q) = 0.$$

We assume that the flow is incompressible, so that

$$(1.2) \quad \nabla \cdot \vec{u}(\vec{x}, t) = 0$$

everywhere. Then the equation (1.1) can be written equivalently as a variable coefficient advection equation

$$(1.3) \quad q_t + \vec{u} \cdot \nabla q = 0.$$

The goal of this paper is to present a very simple framework for developing a hierarchy of methods for the numerical solution of this equation in several space dimensions, starting with the most basic upwind method for (1.3) and adding in a sequence of simple correction terms to achieve better accuracy and stability properties. Differencing based on the advective form (1.3) is often more successful than conservative differencing for this problem (see §4), although this can often cause nonconservative behavior and a change in the total mass that is unacceptable in some problems. The algorithms developed here are in a sense hybrid algorithms in which the first-order upwind method is in advective form but all of the corrections, while based on advective differences, are written in a flux-differencing form. The result is an algorithm that does not suffer the usual difficulties associated with conservative differencing and yet is fully conservative provided that a natural discrete form of the incompressibility constraint (1.2) is satisfied (given by (4.2)).

In particular, a high-resolution method is developed that is second-order accurate when \vec{u} and q are smooth and that also computes sharply resolved solutions when q is discontinuous or has steep gradients. Although not strictly sign preserving or total variation diminishing,

*Received by the editors January 12, 1994; accepted for publication (in revised form) June 14, 1994. This work was supported in part by National Science Foundation grants DMS- 8657319 and DMS-9303404, and Department of Energy grant DE-FG06- 93ER25181.

†Departments of Mathematics and Applied Mathematics, University of Washington, Seattle, WA 98195 (rjl@amath.washington.edu).

the algorithm produces virtually no oscillations or undershoots on a variety of difficult test problems. The method is stable provided

$$\frac{k}{h} \max_{\vec{x}, t} \|\vec{u}(\vec{x}, t)\|_{\infty} \leq 1,$$

where k is the time step, h is the grid spacing. (For simplicity, the mesh spacing is assumed to be uniform and equal in all directions. This is not necessary and the algorithms generalize in the obvious way to nonuniform grid spacings.)

Additional correction terms are also discussed that give a third-order accurate method in the special case where \vec{u} is constant in space and time. These terms can give some improvement in the more general case as well, although there are still some unresolved issues regarding appropriate limiters.

Countless advection algorithms have been developed in recent years by researchers working in various application areas. Some of these contributions are briefly described in §5 and are compared to the present approach. The advection algorithms developed here are adapted from the multidimensional methods for nonlinear systems of conservation laws developed in previous work by the author [29] (see also [26, 27]) and are based on a wave-propagation viewpoint that gives a natural geometrical interpretation to the various high-order correction terms introduced. It also leads to a method that is very simple to implement in spite of multidimensional upwinding of a nature that could be quite difficult to implement otherwise.

This paper is organized as follows. In §2 the one-dimensional algorithm is presented which is essentially a standard flux-limiter method but interpreted in the wave-propagation form that will be valuable later. In §3 the constant coefficient problem in two dimensions is discussed, and the algorithm is then extended to arbitrary incompressible flow in §4. A brief survey of other methods is given in §5. A truncation error analysis is given in §6 that demonstrates second-order accuracy and motivates the third-order correction terms presented in §7. Stability is discussed in §8 and stability regions are shown for various methods. A variety of numerical results are presented in §9 that confirm the theoretical predictions of accuracy and illustrate the power of the method. Boundary conditions are discussed in §10 and extension to three space dimensions is presented in §11.

2. One space dimension. We start by reviewing the form of a high-resolution method for the advection equation in one space dimension. The ideas presented here will be extended directly to two dimensions in the next section. See [28] for more details on one-dimensional algorithms of this form.

In one dimension the incompressibility constraint (1.2) requires $u_x = 0$ and so $u = \text{const}$ and we simply have the constant coefficient advection equation

$$q_t + uq_x = 0.$$

Since u is constant, this can be written equivalently in conservation form as

$$(2.1) \quad q_t + (uq)_x = 0 \quad \text{or} \quad q_t + f(q)_x = 0,$$

where $f(q) = uq$ is the flux function giving the rate of flux of q per unit time. We use a finite volume method in which q_i^n represents an approximation to the cell average of q over the i th cell $C_i = [x_{i-1/2}, x_{i+1/2}]$:

$$q_i^n \approx \frac{1}{h} \int_{C_i} q(x, t_n) dx.$$

Integrating the conservation law (2.1) over $C_i \times [t_n, t_{n+1}]$ gives

$$\frac{1}{h} \int_{C_i} q(x, t_{n+1}) dx = \frac{1}{h} \int_{C_i} q(x, t_n) dx - \frac{k}{h} \left[\frac{1}{k} \int_{t_n}^{t_{n+1}} f(q(x_{i+1/2}, t)) dt - \frac{1}{k} \int_{t_n}^{t_{n+1}} f(q(x_{i-1/2}, t)) dt \right].$$

A finite volume method in conservation form takes the form

$$(2.2) \quad q_i^{n+1} = q_i^n - \frac{k}{h} [F(q^n; i + 1) - F(q^n; i)],$$

where $F(q^n; i)$ is some approximation to the average flux $\frac{1}{k} \int_{t_n}^{t_{n+1}} f(q(x_{i-1/2}, t)) dt$ based on the data q^n at time t_n . For brevity the superscript n will often be left off and it is understood that all data is at time t_n unless otherwise stated.

2.1. Flux-limiter methods. Two standard methods are the first-order *upwind method* in which

$$(2.3) \quad F^{up}(q; i) = \begin{cases} uq_{i-1}, & \text{if } u > 0, \\ uq_i, & \text{if } u < 0, \end{cases}$$

and the second-order *Lax-Wendroff method* in which

$$(2.4) \quad F^{LW}(q; i) = \frac{1}{2}u(q_{i-1} + q_i) - \frac{k}{2h}u^2(q_i - q_{i-1}).$$

The upwind method has excessive numerical dissipation and typically exhibits strong smearing of solutions and low accuracy and resolution. The Lax-Wendroff method can work well on very smooth data but has difficulties if q has steep gradients or discontinuities since it is very dispersive and tends to generate oscillations, also destroying the accuracy. Much better results can be obtained by using a hybrid method that uses the second-order flux in smooth regions but involves some sort of *limiting* based on the gradient of the solution so that near discontinuities it reduces to the monotone upwind method. Note that the Lax-Wendroff flux (2.4) can be decomposed into the upwind flux plus a correction term:

$$F^{LW}(q; i) = F^{up}(q; i) + \frac{1}{2}|u| \left(1 - \frac{k}{h}|u| \right) (q_i - q_{i-1}).$$

This suggests the following flux-limiter method:

$$(2.5) \quad F(q; i) = F^{up}(q; i) + \frac{1}{2}|u| \left(1 - \frac{k}{h}|u| \right) (q_i - q_{i-1}) \Phi_i,$$

where Φ_i is the limiter that depends on the nature of the solution locally. Note that if $\Phi = 0$ then we have the upwind method while if $\Phi = 1$ we have Lax-Wendroff. The limiters we will use here have the form

$$\Phi_i = \phi(\theta_i), \quad \theta_i = \frac{q_I - q_{I-1}}{q_i - q_{i-1}},$$

where

$$I = \begin{cases} i - 1, & \text{if } u > 0, \\ i + 1, & \text{if } u \leq 0. \end{cases}$$

We see that θ_i is the ratio of the slope at the neighboring interface in the upwind direction to the slope at the current interface. Some standard limiters are

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

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

$$\text{van Leer: } \phi(\theta) = \left(\frac{\theta + |\theta|}{1 + |\theta|} \right),$$

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

Note in particular that for minmod, $(q_i - q_{i-1})\phi(\theta_i)$ is simply equal to either $q_i - q_{i-1}$ or $q_I - q_{I-1}$, whichever is smaller in magnitude, unless they have opposite signs (i.e., $\theta_i < 0$) in which case $\phi = 0$.

The van Leer limiter was introduced in [59] and the MC limiter was also introduced by van Leer in a later paper in this series [60]. This latter limiter produces the centered approximation

$$(q_i - q_{i-1})\phi(\theta_i) = \frac{1}{2}((q_i - q_{i-1}) + (q_I - q_{I-1}))$$

unless this is larger than $2(q_i - q_{i-1})$ or $2(q_I - q_{I-1})$, in which case it is appropriately limited. (And again $\phi = 0$ if $\theta < 0$.) The MC limiter seems to be quite a good choice in general. The superbee limiter [41] tends to be “overcompressive”, meaning that it tends to steepen up smooth profiles into discontinuities. For this reason it is useful for problems where q should have a sharp discontinuity that we wish to maintain, but may be inappropriate for problems with smooth q . Some examples will be seen in §9.

The theory of flux-limiters is discussed more fully in Sweby [57] (see also [28]). More recent discussions of limiters that include some potentially valuable extensions include [21] and [47]. Some different approaches to multidimensional limiting are mentioned in §5.

2.2. Slope-limiter methods. The flux-limiter method described above can be viewed in a more geometric way that facilitates the extension to two space dimensions. A quite general class of methods can be derived by the following sequence of steps:

1. From the given cell averages q_i at time t_n construct a function $\tilde{q}(x, t_n)$.
2. Solve the advection equation exactly with this data over a time step of length k , giving $\tilde{q}(x, t_{n+1}) = \tilde{q}(x - uk, t_n)$.
3. Average this shifted function over the grid cells to obtain the new cell averages:

$$q_i^{n+1} = \frac{1}{h} \int_{C_i} \tilde{q}(x, t_{n+1}) dx.$$

This will be referred to as the “shift-and-average” algorithm. The particular method obtained depends on the form of the function \tilde{q} chosen in step 1. If \tilde{q} is piecewise constant with value q_i in cell C_i , we obtain the upwind algorithm. If \tilde{q} is piecewise linear with mean value q_i and some slope σ_i in cell C_i , then it is easy to verify that the resulting method is conservative for any choice of σ_i and can be written in the form (2.2) with the flux

$$(2.6) \quad F(q; i) = F^{up}(q; i) + \frac{1}{2}|u| \left(1 - \frac{k}{h}|u| \right) h\sigma_J,$$

where

$$J = \begin{cases} i - 1, & \text{if } u > 0, \\ i, & \text{if } u < 0. \end{cases}$$

In particular, the choice $\sigma_J = (q_i - q_{i-1})/h$ gives the Lax-Wendroff method. The flux-limiter method (2.5) is easily interpreted as a slope-limiter method with slope $\sigma_J = (\frac{q_i - q_{i-1}}{h})\Phi_i$.

The slope-limiter viewpoint makes it relatively easy to interpret the effect of the limiter in terms of the requirement that the algorithm maintain monotonicity and not increase the total variation of q in any time step. This approach was used by van Leer in his development of the MUSCL scheme for conservation laws, e.g., [60].

2.3. Wave-propagation form. These algorithms can be viewed in yet another way that will also prove useful. In the case where \tilde{q} is piecewise constant (the upwind method), we can view the discontinuity in \tilde{q} at the cell interface $x_{i-1/2}$ as giving rise to a wave that propagates into cell C_i (resp., C_{i-1}) if $u > 0$ (resp., $u < 0$) and modifies the value of q in this cell by the jump $(q_i - q_{i-1})$ as it passes through. After time k it has propagated a distance uk and so the cell average is modified by $\frac{uk}{h}(q_i - q_{i-1})$. This gives

$$q_i^{n+1} = q_i - \frac{k}{h}u(q_i - q_{i-1}),$$

where

$$I = \begin{cases} i, & \text{if } u > 0, \\ i - 1, & \text{if } u < 0. \end{cases}$$

This agrees with the upwind method defined before.

To introduce slopes, we can think of replacing the piecewise constant wave shown in Figure 2.1(a) by a piecewise linear function as shown in Figure 2.1(b). For concreteness we assume $u > 0$ and so the wave originating from the interface $x_{i-1/2}$ affects the cell average q_i which is updated by the shaded area in Figure 2.1(b) divided by the cell length h . If the slope in cell C_{i-1} is σ_{i-1} then the area of the shaded trapezoid is

$$\begin{aligned} & \frac{1}{2}ku \left[\left(q_{i-1} + \frac{h}{2}\sigma_{i-1} - q_i \right) + \left(q_{i-1} + \left(\frac{h}{2} - ku \right) \sigma_{i-1} - q_i \right) \right] \\ & = -ku \left[(q_i - q_{i-1}) - \frac{1}{2} \left(1 - \frac{k}{h}u \right) h\sigma_{i-1} \right]. \end{aligned}$$

The value q_i is also affected by the wave originating in cell C_i that moves to the right. If the slope σ_i is nonzero then this wave changes the cell average q_i by $1/h$ times

$$\frac{1}{2}ku \left(1 - \frac{k}{h}u \right) h\sigma_i.$$

The total update to q_i comes from combining these two terms and gives

$$q_i^{n+1} = q_i^n - \frac{k}{h}u(q_i - q_{i-1}) - \frac{1}{2} \frac{k}{h}u \left(1 - \frac{k}{h}u \right) h(\sigma_i - \sigma_{i-1}).$$

This is the same formula as obtained by using the flux (2.6).

This can also be viewed as a two-step procedure in which we first modify q_i by propagating the piecewise constant wave of Figure 2.1(a) (called the *increment wave*) and then propagate a *correction wave* of the form shown in Figure 2.1(c) that has slope σ_{i-1} but mean value zero.

This decomposition will be useful in two dimensions where these two waves are best viewed as separate entities which may even propagate in different directions.

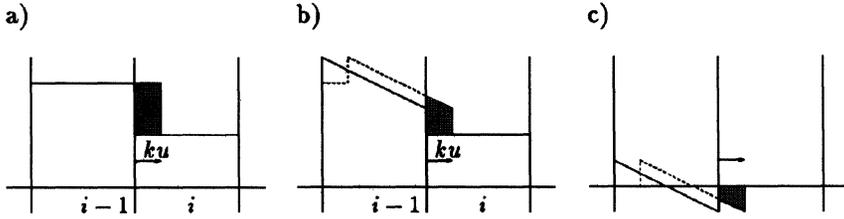


FIG. 2.1. (a) Propagation of a piecewise constant wave into the i th cell (the increment wave). (b) Propagation of a piecewise linear wave. (c) The correction wave. The wave shown in (b) can be decomposed into the increment wave and the correction wave.

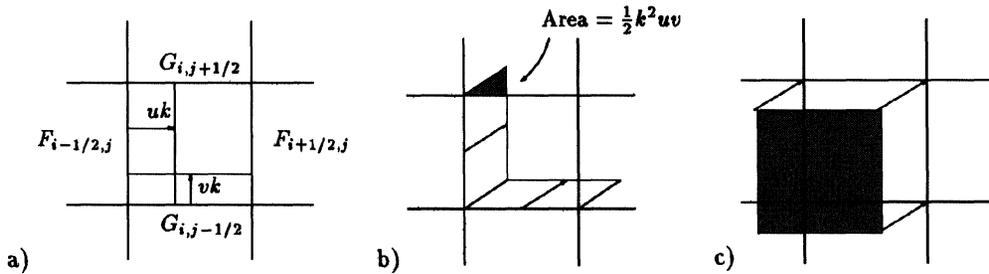


FIG. 3.1. (a) Wave-propagation interpretation of the upwind method with fluxes (3.2). (b) Wave propagation for the modified upwind method with fluxes corrected by (3.3) and (3.4). (c) The new cell value can equivalently be computed as the weighted average of four cell values overlapped by the shaded region.

3. Two-dimensional constant flow. We now extend the above method to two space dimensions in the case where u and v are constant. We assume a uniform grid with equal spacing h in both directions. Let C_{ij} be the (i, j) grid cell $[x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ and let q_{ij}^n represent an approximation to the cell average,

$$q_{ij}^n \approx \frac{1}{h^2} \int_{C_{ij}} q(x, y, t_n) dx dy.$$

For concreteness in describing the algorithm we will assume that u and v are positive throughout this section. The general algorithm is presented in §4.

A conservative finite volume method in flux-differencing form now takes the form

$$(3.1) \quad q_{ij}^{n+1} = q_{ij} - \frac{k}{h} [F_{i+1/2,j} - F_{i-1/2,j} + G_{i,j+1/2} - G_{i,j-1/2}],$$

where $F_{i-1/2,j}$ represents the flux at the left edge of cell C_{ij} and $G_{i,j-1/2}$ is the flux at the bottom. The simplest upwind method (the “donor-cell” method) would use

$$(3.2) \quad \begin{aligned} F_{i-1/2,j} &= uq_{i-1,j}, \\ G_{i,j-1/2} &= vq_{i,j-1}. \end{aligned}$$

Note that as a wave-propagation method this has the interpretation shown in Figure 3.1(a). Waves carrying the jumps $(q_{ij} - q_{i-1,j})$ and $(q_{ij} - q_{i,j-1})$ propagate independently into the cell in the x - and y -directions at speeds given by the velocities u and v in the directions normal to each interface.

Clearly a superior method should be obtained by propagating each of these waves at the proper speed (u, v) oblique to the grid, as shown in Figure 3.1(b). This can be most

easily implemented as a two-step procedure in which the wave is first propagated normal to the interface, giving a provisional value for the flux at the interface as above, and then the triangular piece of the wave that moves into an adjacent cell is used to update the flux between the two cells affected by this transverse motion. Note that the area of this triangle is $\frac{1}{2}k^2uv$ so that the cell average is modified by $\frac{1}{2}\frac{k^2}{h^2}uv\Delta q$, where Δq is the jump across the wave. One cell average is increased by this amount while the other is decreased by this same amount, and so this transfer can be accomplished by modifying the flux $G_{i,j+1/2}$ by $\frac{1}{2}\frac{k}{h}uv\Delta q$. (The other factor of k/h appears in the flux-differencing expression (3.1).)

Because the wave from each interface now affects two different fluxes, this is most easily implemented by initializing all $F_{i-1/2,j}$ and $G_{i,j-1/2}$ to zero and then looping over the cell interfaces, updating the appropriate fluxes as needed while dealing with each interface in turn. The wave propagating from the interface between cells $C_{i-1,j}$ and C_{ij} affects the fluxes $F_{i-1/2,j}$ and $G_{i,j+1/2}$ (recall $u, v > 0$) by

$$(3.3) \quad \begin{aligned} F_{i-1/2,j} &:= F_{i-1/2,j} + uq_{i-1,j}, \\ G_{i,j+1/2} &:= G_{i,j+1/2} - \frac{1}{2}\frac{k}{h}uv(q_{ij} - q_{i-1,j}). \end{aligned}$$

Similarly, the wave from the interface between $C_{i,j-1}$ and C_{ij} leads to the updates

$$(3.4) \quad \begin{aligned} G_{i,j-1/2} &:= G_{i,j-1/2} + vq_{i,j-1}, \\ F_{i+1/2,j} &:= F_{i+1/2,j} - \frac{1}{2}\frac{k}{h}uv(q_{ij} - q_{i,j-1}). \end{aligned}$$

This modified version of the upwind method is both more accurate and also more stable than the original version (3.2). The original method requires

$$(3.5) \quad \frac{k}{h}(|u| + |v|) \leq 1,$$

whereas the modified method only requires

$$(3.6) \quad \frac{k}{h} \max(|u|, |v|) \leq 1.$$

For advection at 45 degrees to the grid this allows a doubling of the time step. These well-known bounds are derived in §8 and the stability regions are shown in Figure 8.1. This improved first-order accurate method is called the corner transport upwind (CTU) method by Colella [8].

An easy computation shows that this method can also be interpreted as a shift-and-average method, extended in the obvious way to two dimensions, in the case where the function $\tilde{q}(x, y, t_n)$ is chosen to be piecewise constant with value q_{ij} in cell C_{ij} . The resulting value q_{ij}^{n+1} is an average of q_{ij}^n over four neighboring cells as indicated in Figure 3.1(c), but is implemented in quite a different way that simplifies both implementation and also extension to higher accuracy.

3.1. Second-order accuracy. To achieve second-order accuracy we need to model second-derivative terms. Following the Lax-Wendroff approach to developing second-order methods, we expand $q(x, y, t_{n+1})$ in a Taylor series in time:

$$q(x, y, t_{n+1}) = q(x, y, t_n) + kq_t(x, y, t_n) + \frac{1}{2}k^2q_{tt}(x, y, t_n) + O(k^3).$$

In the constant coefficient problem,

$$q_t = -uq_x - vq_y$$

and

$$q_{tt} = u^2q_{xx} + 2uvq_{xy} + v^2q_{yy}.$$

The idea is now to approximate the spatial derivatives by finite differences. The standard Lax-Wendroff method is obtained by using centered approximations to each of the derivatives. The basic upwind method approximates only the kq_t term using first-order upwind approximations to kuq_x and kvq_y . The CTU method with transverse propagation of the waves introduces an approximation to the cross-derivative term k^2uvq_{xy} . Note, for example, that the G -flux modification in (3.3) involves an approximation to q_x . When the G 's are differenced in the y -direction in (3.1), this gives the q_{xy} term. Similarly, the modification to F in (3.4), when differenced in the x -direction, gives an approximation to q_{yx} which also contributes to the cross-derivative term.

To obtain second-order accuracy, we need to replace the upwind approximations to kuq_x and kvq_y by centered approximations and also to introduce approximations to $\frac{1}{2}k^2u^2q_{xx}$ and $\frac{1}{2}k^2v^2q_{yy}$. These are both accomplished by using the same flux corrections used in the one-dimensional method. In considering the interface between $C_{i-1,j}$ and C_{ij} , in addition to the flux modification (3.3), we also perform

$$(3.7) \quad F_{i-1/2,j} := F_{i-1/2,j} + \frac{1}{2}|u| \left(1 - \frac{k}{h}|u| \right) (q_{ij} - q_{i-1,j}),$$

regardless of the sign of u . If a limiter is needed to avoid oscillations, we replace $(q_{ij} - q_{i-1,j})$ in this step by a limited version as described in §2. Note that this limiter should be applied only in this term, not in the modification to $G_{i,j+1/2}$ in (3.3). It is not needed in (3.3) because that term only introduces transverse propagation of the piecewise constant wave to give the CTU method, a monotone method that does not suffer from oscillations.

An analogous modification is made following (3.4):

$$(3.8) \quad G_{i,j-1/2} := G_{i,j-1/2} + \frac{1}{2}|v| \left(1 - \frac{k}{h}|v| \right) (q_{ij} - q_{i,j-1}),$$

perhaps with a limiter. In the absence of limiters, the method just described is second-order accurate on smooth data. This is verified by a more careful error analysis in §6 and by numerical results in §9.

These second-order corrections are exactly the same as the corresponding terms in the standard Lax-Wendroff method [19, 56] for the constant coefficient problem (except for the limiter). Note, however, that the cross-derivative terms are modeled in a different way. The approximations used here are one-sided approximations based on the direction (u, v) while the standard method uses centered approximations. The standard Lax-Wendroff method is obtained if the modification $-\frac{1}{2}\frac{k}{h}uv(q_{ij} - q_{i-1,j})$ from (3.3) is split evenly between the four fluxes $G_{i-1,j-1/2}$, $G_{i-1,j+1/2}$, $G_{i,j-1/2}$, and $G_{i,j+1/2}$ (modifying each by $-\frac{1}{8}\frac{k}{h}uv(q_{ij} - q_{i-1,j})$) rather than being applied only in the upwind direction to $G_{i,j+1/2}$. The other terms are modeled identically in the two versions, but this upwind modification of the cross-derivative terms is enough to increase the stability bound so that time steps k satisfying (3.6) are allowed, just as with the first-order version. By contrast, the standard Lax-Wendroff method has a time-step restriction even more stringent than (3.5). The stability region is shown in Figure 8.1. The version presented here allows larger time steps in addition to allowing the easy introduction of limiters to give high-resolution results for nonsmooth data.

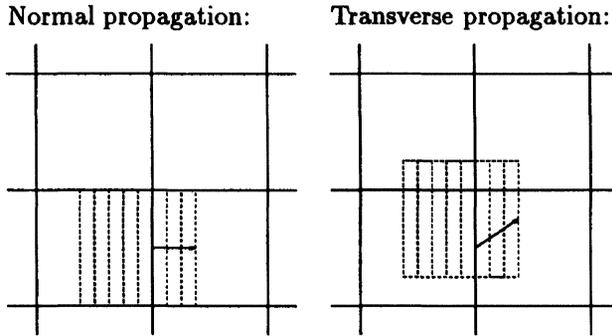


FIG. 3.2. Propagation of the correction wave in the flow direction. The vertical dashed lines are contour lines of the correction wave.

Although second-order accuracy has already been achieved, one might suspect that even better results would be obtained by also including transverse propagation of the correction waves, just as better first-order results are obtained by including tangential propagation of the increment waves. This turns out to be true and is also quite easy to implement. The correction wave affects two cells (recall Figure 2.1(c)) and so the transverse motion of this wave will affect two fluxes in the transverse direction. Figure 3.2 shows the correction wave from the interface between $C_{i-1,j}$ and C_{ij} moving distance (ku, kv) . It modifies the flux $F_{i-1/2,j}$ according to (3.7) but now also modifies $G_{i-1,j+1/2}$ and $G_{i,j+1/2}$ by (assuming again $u, v > 0$ for concreteness)

$$(3.9) \quad G_{i-1,j+1/2} := G_{i-1,j+1/2} - \frac{1}{2} \frac{k}{h} uv \left(1 - \frac{k}{h} u \right) (q_{ij} - q_{i-1,j}),$$

$$G_{i,j+1/2} := G_{i,j+1/2} + \frac{1}{2} \frac{k}{h} uv \left(1 - \frac{k}{h} u \right) (q_{ij} - q_{i-1,j}).$$

Note that the modification to each G is just $\pm kv/h$ times the modification to the F made in (3.7), making it quite trivial to compute and reflecting the fact that this wave is moving upward at speed kv and hence moving through the fraction kv/h of the neighboring cell.

Similarly, following (3.8) we can perform

$$(3.10) \quad F_{i+1/2,j-1} := F_{i+1/2,j-1} - \frac{1}{2} \frac{k}{h} uv \left(1 - \frac{k}{h} v \right) (q_{ij} - q_{i,j-1}),$$

$$F_{i+1/2,j} := F_{i+1/2,j} + \frac{1}{2} \frac{k}{h} uv \left(1 - \frac{k}{h} v \right) (q_{ij} - q_{i,j-1}).$$

These modifications do not affect the formal order of accuracy of the method but can reduce the error, particularly on problems with steep gradients or discontinuities in q where it can help to minimize the mild oscillations that may arise in two dimensions even when limiters are used. (An example is given in §9.) If limiters are used, then limited values of the jumps in q are used in (3.9) and (3.10), just as in (3.7) and (3.8).

The above formulas assume that $u, v > 0$. The general case can be handled with only minor changes to the formulas and some logic to determine which fluxes are affected based on the direction of propagation. The general algorithm is presented as Algorithm 4.1 in §4 after introducing the modifications needed to deal with variables u and v .

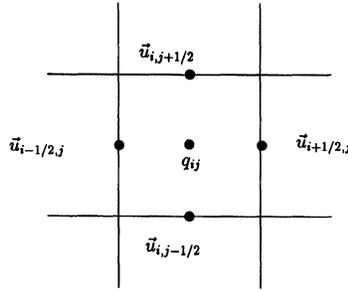


FIG. 4.1. The interface velocity values.

4. Two-dimensional nonconstant flow. We now consider a more general specified velocity field $(u(x, y, t), v(x, y, t))$ which is assumed to be incompressible:

$$(4.1) \quad u_x(x, y, t) + v_y(x, y, t) = 0 \quad \text{for all } x, y, t.$$

The method presented above can be generalized quite easily by simply replacing u and v in most of the formulas by the values of u and v at the midpoint of the interface giving rise to the wave being propagated. We thus need values of u and v at the points $(x_{i\pm 1/2}, y_j, t_{n+1/2})$ and $(x_i, y_{j\pm 1/2}, t_{n+1/2})$. These are also evaluated at the midpoint in time $t_{n+1/2} = t_n + k/2$ to preserve second-order accuracy in the case where the flow is time dependent. For the method described below to be conservative, we require that these discrete values satisfy

$$(4.2) \quad (u_{i+1/2,j}^{n+1/2} - u_{i-1/2,j}^{n+1/2}) + (v_{i,j+1/2}^{n+1/2} - v_{i,j-1/2}^{n+1/2}) = 0.$$

This involves the u and v velocities at the points shown in Figure 4.1 and dividing by h shows that (4.2) is a natural discrete version of (4.1) over cell C_{ij} .

If the velocity field is calculated by an incompressible Navier-Stokes solver that operates on a staggered grid of the form shown in Figure 4.1 (such as the MAC method [15] or the method of Bell, Colella, and Glaz [1]) then this condition may be automatically satisfied by the computed values, at least at the full time steps. Values satisfying (4.2) at the half time steps could be obtained by averaging these values in time. Other algorithms, such as Chorin's original projection method [7] do not use a staggered grid but rather produce cell-centered values (u_{ij}^n, v_{ij}^n) that satisfy

$$(u_{i+1,j}^n - u_{i-1,j}^n) + (v_{i,j+1}^n - v_{i,j-1}^n) = 0.$$

In this case we can define

$$u_{i-1/2,j}^{n+1/2} = \frac{1}{4}(u_{i-1,j}^n + u_{ij}^n + u_{i-1,j}^{n+1} + u_{ij}^{n+1}),$$

$$v_{i,j-1/2}^{n+1/2} = \frac{1}{4}(v_{i,j-1}^n + v_{ij}^n + v_{i,j-1}^{n+1} + v_{ij}^{n+1})$$

and obtain interface values that satisfy (4.2).

Finally, if we are given $u(x, y, t)$ and $v(x, y, t)$ as functions that are divergence-free in the sense of (4.1), then evaluating these functions at the cell interfaces gives discrete velocities that may not satisfy the discrete condition (4.2). However, we could apply a projection to this velocity field to make it divergence-free in the discrete sense with only an $O(h^2)$ modification to the values (see, e.g., [2]).

The advection equation can be written in either the conservative form

$$(4.3) \quad q_t + (qu)_x + (qv)_y = 0$$

or in the advective form

$$(4.4) \quad q_t + uq_x + vq_y = 0.$$

Mathematically the two are equivalent if (4.1) is satisfied, but numerical algorithms may behave quite differently depending on which form is modeled. (See, e.g., [10, 13, 44] for some discussions of this issue.) While it is generally desirable to maintain conservation and so (4.3) might seem preferable, advective-form algorithms often give better accuracy.

One difficulty with the conservative form is easy to appreciate if we consider a problem in which q is constant, say $q \equiv 1$, but u and v are varying. Of course q should remain constant but since the flux functions qu and qv are not constant, this will only be true numerically if the flux differences in (3.1) across each cell happen to cancel out. While they should cancel out in theory because of (4.1), when the fluxes are computed with a high-resolution method involving limiter functions, this may not be the case and constant flow might not be preserved. This intolerable behavior indicates severe difficulties with such methods.

The method presented in the previous section, when properly generalized to nonconstant flows, appears to combine the advantages of both types of algorithm. The method is still written in the flux-differencing form (3.1), and hence is conservative. However, the fluxes are calculated in a form that is essentially advective, in that the formulas involve velocities u or v multiplying q -differences, rather than differences of the quantities qu or qv . Of course for the constant coefficient problem there is no difference since the constant velocities can be factored in or out of the differencing at will.

In the case of nonconstant flow, the basic donor-cell upwind method can be written in two different forms. The natural generalization based on fluxes would be to set (again assuming $u, v > 0$ everywhere)

$$F_{i-1/2,j} = u_{i-1/2,j}^{n+1/2} q_{i-1,j},$$

$$G_{i,j-1/2} = v_{i,j-1/2}^{n+1/2} q_{i,j-1}.$$

Doing so gives the upwind method

$$(4.5) \quad q_{ij}^{n+1} = q_{ij} - \frac{k}{h}(u_{i+1/2,j}^{n+1/2} q_{ij} - u_{i-1/2,j}^{n+1/2} q_{i-1,j} + v_{i,j+1/2}^{n+1/2} q_{ij} - v_{i,j-1/2}^{n+1/2} q_{i,j-1}).$$

This is clearly an approximation to the conservative form (4.3).

Another possibility would be to use the advective form directly, setting

$$(4.6) \quad q_{ij}^{n+1} = q_{ij} - \frac{k}{h} \Delta_{ij},$$

where

$$(4.7) \quad \Delta_{ij} = u_{i-1/2,j}^{n+1/2}(q_{ij} - q_{i-1,j}) + v_{i,j-1/2}^{n+1/2}(q_{ij} - q_{i,j-1}).$$

This form appears preferable in the situation described above, since if $q_{ij} \equiv \text{constant}$ then $\Delta_{ij} \equiv 0$. On the other hand, it is not clear that it is conservative, since it is not written in

flux-differencing form. In fact, the two forms give precisely the same q_{ij}^{n+1} in the case of incompressible flow. We can rewrite Δ_{ij} as

$$\begin{aligned}\Delta_{ij} &= [u_{i+1/2,j}^{n+1/2}q_{ij} - (u_{i+1/2,j}^{n+1/2} - u_{i-1/2,j}^{n+1/2})q_{ij}] - u_{i-1/2,j}^{n+1/2}q_{i-1,j} \\ &\quad + [v_{i,j+1/2}^{n+1/2}q_{ij} - (v_{i,j+1/2}^{n+1/2} - v_{i,j-1/2}^{n+1/2})q_{ij}] - v_{i,j-1/2}^{n+1/2}q_{i,j-1} \\ &= u_{i+1/2,j}^{n+1/2}q_{ij} - u_{i-1/2,j}^{n+1/2}q_{i-1,j} + v_{i,j+1/2}^{n+1/2}q_{ij} - v_{i,j-1/2}^{n+1/2}q_{i,j-1} \\ &\quad - (u_{i+1/2,j}^{n+1/2} - u_{i-1/2,j}^{n+1/2} + v_{i,j+1/2}^{n+1/2} - v_{i,j-1/2}^{n+1/2})q_{ij}.\end{aligned}$$

Using the constraint (4.2), the last term drops out and we see that this gives the same update as the flux-differencing formula (4.5).

The flux updates required to introduce transverse propagation and second-order accuracy are identical to the formulas already presented, with the constants u and v replaced by u and v evaluated at the midpoint of the interface from which the wave propagates. Algorithm 4.1 gives the complete set of formulas, including the logic needed to generalize from $u, v > 0$ to the general case where the sign of u and v at each interface is used to determine the direction of propagation of the waves. These formulas follow almost directly from the formulas (4.6), (3.3), and (3.7). For clarity and ease of comparison of different methods, it is indicated in this algorithm where to break out of the loop in order to implement each of four different methods. Continuing on simply adds additional correction terms. The four methods are listed below.

Method 1. The upwind method (4.6).

Method 2. The upwind method with transverse propagation using (3.3).

Method 3. The second-order method with correction waves propagating normal to the interfaces using (3.7).

Method 4. The improved second-order method with transverse propagation of the correction waves using (3.9).

A Fortran implementation of these algorithms is also available by anonymous ftp (see §12). This gives complete implementation details and also includes test data for the numerical examples presented in §9.

5. Other approaches and related methods. The advection problem has been extensively studied, both because of its own importance and as a model problem for other fluid dynamical equations. The recent survey by Rood [44], which is mainly for one-dimensional algorithms, mentions that over 100 algorithms were found and contains an extensive bibliography. Leonard [21] and Zalesak [63] also give comparisons of many methods applied to one-dimensional advection problems. It is clearly impossible to give a complete survey here, but it may be useful to briefly describe a few other approaches and how they relate to the methods derived here. (See also [6, 22, 50, 61].)

Methods with at least second-order accuracy were initially obtained by using either the approach of Lax and Wendroff, by using centered approximations to derivatives (e.g., [19, 34]) or by applying dimensional splitting to one-dimensional algorithms (e.g., [20, 10]). Crowley [10] developed fourth-order accurate algorithms and also studied the difference between conservative and advective forms of the equations and noted the superiority of the advective form.

The use of centered approximations for the second- and cross-derivative terms leads to oscillations, of course, and the earliest attempts to eliminate these oscillations relied on the addition of “artificial viscosity”. This was suggested already by Lax and Wendroff [18]. Dukowicz and Ramshaw [12] developed the tensor viscosity method as an approach to handling the cross-derivative terms in a more physical manner to reduce these oscillations. Smolarkiewicz

```

# Initialize increments and fluxes:
for each i, j do
     $F_{i-1/2,j} := 0, \quad G_{i,j-1/2} := 0$ 
# Update increments and fluxes based on interfaces in x-direction:
for each i, j do
    # consider interface between cells  $C_{i-1,j}$  and  $C_{ij}$ :
     $U := u_{i-1/2,j}^{n+1/2}$ 
     $V := v_{i-1/2,j}^{n+1/2}$ 
     $R := q_{ij} - q_{i-1,j}$ 
    if  $U > 0$  then  $I := i - 1$  else  $I := i$ 
     $F_{i-1/2,j} := F_{i-1/2,j} + Uq_{I,j}$ 
    # if method = 1 then end loop here
    if  $U > 0$  then  $I := i$  else  $I := i - 1$ 
    if  $V > 0$  then  $J := j + 1$  else  $J := j$ 
     $G_{I,J-1/2} := G_{I,J-1/2} - \frac{1}{2} \frac{k}{h} UV R$ 
    # if method = 2 then end loop here
     $R :=$  limited version of  $R$ 
     $S := \frac{1}{2}|U| \left( 1 - \frac{k}{h}|U| \right) R$ 
     $F_{i-1/2,j} := F_{i-1/2,j} + S$ 
    # if method = 3 then end loop here
     $G_{i,J-1/2} := G_{i,J-1/2} + \frac{k}{h} V S$ 
     $G_{i-1,J-1/2} := G_{i-1,J-1/2} - \frac{k}{h} V S$ 
    # if method = 4 then end loop here

# Update increments and fluxes based on interfaces in y-direction:
# Similar to the above but with the roles of i and j,
# u and v, and F and G switched

# Update q:
for each i, j do
     $q_{ij}^{n+1} = q_{ij} - \frac{k}{h} [F_{i+1/2,j} - F_{i-1/2,j} + G_{i,j+1/2} - G_{i,j-1/2}]$ 

```

ALGORITHM 4.1. One time step of the algorithm for general two-dimensional flow.

[52] developed MPDATA (multidimensional positive-definite advective transport algorithm) based on iteratively applying the donor-cell upwind method to modified equations determined by the truncation error of the previous iteration. Smolarkiewicz and Clark [53] showed how to include the cross-derivative terms in the multidimensional version of this approach and later work [49, 54] has directly incorporated limiters as well. Harten [16] took a similar approach to developing TVD schemes, applying a monotone method to a modified conservation law to achieve second-order accuracy.

The idea of limiters arose in many contexts. Boris and Book [5] developed the flux corrected transport (FCT) method based on correcting a first-order flux by the addition of a limited portion of a second-order correction. Harten and Zwas [17] considered similar hybrid methods and van Leer [59, 60] developed slope-limiter MUSCL methods (monotonic upstream-centered scheme for conservation laws) based on piecewise linear functions. This was later extended to piecewise quadratics in the piecewise parabolic method (PPM) of Colella and Woodward [9]. These methods were developed for more general nonlinear conservation laws in one space dimension and were typically extended to more dimensions by dimensional splitting.

Colella's multidimensional method for conservation laws [8] involves upwinding and limiting of cross-derivative terms in a manner similar to that proposed here. The basic idea in his approach is to extrapolate from the cell averages, viewed as pointwise values at the center of each cell at time t_n , to cell interface values at time $t_{n+1/2}$, using Taylor series expansions. Replacing the time derivative by spatial derivatives brings in the transverse derivative term and limiters are introduced in approximating the spatial derivatives. Different interface values are obtained from each side and these two values are then resolved by solving a Riemann problem, which for the advection equation simply amounts to using the upwind value. For the constant coefficient problem, the first-order CTU algorithm of [8] agrees with Method 2 here while the second-order algorithm agrees with Method 3, though the implementations are viewed quite differently. For variable coefficients the methods are distinct.

A three-dimensional extension is given by Saltzman [46] (see also [58]). The advection version of this algorithm has also been employed by Bell, Colella, and Glaz [1] in their second-order projection method for the incompressible Navier–Stokes equations (see also [2, 4]) and by Pilliod and Puckett [37] for advection algorithms in the context of volume-of-fluid interface tracking. Bell, Dawson, and Shubin [3] developed an algorithm for porous media flow based on Colella's approach and also employ multidimensional limiters as discussed below.

The development of the wave-propagation framework used here was based on work by Roe (e.g., [40, 39, 41]), who introduced the idea of defining fluxes by shifting "fluctuations". This has more recently been used to develop multidimensional fluctuation-splitting methods for hyperbolic systems and associated upwind advection algorithms (e.g., [42]). Radvugin [38] has introduced a multidimensional algorithm for both advection and hyperbolic systems that also uses similar ideas.

The shift-and-average form of the algorithm from §3 can be interpreted as reconstructing a function from given grid data, moving this function forward in time by the advection equation, and then averaging to obtain new grid values. Alternatively, one could define new grid values by starting at a grid point and tracing backwards in time via the advection algorithm to find the departure point of the particle now at this grid point and then interpolating between grid values at the previous time level to obtain the value of q at the departure point.

In the case where (3.6) is satisfied and bilinear interpolation is used between the four grid points surrounding the departure point exactly parallels Method 2 (the CTU method). A more general advantage of this approach is that time steps much larger than those allowed by (3.6) can often be used. Numerous advection algorithms have been developed along these lines, such as semi-Lagrangian methods (see the recent review [55]), the modified method of characteristics (e.g., [11]), and the characteristic Galerkin method (e.g., [31, 32, 35, 36]). These methods are also quite successful for advection–diffusion problems where the advective terms dominate. Note that q_{ij} is now viewed as the value of q at a grid point rather than a cell average, and conservation may be harder to maintain. Scroggs and Semazzi [48] have recently addressed this problem.

Leonard, MacVean, and Lock [23] also use this interpolation viewpoint to develop second- and third-order multidimensional methods for constant velocity problems, which in the absence of limiters agree with Method 3 from §3 and Method 5 (introduced in §7), resp. (See also [21, 22].) They also develop multidimensional limiters that take into account the total contribution to each cell value from fluxes at all sides of the cell. This is a more sophisticated limiter than that used in the present algorithm (which limits each flux based only on information about the solution in that coordinate direction).

Multidimensional limiting has the advantage that it is possible to eliminate oscillations and undershoots entirely; this is particularly important in applications where the method must be positivity preserving, for example. Other forms of multidimensional limiting have also been developed, starting with the fundamental work of Zalesak [62] who extended the FCT method to a multidimensional version that does not rely on dimensional splitting. The version of MPDATA developed by Smolarkiewicz and Grabowski [49] (based on the FCT limiter) also has this property. Saltzman [45] has proposed a similar multidimensional limiter based on van Leer's MUSCL scheme. Roe and Sidilkover [43] also discuss multidimensional upwinding from the standpoint of maintaining positivity.

The second-order algorithms developed in this paper use one-dimensional flux-limiters applied to the second-order correction term. This is done for simplicity and leads to algorithms that are quite easy to implement and efficient to run. They may exhibit slight oscillations and lack of positivity, but these effects (with Method 4) are typically very slight and for many applications the relative ease and sharp resolution may be more important. If desired, the methodology developed here could presumably be combined with some of the multidimensional limiting techniques mentioned above. This could be particularly valuable for the third-order method developed in §7, for which good one-dimensional limiters have not yet been developed.

In spite of the plethora of advection algorithms already in the literature, the method proposed here seems to have some unique features. It uses a multidimensional approach, rather than splitting, in a manner that is quite simple to implement. Dimensional splitting is still widely used in practice because of its simplicity, but in many applications it would be desirable to use a multidimensional approach. The simple form of flux limiting was chosen to keep the implementation quite simple and should be sufficient in many applications. The method is conservative and yet has the advantages of advection-form algorithms. The geometric interpretation of wave propagation lends itself to generalizing these ideas to more interesting situations, e.g., algorithms on curvilinear or even unstructured grids, or to more complicated equations with advective behavior. The basic ideas generalize quite naturally to nonlinear systems of conservation laws and were actually developed there first [26, 29], whereas many specialized advection algorithms do not generalize well.

The next three sections contain truncation error analysis, generalization to third-order accuracy, and stability analysis. The casual reader may wish to advance to §9 at this point.

6. Truncation error analysis. The truncation error of the method presented in Algorithm 4.1 is easy to compute in principle but complicated by the algorithmic form. For concreteness we will assume that u and v are positive in the neighborhood of the point (i, j) under consideration, i.e., at all nearby interfaces. The analysis is similar if u or v is negative, but the formulas then involve different grid points because of the upwind nature of the algorithm. Even with this assumption, which fixes the stencil, it is still not easy to write out the difference scheme in the classical form. Algorithm 4.1 looks deceptively simple but the method in finite difference form is quite complicated. Rather than writing this out in full and then expanding each grid value in Taylor series to compute the local truncation error, we will

consider each step of the algorithm and compute its contribution to the local error. This not only helps to organize the computation but also leads to a clear understanding of the effect of each step on the error. We will consider the full method (Method 4) in the absence of limiters, although our analysis will show that second-order accuracy is also achieved by using Method 3. We begin by assuming that u and v are time independent, although they may vary spatially. We will then see that second-order accuracy is maintained in the time-dependent case by evaluating the velocities at the half time step.

We will compute in detail only the contributions to the error that come from differencing the F fluxes and from the portion of Δ_{ij} that corresponds to wave propagation in the x -direction. The contributions from differencing the G fluxes and the upwind method in the y -direction will then follow easily by symmetry. We will determine how q_{ij} is updated in one time step in computing q_{ij}^{n+1} and compare this with the correct update given by

$$(6.1) \quad q_{ij}^{n+1} = q_{ij}^n + kq_t + \frac{1}{2}k^2q_{tt} + \frac{1}{6}k^3q_{ttt} + \dots$$

To demonstrate second-order accuracy we need to show that the q_t and q_{tt} terms are matched correctly. We will also keep track of the $O(k^3)$ terms since we will see that for the special case of constant coefficients a simple modification of the method gives third-order accuracy. In this case the modifications used in Method 4 are essential since they provide the cross-derivative terms appearing in q_{ttt} . This is discussed in the next section.

Note that

$$(6.2) \quad q_t = -(uq_x + vq_y)$$

and

$$(6.3) \quad \begin{aligned} q_{tt} &= -(uq_{tx} + vq_{ty}) \\ &= u(uq_x + vq_y)_x + v(uq_x + vq_y)_y \\ &= (u^2q_x + uvq_y)_x + (vuq_x + v^2q_y)_y, \end{aligned}$$

where we have used the fact that $u_x + v_y = 0$.

The basic first-order flux in the x -direction (see (4.6)) modifies q_{ij} by

$$-\frac{k}{h}u_{i-1/2,j}(q_{ij} - q_{i-1,j}).$$

Expanding this in a Taylor series about (x_i, y_j) shows that this is equal to

$$(6.4) \quad -kuq_x + \frac{1}{2}kh(uq_x)_x - \frac{1}{24}kh^2(3(uq_x)_{xx} + uq_{xxx}) + O(k^4),$$

where all functions are evaluated at (x_i, y_j) and we assume k/h is fixed so that $h = O(k)$.

The modification (3.4) corresponding to transverse propagation of the increment waves updates q_{ij} by (after computing $-\frac{k}{h}(F_{i+1/2,j} - F_{i-1/2,j})$)

$$(6.5) \quad \begin{aligned} &\frac{k}{h} \left[\frac{1}{2} \frac{k}{h} u_{i,j-1/2} v_{i,j-1/2} (q_{ij} - q_{i,j-1}) \right. \\ &\quad \left. - \frac{1}{2} \frac{k}{h} u_{i-1,j-1/2} v_{i-1,j-1/2} (q_{i-1,j} - q_{i-1,j-1}) \right] \\ &= \frac{1}{2}k^2(uvq_y)_x - \frac{1}{4}k^2h[(uvq_y)_{xx} + (uvq_y)_{xy}] + O(k^4). \end{aligned}$$

The slope modification (3.7) updates q_{ij} by

$$\begin{aligned}
 & -\frac{k}{h} \left[\frac{1}{2} u_{i+1/2,j} \left(1 - \frac{k}{h} u_{i+1/2,j} \right) (q_{i+1,j} - q_{ij}) \right. \\
 & \quad \left. - \frac{1}{2} u_{i-1/2,j} \left(1 - \frac{k}{h} u_{i-1/2,j} \right) (q_{ij} - q_{i-1,j}) \right] \\
 (6.6) \quad & = -\frac{1}{2} kh (uq_x)_x + \frac{1}{2} k^2 (u^2 q_x)_x + O(k^4).
 \end{aligned}$$

Finally, propagating the slope modifications from (3.8) in the transverse direction gives modifications to the F fluxes analogous to (3.9). There are four such modifications that combine to affect q_{ij} , giving a total contribution to q_{ij} of

$$\begin{aligned}
 & -\frac{k}{h} \left[-\frac{1}{2} \frac{k}{h} v_{i,j+1/2} u_{i,j+1/2} \left(1 - \frac{k}{h} v_{i,j+1/2} \right) (q_{i,j+1} - q_{ij}) \right. \\
 & \quad + \frac{1}{2} \frac{k}{h} v_{i,j-1/2} u_{i,j-1/2} \left(1 - \frac{k}{h} v_{i,j-1/2} \right) (q_{ij} - q_{i,j-1}) \\
 (6.7) \quad & \quad + \frac{1}{2} \frac{k}{h} v_{i-1,j+1/2} u_{i-1,j+1/2} \left(1 - \frac{k}{h} v_{i-1,j+1/2} \right) (q_{i-1,j+1} - q_{i-1,j}) \\
 & \quad \left. - \frac{1}{2} \frac{k}{h} v_{i-1,j-1/2} u_{i-1,j-1/2} \left(1 - \frac{k}{h} v_{i-1,j-1/2} \right) (q_{i-1,j} - q_{i-1,j-1}) \right] \\
 & = \frac{1}{2} k^2 h (uvq_y)_{yx} - \frac{1}{2} k^3 (uv^2 q_y)_{yx} + O(k^4).
 \end{aligned}$$

Note that this final update is $O(k^3)$ and does not affect the second-order accuracy of the method, as claimed in §3.

Adding up the updates (6.4), (6.5), and (6.6) gives

$$-kuq_x + \frac{1}{2} k^2 ((u^2 q_x)_x + (uvq_y)_x) + O(k^3)$$

and adding in the corresponding terms from the G -differencing and the y -component of the upwind method gives the total update of q_{ij} :

$$\begin{aligned}
 (6.8) \quad q_{ij}^{n+1} & = q_{ij} - k(uq_x + vq_y) \\
 & \quad + \frac{1}{2} k^2 ((u^2 q_x)_x + (uvq_y)_x + (vuq_x)_y + (v^2 q_y)_y) + O(k^3).
 \end{aligned}$$

Comparing this with (6.1) using (6.2) and (6.3) shows that the method is second-order accurate.

Now suppose that the velocities are time dependent and we use the algorithm developed in §4 with u and v evaluated at time $t_{n+1/2}$. Then we still obtain (6.8) but with u and v evaluated at the half time step and so using a Taylor series in time to expand u and v in (6.8) about (x_i, y_j, t_n) gives

$$\begin{aligned}
 (6.9) \quad q_{ij}^{n+1} & = q_{ij} - k((u + ku_t/2)q_x + (v + kv_t/2)q_y) \\
 & \quad + \frac{1}{2} k^2 ((u^2 q_x)_x + (uvq_y)_x + (vuq_x)_y + (v^2 q_y)_y) + O(k^3) \\
 & = q_{ij} - k(uq_x + vq_y) + \frac{1}{2} k^2 ((u^2 q_x)_x + (uvq_y)_x \\
 & \quad + (vuq_x)_y + (v^2 q_y)_y - (u_t q_x + v_t q_y)) + O(k^3).
 \end{aligned}$$

This shows that we again have second-order accuracy since in the time-dependent case q_t is still given by (6.2) while (6.3) is replaced by

$$\begin{aligned} q_{tt} &= -(uq_{tx} + vq_{ty}) - (u_tq_x + v_tq_y) \\ &= (u^2q_x + uvq_y)_x + (vuq_x + v^2q_y)_y - (u_tq_x + v_tq_y). \end{aligned}$$

7. Third-order accuracy for constant coefficients. For the general variable coefficient problem it seems difficult to determine what additional correction terms must be added to obtain third-order accuracy. For the special case of constant coefficients, however, this term is quite simple. Combining the $O(k^3)$ terms from (6.4)-(6.7) and the corresponding terms from G -differencing gives

$$\begin{aligned} &-\frac{1}{6}kh^2uq_{xxx} - \frac{1}{4}k^2h(uvq_{xxy} + uvq_{xyy}) + \frac{1}{2}k^2huvq_{xyy} \\ &\quad - \frac{1}{2}k^3uv^2q_{yyy} + \text{similar terms from } G\text{-differencing} \\ (7.1) \quad &= -\frac{1}{6}kh^2(uq_{xxx} + vq_{yyy}) - \frac{1}{2}k^3(uv^2q_{xyy} + u^2vq_{xxy}). \end{aligned}$$

To achieve third-order accuracy we need to match the $O(k^3)$ term in (6.1), which is

$$(7.2) \quad \frac{1}{6}k^3q_{ttt} = -\frac{1}{6}k^3(u^3q_{xxx} + 3u^2vq_{xxy} + 3uv^2q_{xyy} + v^3q_{yyy}).$$

Comparing (7.1) with (7.2) shows that we need to make an additional update to q_{ij} of the form

$$\frac{1}{6}kh^2\left(1 - \frac{k^2}{h^2}u^2\right)uq_{xxx} + \frac{1}{6}kh^2\left(1 - \frac{k^2}{h^2}v^2\right)vq_{yyy}.$$

This can be easily accomplished by modifying the F flux by a term modeling

$$-\frac{1}{k}\left(\frac{1}{6}kh^2\left(1 - \frac{k^2}{h^2}u^2\right)uq_{xx}\right)$$

and making a similar modification to the G flux. For example, we could use

$$(7.3) \quad F_{i-1/2,j} := F_{i-1/2,j} - \frac{1}{6}u\left(1 - \frac{k^2}{h^2}u^2\right)(q_{i+1,j} - 2q_{ij} + q_{i-1,j})$$

in the x -sweeps, and

$$(7.4) \quad G_{i,j-1/2} := G_{i,j-1/2} - \frac{1}{6}v\left(1 - \frac{k^2}{h^2}v^2\right)(q_{i,j+1} - 2q_{ij} + q_{i,j-1})$$

in the y -sweeps, where

$$I = \begin{cases} i, & \text{if } u < 0, \\ i - 1, & \text{if } u \geq 0, \end{cases} \quad \text{and} \quad J = \begin{cases} j, & \text{if } v < 0, \\ j - 1, & \text{if } v \geq 0, \end{cases}$$

are chosen so that the third-order corrections are properly upwinded. Note that these correction terms involve only pure x - and y -derivatives. All of the cross-derivative terms have already been correctly modeled by the transverse propagation of the lower-order terms.

In developing the second-order accurate method we found that it was advantageous to also propagate the correction waves in the transverse direction even though this is not strictly necessary for second-order accuracy. Also here it turns out to be advantageous to propagate these third-order corrections in the transverse direction by using the additional updates (again assuming $u, v > 0$)

$$(7.5) \quad G_{i-1,j+1/2} := G_{i-1,j+1/2} - \frac{kv}{h} \left\{ \frac{1}{6}u \left(1 - \frac{k^2}{h^2}u^2 \right) ((q_{I+1,j} - 2q_{Ij} + q_{I-1,j})) \right\},$$

$$G_{i,j+1/2} := G_{i,j+1/2} + \frac{kv}{h} \left\{ \frac{1}{6}u \left(1 - \frac{k^2}{h^2}u^2 \right) (q_{i,J+1} - 2q_{iJ} + q_{i,J-1}) \right\},$$

following (7.3) and similar updates to the F fluxes following (7.4). We will refer to the algorithm in which only the modifications (7.3) and (7.4) are made as Method 5 and the improved version with transverse propagation (7.5) (and similar modifications to F) as Method 6. The latter method appears to have a smaller error constant when solving the constant coefficient problem. In this case including the transverse derivatives also enlarges the stability region. Method 6 is stable provided (3.6) is satisfied whereas Method 5 has a stability region closer to that specified by (3.5). These stability regions are shown in Figure 8.1.

Note that Algorithm 4.1 requires only a trivial modification to convert it from Method 4 to Method 6. Simply replace the line

$$S := \frac{1}{2}|U| \left(1 - \frac{k}{h}|U| \right) R$$

by

$$S := \frac{1}{2}|U| \left(1 - \frac{k}{h}|U| \right) R - \frac{1}{6}U \left(1 - \frac{k^2}{h^2}U^2 \right) (q_{I+1,j} - 2q_{Ij} + q_{I-1,j}).$$

This modification to the algorithm can be used even when \vec{u} is not constant. In this case the method is no longer third-order accurate, but numerical tests presented below indicate that including these terms can improve the error constant substantially.

These methods give quite good accuracy on problems where q is smooth. With steep gradients or discontinuities one would expect that some form of limiter will be needed for the third-order correction terms analogous to what is used for the second-order correction terms. Preliminary attempts to introduce such limiters have not been completely successful and more work is needed in this direction. One possibility is to use multidimensional limiters as in [23, 45, 62].

It is interesting to note, however, that surprisingly good results are obtained by using Method 6 with no limiter on any of the correction terms (see Example 9.4). Presumably this is due to the fact that the dominant term in the third-order method is a dissipative term rather than a dispersive term.

Method 6 may be particularly useful for solving advection-diffusion equations where there is some physical diffusion to insure that the solution is smooth.

8. Stability. In this section the stability region of each method is displayed for the case of constant coefficients, periodic boundary conditions, and no limiters, so that von Neumann analysis can be performed (see, e.g., Strikwerda [56] for a general discussion of this approach). With u and v constant we define the Courant numbers μ and ν by

$$\mu = uk/h \quad \text{and} \quad \nu = vk/h.$$

Von Neumann analysis is performed by considering modes of the form

$$q_{IJ} = e^{i(\xi I + \eta J)},$$

where $i = \sqrt{-1}$ here and the frequencies ξ and η range between $-\pi$ and π . Applying any of the above methods then results in an expression of the form

$$q_{IJ}^{n+1} = g(\xi, \eta; \mu, \nu) q_{IJ}^n,$$

where the *amplification factor* g depends only on the frequencies ξ and η and the Courant numbers μ and ν . We also define

$$g_{\max}(\mu, \nu) = \max_{-\pi \leq \xi, \eta \leq \pi} |g(\xi, \eta; \mu, \nu)|.$$

The method is stable for a given set of Courant numbers (μ, ν) if $g_{\max}(\mu, \nu) \leq 1$. Note that with all of the methods $g(0, 0; \mu, \nu) = 1$, so that $g_{\max}(\mu, \nu) \geq 1$ everywhere and so the stability region is the set of points in the (μ, ν) plane where $g_{\max}(\mu, \nu) = 1$. We will assume $\mu, \nu \geq 0$, and determine the portion of the stability region in the first quadrant. The rest of the region follows by reflection across the axes.

For the first-order methods it is possible to explicitly calculate the stability region. With Method 1 we have

$$g(\xi, \eta; \mu, \nu) = 1 - \mu(1 - e^{-i\xi}) - \nu(1 - e^{-i\eta})$$

and $|g|$ is bounded by 1 for all ξ and η if and only if $\mu + \nu \leq 1$.

For Method 2 the introduction of transverse derivatives gives an additional term, and

$$\begin{aligned} g(\xi, \eta; \mu, \nu) &= 1 - \mu(1 - e^{-i\xi}) - \nu(1 - e^{-i\eta}) + \mu\nu(1 - e^{-i\xi})(1 - e^{-i\eta}) \\ &= (1 - \mu(1 - e^{-i\xi}))(1 - \nu(1 - e^{-i\eta})). \end{aligned}$$

Now g is the product of two factors, each of which is bounded by 1 provided $\mu \leq 1$ and $\nu \leq 1$, so the stability bound is $\max(\mu, \nu) \leq 1$.

For the second- and third-order methods, formulas for g are easy to derive but it is not so easy to determine the stability regions analytically.¹ Instead the region has been determined numerically by computing $g_{\max}(\mu, \nu)$ on a 60×60 grid of values in the first quadrant of the (μ, ν) plane and plotting contour lines. Each value of $g_{\max}(\mu, \nu)$ is determined by calculating $g(\xi, \eta; \mu, \nu)$ on a 60×60 grid of (ξ, η) values and finding the maximum.

The results are shown in Figure 8.1, along with the stability region for the Lax-Wendroff method for comparison. Contour lines are shown at $g_{\max}(\mu, \nu) = 1.001, 1.1, 1.2, 1.3, \dots$. The area between the origin and the first contour line is the stability region. These results show that Methods 3, 4, and 6 are stable for all Courant numbers up to 1, i.e., for $\max(\mu, \nu) \leq 1$. Method 5 appears to be stable in roughly the same region as Method 1, though the instability is milder over the remainder of the unit square. In fact the amplification factor increases and then decreases back to the value 1 at the corner (1,1), which is an isolated stable point due to the fact that the algorithm happens to be exact in this special case. This was also noted by Leonard, MacVean, and Lock [23], who propose Method 5 as a third-order accurate method for advection with constant flow.

¹Note added in proof. Richard Liska [33] has recently proved the stability of Methods 3 and 4 for $\max(\mu, \nu) \leq 1$ using the computer algebra system REDUCE and the package QEPCAD built on the symbolic manipulation library SACLIB.

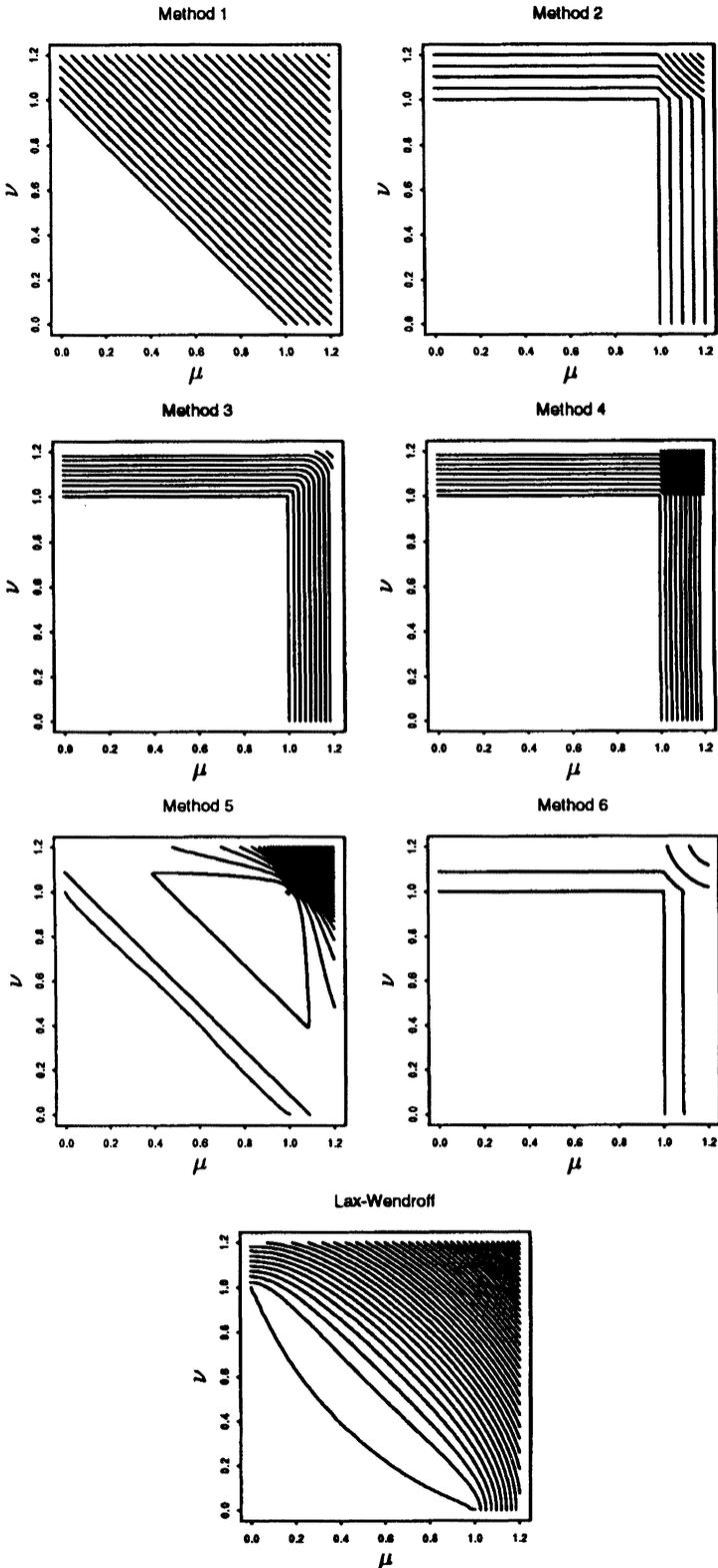


FIG. 8.1. Stability regions for each method in the (μ, ν) plane. Contours of the amplification factor are shown and the blank region between the origin and the first contour line is the stability region. Contours are at $|g| = 1.001, 1.1, 1.2, \dots$

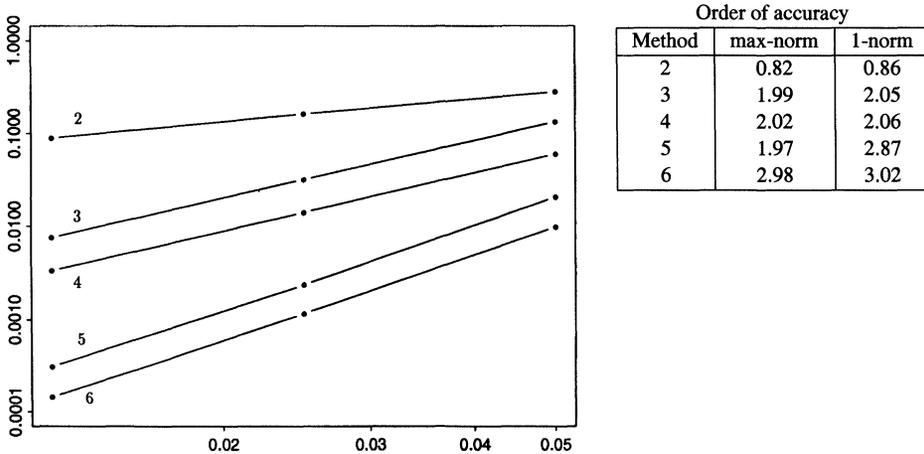


FIG. 9.1. Log-log plot of the 1-norm error vs. h for Methods 2-6 on the constant coefficient problem of Example 9.1, with $u = 1$ and $v = 2$.

9. Numerical results. Numerous tests have been performed to investigate the behavior of the methods developed in the previous sections. Some of these results are presented here. All of the flow fields used in these computations have the property that simply evaluating the velocity at the cell interfaces gives discrete values satisfying the discrete divergence-free condition (4.2).

Example 9.1. We first compare the errors and observed order of accuracy for a variety of methods and limiters on test problems with smooth initial data. Figure 9.1 shows the errors obtained on a problem with constant coefficients

$$(9.1) \quad u(x, y, t) \equiv 1, \quad v(x, y, t) \equiv 2$$

and the initial data

$$(9.2) \quad q(x, y, 0) = \sin(2\pi x) \sin(2\pi y).$$

In this case periodic boundary conditions are used and we compute up to time $t = 1$ at which point the initial data should be recovered. The time step $k = 0.4h$ (Courant number 0.8) is used in all the computations. (Note that Method 1, the donor-cell upwind method, is unstable with this time step.) The figure shows a log-log plot of error vs. h for a sequence of grids with $h = 0.05, 0.025, 0.0125$. Methods 5 and 6 refer to the methods introduced in §7. These methods should be third-order accurate on constant coefficient problems.

The table in Figure 9.1 shows the observed order of accuracy of each method in both the 1-norm and the max-norm, as computed by comparing the errors on the finest two grids,

$$\text{order} = \log_2(E(h)/E(h/2)),$$

where $E(h)$ is the norm of the error with grid spacing h , relative to the true solution. We see the expected rate of convergence in all cases. Moreover we see that the correction terms (3.9) and (3.10) corresponding to transverse propagation of the slopes improve the accuracy (going from Method 3 to Method 4) even though the order of accuracy is unchanged. The same effect is seen in going from Method 5 to Method 6, which introduces transverse propagation of the third-order correction terms.

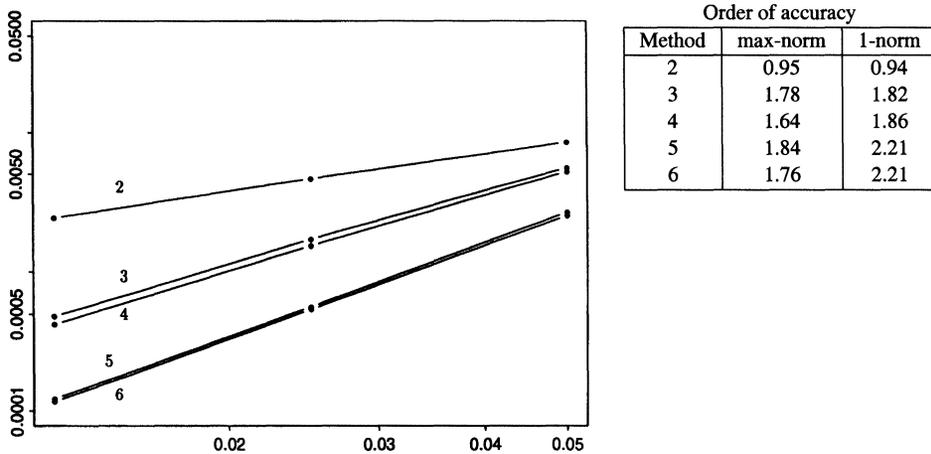


FIG. 9.2. Log-log plot of the 1-norm error vs. h for Methods 2-6 on the rotating flow problem of Example 9.2.

Example 9.2. Next we consider a problem where the flow is not constant. We consider solid body rotation with

$$(9.3) \quad u = -(y - 1/2), \quad v = (x - 1/2).$$

As initial data we take a smooth hump of the form

$$(9.4) \quad q(x, y, 0) = \frac{1}{4}(1 + \cos(\pi r(x, y))),$$

where

$$r(x, y) = \min\left(\sqrt{(x - x_0)^2 + (y - y_0)^2}, r_0\right) / r_0.$$

This computation was done on the domain $[0, 1] \times [0.5, 1.5]$ with $x_0 = 0.5$, $y_0 = 1.25$, and $r_0 = 0.2$. The strange location of the domain was motivated by later tests of boundary conditions using the same rotating hump (presented in the next section, where Figure 10.2(a) shows the hump rotating up to time $t = \pi$). Here we compute only up to time $t = 0.25$ so the hump stays within the computational domain.

Figure 9.2 shows a log-log plot of the errors for this case and a table of the computed order of accuracy from these results. In this case we see that including transverse propagation gives only a slight improvement (going from Method 3 to Method 4 and from Method 5 to Method 6). Moreover, Methods 5 and 6 are now only second-order accurate rather than third-order accurate, as expected, although the error constant is smaller than with Methods 3 and 4.

In the above computations no limiter was used. Figure 9.3 shows a plot of the errors in the rotating flow case with various limiters. On this particular initial data the use of a limiter generally improves the accuracy, although the order of accuracy is diminished slightly with minmod and substantially with the superbee limiter. The van Leer limiter gives results on this problem that are slightly worse than MC, but on other examples appears to do better.

Example 9.3. Solid body rotation is frequently used as a test case for advection algorithms. Zalesak [62] uses this with a density function that has the shape of a slotted disk. Smolarkiewicz [49], [51] and others have used a rotating cone. To allow comparison of the method proposed here with other results in the literature we use the initial data shown in Figure 9.4, which includes both a slotted disk and a cone as well as a smooth hump of the form (9.4) with

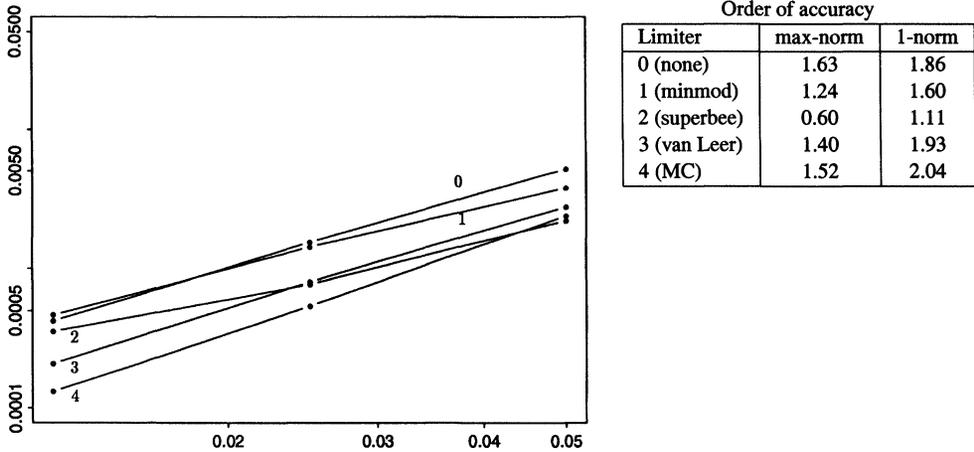


FIG. 9.3. Log-log plot of the 1-norm error vs. h for Method 4 with various limiters on the rotating flow problem of Example 9.2.

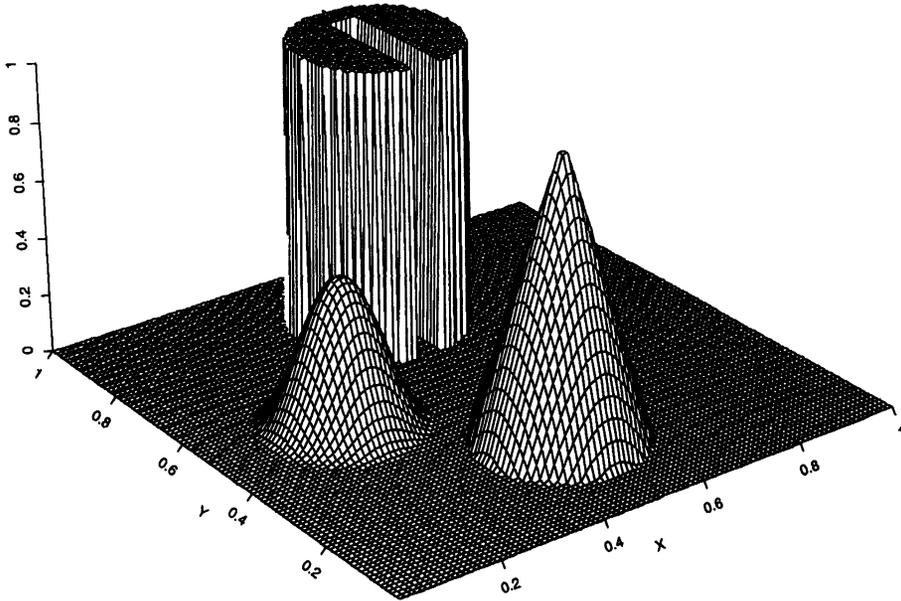
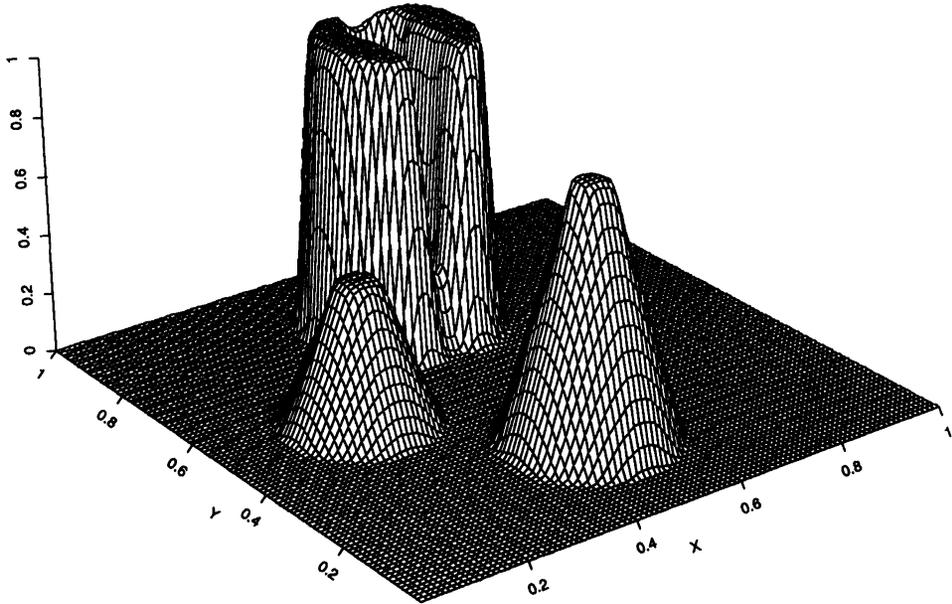


FIG. 9.4. Initial data for solid body rotation tests.

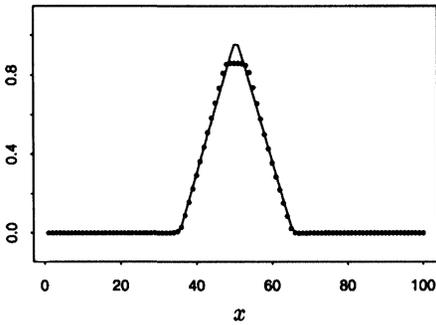
$x_0 = 0.25$, $y_0 = 0.5$, and $r_0 = 0.15$. The cone and disk also have radius 0.15 and are centered at $(0.5, 0.25)$ and $(0.5, 0.75)$, resp.

Figure 9.5 shows the computed results after one revolution (628 time steps) on a 100×100 grid with $k = 0.01$ and hence a Courant number of 1. Method 4 is used along with the superbee limiter. Figure 9.6 shows results after six full revolutions (3768 time steps).

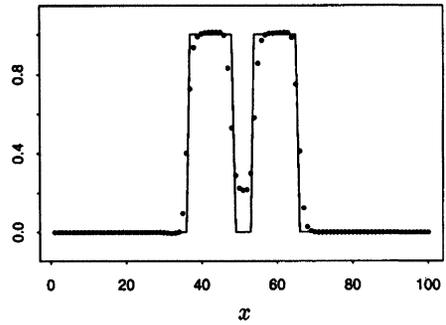
Several other limiters have been tested on this problem. Superbee gives the sharpest results since it is overcompressive, although this has the disadvantage that smooth gradients also tend to be sharpened, as is clearly seen in the evolution of the smooth hump in this test problem. It is also interesting to note that the maximum value of q increases above 1 in the



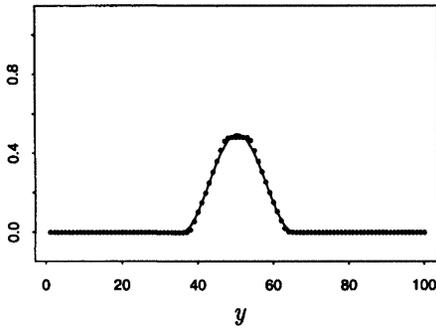
q at $y = 0.25$



q at $y = 0.75$



q at $x = 0.25$



q at $x = 0.5$

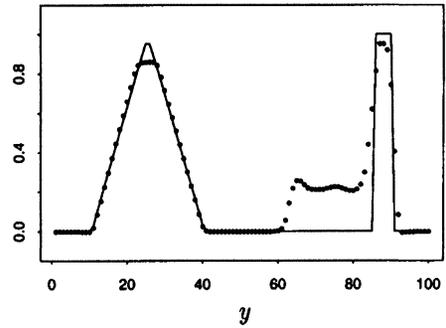
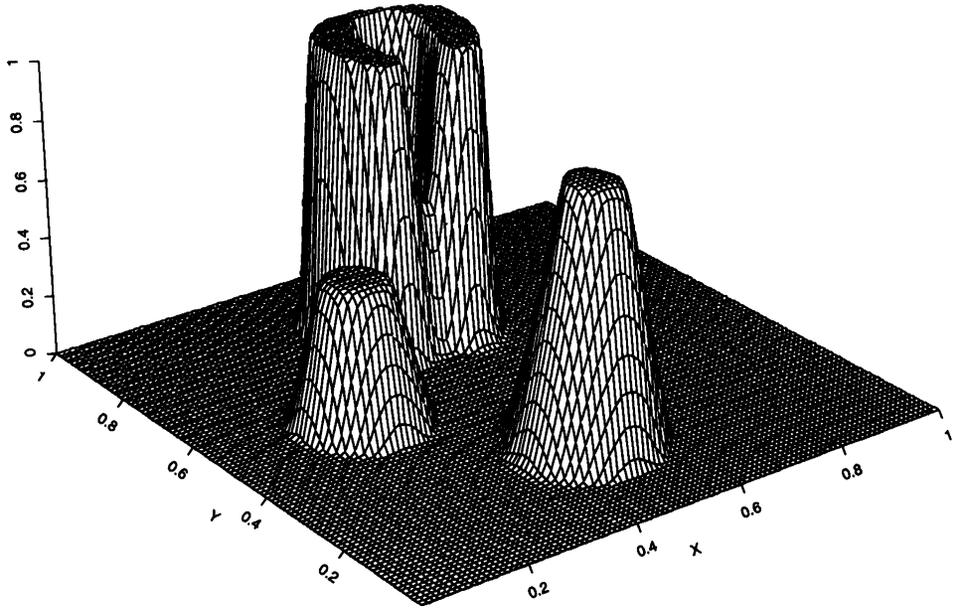
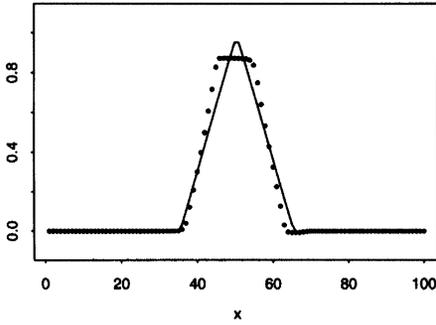


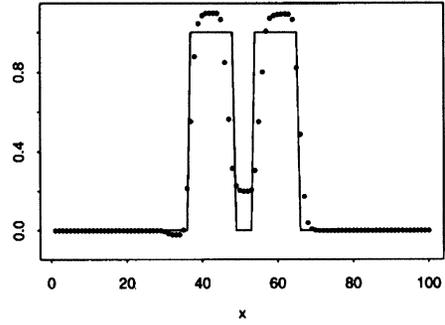
FIG. 9.5. Numerical results for Example 9.3 after one revolution (628 time steps) using Method 4 and the superbee limiter. Four different cross sections of the solution are shown along with the perspective plot. The solid lines are the true solution.



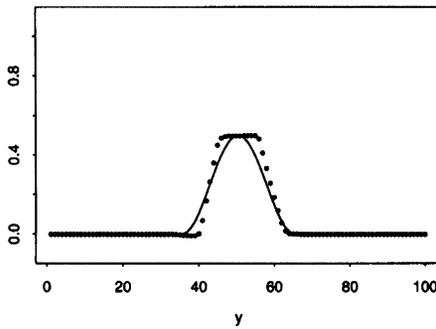
q at $y = 0.25$



q at $y = 0.75$



q at $x = 0.25$



q at $x = 0.5$

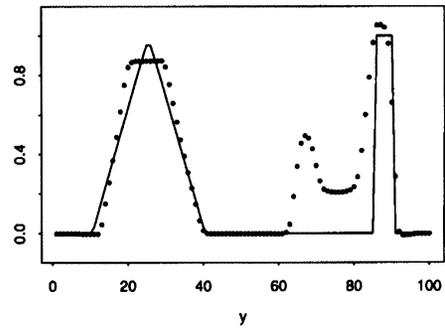


FIG. 9.6. Numerical results for Example 9.3 after six revolutions (3768 time steps) using Method 4 and the superbee limiter. Four different cross sections of the solution are shown along with the perspective plot. The solid lines are the true solution.

slotted disk. This does not happen with most other limiters, which instead allow increased smearing.

There are essentially no oscillations visible in these computations. Conservation has also been verified. In this computation, $\sum q_{ij}^0 = 929.0382$ initially and after six revolutions $\sum q_{ij}^{3768} = 929.0452$. The method fails to be exactly conservative only because of flow through the boundaries with this rotating flow.

Example 9.4. Figure 9.7 shows results when Method 6 is used with no limiter and $k = 0.01$. Since the velocity field is not constant, the method is not third-order accurate in this case. In spite of the fact that no limiter is used, the oscillations produced are quite mild and occur mainly near the discontinuity in the slotted disk. The hump and cone are captured very well.

Example 9.5. A more severe test is obtained by using a swirling deformation flow of the form

$$(9.5) \quad u = \sin^2(\pi x) \sin(2\pi y)g(t), \quad v = -\sin^2(\pi y) \sin(2\pi x)g(t).$$

This flow satisfies $u = v = 0$ on the boundaries of the unit square. The function $g(t)$ is used to introduce time dependence in the flow field and here we use

$$(9.6) \quad g(t) = \cos(\pi t/T)$$

on the time interval $0 \leq t \leq T$. The flow slows down and reverses direction in such a way that the initial data should be recovered at time T : $q(x, y, T) = q(x, y, 0)$. This gives a very useful test problem since we then know the true solution at time T even though the flow field has a quite complicated structure. (This same trick can be used with any other incompressible flow field.)

Here we use $T = 1.5$. At time $T/2$ the initial data is quite deformed. Figure 9.8 shows a contour plot of the computed solution at this time using the same initial data as in Example 9.3. Figure 9.9 shows the results at time T , again using Method 4 and the superbee limiter. The initial shapes have been recovered fairly successfully.

Example 9.6. As a final example we use the swirling flow (9.5) with $g(t) \equiv 1$ and initial data

$$q(x, y, 0) = \begin{cases} 1, & \text{if } (x - 1)^2 + (y - 1)^2 < 0.8, \\ 0, & \text{otherwise.} \end{cases}$$

The computed results obtained at time 2.5 with Method 4 are shown in Figure 9.10. Figure 9.11(a) shows a slice through these results at $x = 0.5$. The results are sharp and the maxima and minima are well preserved. Over the entire domain the values of q_{ij} lie between -0.0045 and 1.0050 .

To demonstrate that the transverse propagation of correction waves (3.9) is valuable even though they are not required to achieve second-order accuracy, Figure 9.11(b) shows results for the same computations but without these terms, i.e., with Method 3. Mild oscillations are clearly visible.

10. Boundary conditions. Boundary conditions can be incorporated quite easily into the methods described above. The computational grid is extended by one or two rows of cells along each edge and values of q are assigned to each of these cells at the beginning of each time step in a way that depends on the nature of the boundary conditions (as described below). The method is then applied over the interior of the expanded grid so that fluxes at interfaces corresponding to the original boundary are determined. For concreteness in the discussion

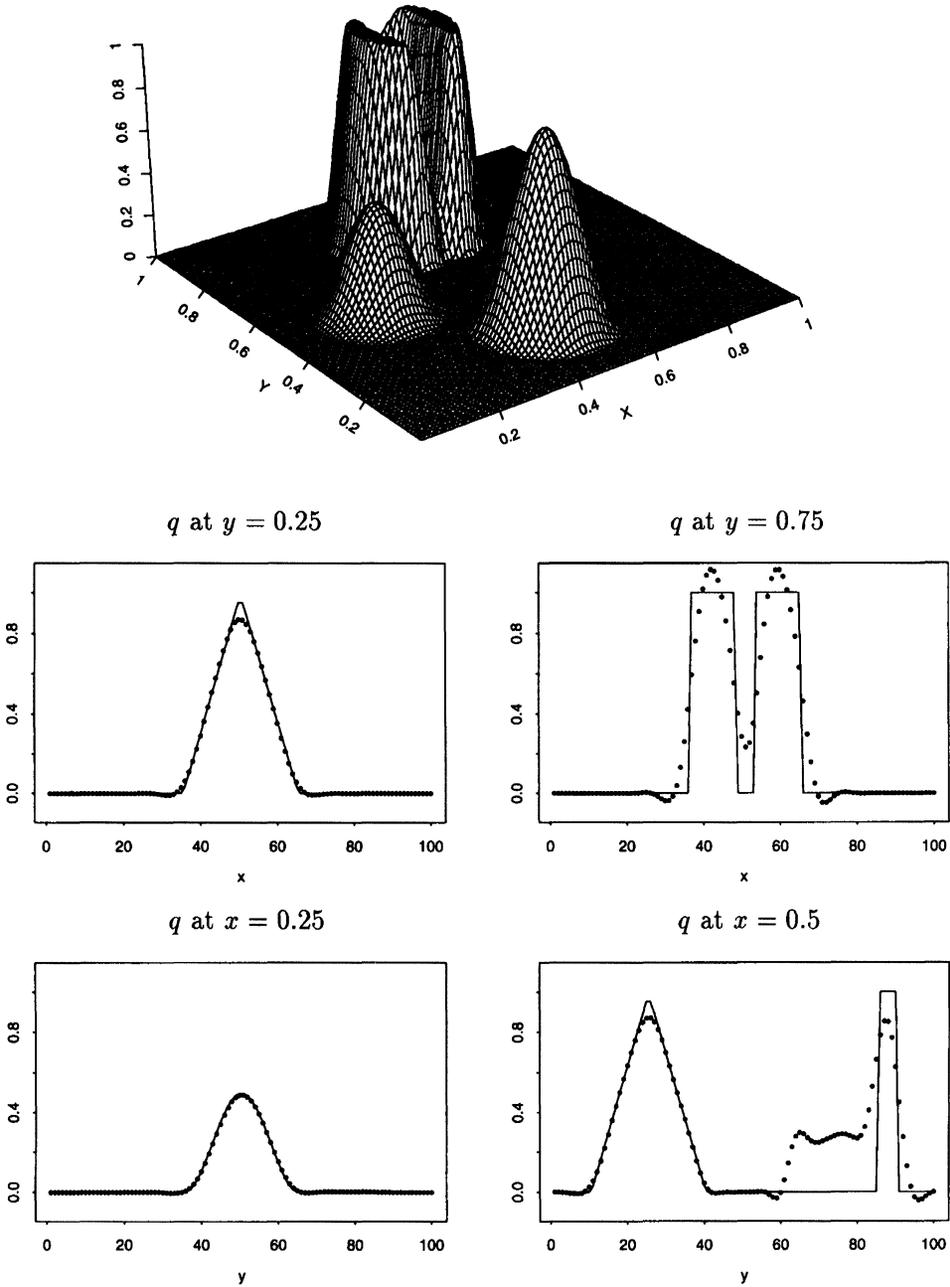


FIG. 9.7. Numerical results for Example 9.4 after one revolution (628 time steps) using Method 6 and no limiters. Four different cross sections of the solution are shown along with the perspective plot. The solid lines are the true solution.

below, consider the left boundary at $x = 0$ so that the boundary flux to be determined is $F_{j+1/2,0}$ (see Figure 10.1).

If no limiter is used then only one additional row of cells needs to be introduced. The flux value $F_{j+1/2,0}$ depends only on the values q_{im} for $i = 0, 1$ and $m = j - 1, j, j + 1$ and is determined in the usual way. If limiters are used then the jump $q_{1j} - q_{0j}$ may be limited by

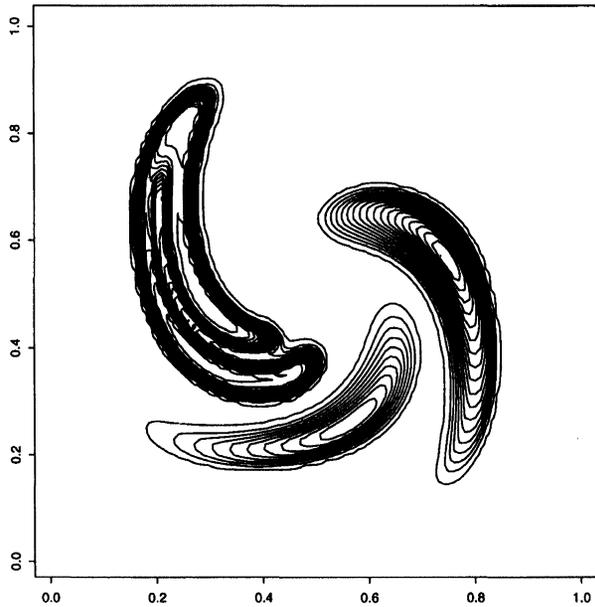


FIG. 9.8. Contour plot of numerical results at time $T/2$ for the deformation flow in Example 9.5. This is the time of maximum deformation.

comparing it with $q_{0j} - q_{-1,j}$ (if $u_{j+1/2,0} > 0$) and so a second row of exterior cells is needed near an inflow boundary.

In some of the test problems presented in the previous section, periodic boundary conditions were used. These are easily implemented simply by copying data from the opposite boundary. For example, if the original grid is $M \times M$ then we set

$$q_{0j}^n = q_{Mj}^n, \quad q_{-1,j}^n = q_{M-1,j}^n$$

at the beginning of each time step and similarly at the other boundaries.

In practice we often have inflow or outflow boundary conditions, or possibly no-flow boundary conditions at points where the normal velocity is zero, such as at an impermeable wall. These can all be handled by again introducing extra rows of cells with appropriately chosen values of q .

A no-flow boundary condition at $x = 0$ would correspond to $u_{j+1/2,0} = 0$. The correct flux at such a point is $F_{j+1/2,0} = 0$. Of course one could simply set $F_{j+1/2,0} = 0$ at such boundaries rather than introducing the exterior cells at all, but the introduction of exterior cells allows a variety of boundary conditions (or a mixture of different conditions at different points) within a unified framework, simplifying implementation. The waves originating at the interface between cells C_{0j} and C_{1j} (i.e., at the physical boundary) propagate with zero speed and have no effect on any fluxes. The wave originating at the interface between cells C_{0j} and $C_{0,j+1}$ also has no effect on the fluxes in the physical domain provided that $u_{0,j+1/2} \leq 0$. This can easily be arranged since this is outside the physical domain. The wave originating between cells C_{1j} and $C_{1,j+1}$ could also affect $F_{j+1/2,0}$ if $u_{1,j+1/2} < 0$. If so, this velocity should be set to zero. This amounts to simply disabling transverse propagation of waves that would cross the physical boundary. Note that this modification does not affect conservation. With these conditions satisfied, the values of q specified in the exterior cells are completely arbitrary and the resulting flux $F_{j+1/2,0}$ will be zero independent of the choice of q .

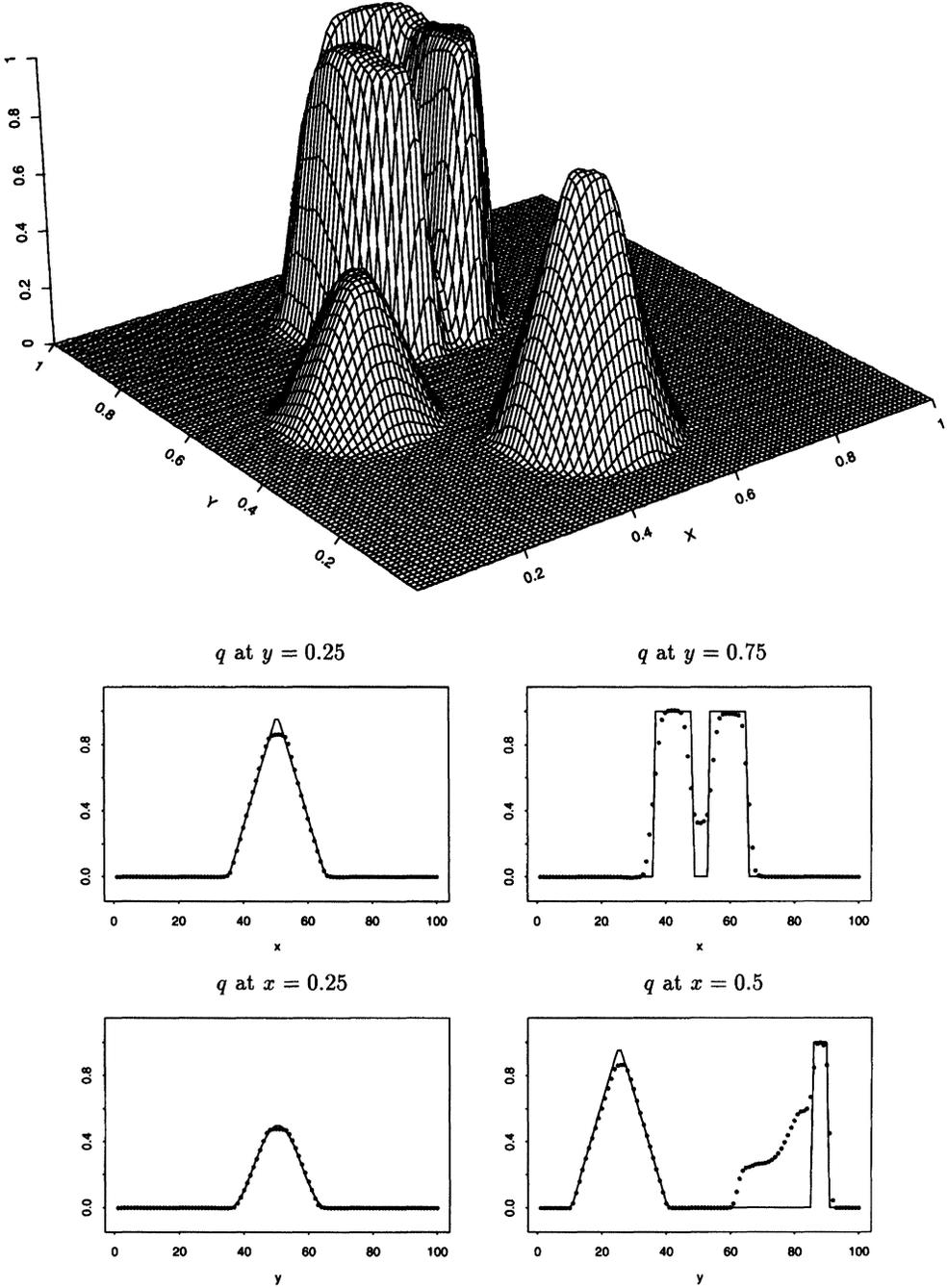


FIG. 9.9. Numerical results at time T for the deformation flow in Example 9.5 using Method 4 and the superbee limiter. Four different cross sections of the solution are shown along with the perspective plot. The solid lines are the true solution.

Outflow boundaries are characterized by the normal velocity being in the outward direction, e.g., $u_{j+1/2,0} < 0$. At such boundaries the increment wave goes outward and so in this step the value of q_0^n is again immaterial. The correction wave, however, does have an effect on q_{1j}^{n+1} and the choice of q_0^n affects the slope used in the correction wave. At an outflow

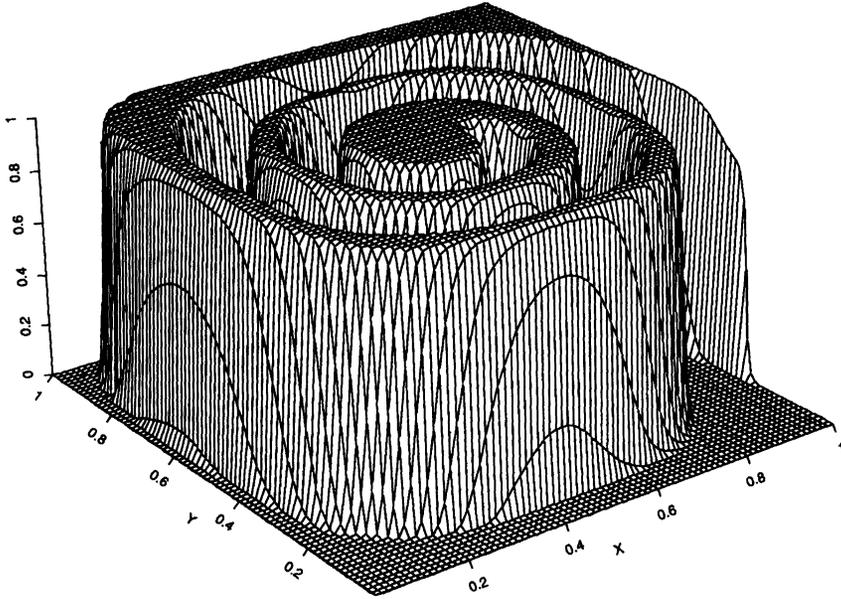


FIG. 9.10. Deformation flow of Example 9.6 computed with Method 4.

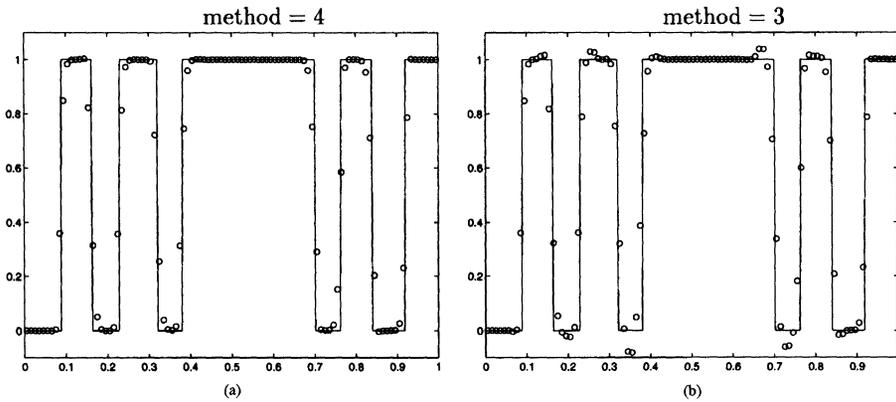


FIG. 9.11. (a) Cross section along $x = 0.5$ of the results from Figure 9.10 when Method 4 is used. (b) Cross section of a similar computation with Method 3. The solid line shows the true solution in each case.

boundary it is reasonable to use extrapolation from the interior of the domain to choose the values q_{0j}^n . Zero-order extrapolation gives

$$q_{0j}^n = q_{1j}^n.$$

Note that this results in zero slope and hence a correction wave of zero strength (i.e., we reduce to the fully upwind first-order Method 2 at an outflow boundary).

First-order (or linear) extrapolation would suggest

$$q_{0j}^n = 2q_{1j}^n - q_{2j}^n.$$

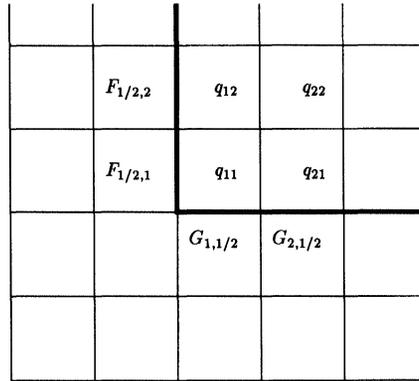


FIG. 10.1. Portion of extended grid used for imposing boundary conditions. The heavy line is the physical boundary and two additional rows of cells are added in each direction.

Note that in this case $q_{1j} - q_{0j} = q_{2j} - q_{1j}$ with the result that the limiter does not limit the slope. In practice this choice of extrapolation seems to work well and gives little error at an outflow boundary.

At an inflow boundary the normal velocity points inward, e.g., $u_{j+1/2,0} > 0$. At such a boundary the value $q(0, y, t)$ must be specified as part of the problem in order to determine a unique solution. For the finite volume method we must specify $q_{0j}^n \approx q(-h/2, y_j, t_n)$ and also $q_{-1,j} \approx q(-3h/2, y_j, t_n)$ for use in the limiter. One approach is to extrapolate from the known boundary data (and perhaps values in the interior grid), using, e.g.,

$$\begin{aligned} &\text{zero-order: } q_{0j}^n = q_{-1,j}^n = q(0, y_j, t_n) \\ \text{or} \quad &\text{first-order: } q_{0j}^n = 2q(0, y_j, t_n) - q_{1j}^n, \\ &q_{-1,j}^n = 4q(0, y_j, t_n) - 3q_{1j}^n. \end{aligned}$$

In practice first-order extrapolation works quite well.

One advantage of the wave propagation approach used here is that simple boundary conditions of the type proposed above are very robust and give no stability problems. This is not always true with standard finite difference methods, particularly centered methods, where one must be quite careful to avoid instabilities arising from the numerical boundary conditions. The upwind nature of the wave propagation approach seems to avoid these difficulties.

Figure 10.2(b) shows the results of a computation where linear extrapolation is used at both inflow and outflow boundaries. Method 4 is used with the monotized central limiter. The problem is the same as in Example 9.2, except that the computational domain is now $[-1, 0] \times [0, 1]$ and we solve over time $0 \leq t \leq \pi$. Then $q(x, y, 0) \equiv 0$ initially within the computational domain and the hump enters through the upper right corner, rotates through the domain, and leaves through the lower right corner. The exact solution is used as boundary data on the inflow portion of the boundary. In this computation $h = 0.025$ and $k = h/2$.

For comparison, Figure 10.2(a) shows the same computation over a larger domain $[-1, 1] \times [-0.5, 0.5]$ so that the hump is always within the domain. In each case the figure shows a superposition of the contours of the computed solution at six different times.

Figure 10.2(c) shows the max-norm of the error as a function of time for three different grids with $h = 0.05, 0.025, 0.0125$ and $k = h/2$. Second-order convergence is observed. Moreover we see that the error is very small for large t , after the hump has left the domain and the true solution is again identically zero. Although the computed solution does not

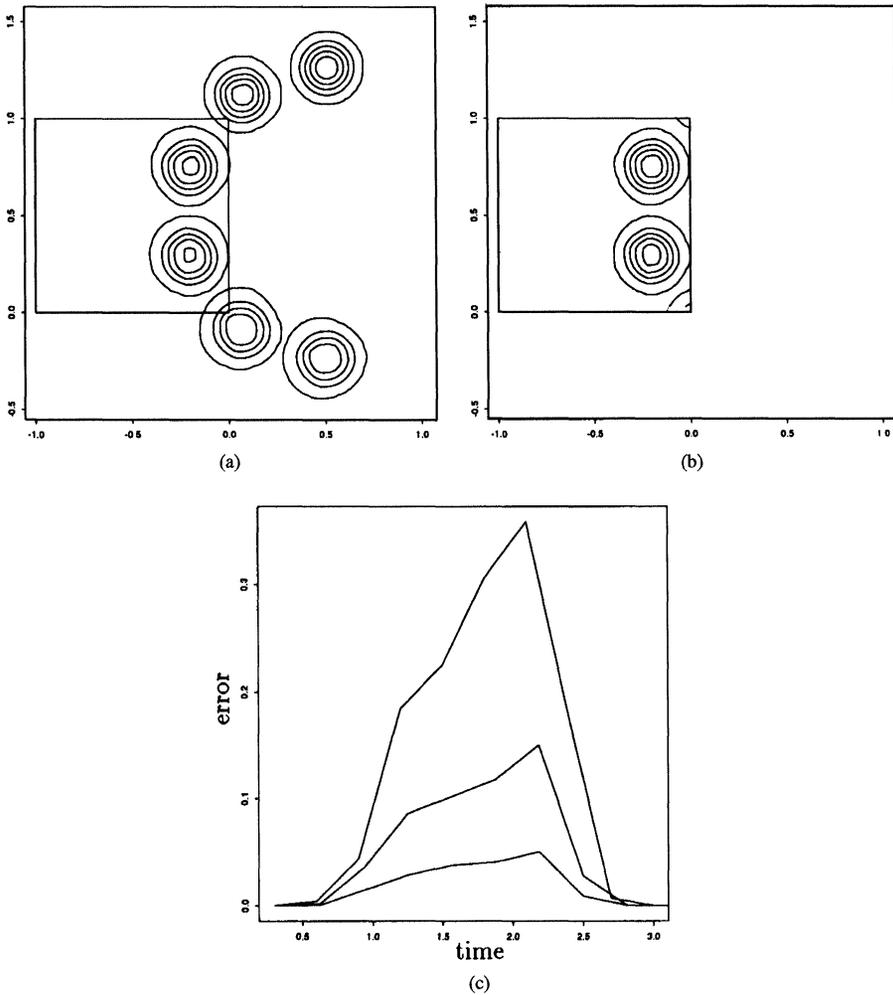


FIG. 10.2. (a) Rotating hump computation on a large domain with $h = 0.025$. Contour plots at six different times are superimposed (rotation is counterclockwise). Contour levels are at $q = 0.01, 0.1, 0.2, 0.3, 0.4$. (b) Rotating hump computation on the smaller domain with inflow and outflow boundary conditions. (c) Error as a function of time in the small grid computation. The error is shown for three different grids, with $h = 0.05, 0.025, 0.0125$ from top to bottom.

quite reach this pristine state, the residual error is much too small to show up on the scale of Figure 10.2(c). The final max-norm error at time $t = 3.125$ on each of the three grids is on the order of $10^{-5}, 10^{-13},$ and 10^{-21} , resp.

11. Three dimensions. The ideas and algorithms presented above carry over almost directly to three space dimensions. We now let q_{ijk} be the cell average of q over grid cell C_{ijk} . Since k is now a grid index, we use Δt for the time step and also use Δx ($= \Delta y = \Delta z$) for the grid spacing so there is no confusion with the flux in the z -direction, which is normally denoted by h . The numerical method takes the form

$$q_{ijk}^{n+1} = q_{ijk} - \frac{\Delta t}{\Delta x} [F_{i+1/2,j,k} - F_{i-1/2,j,k} + G_{i,j+1/2,k} - G_{i,j-1/2,k} + H_{i,j,k+1/2} - H_{i,j,k-1/2}].$$

We assume that the given velocity field $\vec{u} = (u, v, w)$ satisfies the discrete divergence-free condition

$$(u_{i+1/2,j,k}^{n+1/2} - u_{i-1/2,j,k}^{n+1/2}) + (v_{i,j+1/2,k}^{n+1/2} - v_{i,j-1/2,k}^{n+1/2}) + (w_{i,j,k+1/2}^{n+1/2} - w_{i,j,k-1/2}^{n+1/2}) = 0.$$

The first- and second-order algorithms are shown in Algorithm 11.1 in a form analogous to Algorithm 4.1. Note that here we incorporate the factor $\Delta t / \Delta x$ into the definition of V and W , which simplifies many of the expressions. For the most part this is a direct generalization of Algorithm 4.1. A few things require some comment. The first corrections made to the H fluxes (for the transverse propagation of the increment wave) include a new factor $\frac{1}{3}U|V|WR$. This is necessary for the proper corner coupling between cells in three dimensions. The increment wave propagating in the x -direction now has transverse motion in both the y - and z -directions. In general three additional cells will be affected by this wave, and hence three fluxes must be modified. (One could instead modify two G fluxes and one H flux to achieve the same effect.)

The final four modifications to H values result from the transverse propagation of the correction waves. Referring back to Figure 3.2(b), we should now think of propagating the piecewise linear function in the third direction, normal to the plane of the figure. This is accomplished by modifying four different H values corresponding to the four cells affected by this wave. Recall that this correction wave is linear in x but constant in y and will also be constant in z . So it is simply the correction S , weighted by the volume of overlap with each cell, that is used to modify the fluxes.

For the case in which $\vec{u}(\vec{x}, t) \equiv \text{const}$, it is again possible to achieve third-order accuracy with an additional minor modification. A truncation error analysis again shows that the transverse propagation already gives all the cross-derivative terms needed for third-order accuracy. The only missing terms are the pure third-order derivatives q_{xxx} , q_{yyy} , and q_{zzz} . Exactly the same modification as in Algorithm 4.1 gives the three-dimensional version of Method 6. Simply replace the line

$$S := \frac{1}{2}|U| \left(1 - \frac{\Delta t}{\Delta x}|U| \right) R$$

by

$$S := \frac{1}{2}|U| \left(1 - \frac{\Delta t}{\Delta x}|U| \right) R - \frac{1}{6}U \left(1 - \left(\frac{\Delta t}{\Delta x} \right)^2 U^2 \right) (q_{I+1,j,k} - 2q_{I,j,k} + q_{I-1,j,k}).$$

Note that in implementing this method (and also the two-dimensional version) it is not necessary to write out separate versions of this loop in the y - and z -directions. It is possible to write a single subroutine that is called three times with different assignments of u, v, w and F, G, H for each direction sweep. This simplifies implementation and debugging considerably. More details can be seen in the program `Advect3d.f`, which is available by anonymous ftp (see §12).

Example 11.1. The first numerical test is similar to Example 9.1. We use constant flow $u = v = w = 1$ with periodic boundary conditions and smooth initial data

$$(11.1) \quad q(x, y, z) = \sin(2\pi x) \sin(2\pi y) \sin(2\pi z).$$

The time step is $\Delta t = 0.8\Delta x$ and we compute up to time $t = 1$, when the initial data should be recovered. Figure 11.1 shows log-log plots of the error as the grid is refined. Three different $M \times M \times M$ grids are used with $M = 8, 16, 32$. Again the order of accuracy reported is based on the ratio of errors from the two finest grids.

Initialize increments and fluxes:

for each i, j, k do

$$F_{i-1/2,j,k} := 0, \quad G_{i,j-1/2,k} := 0, \quad H_{i,j,k-1/2} := 0$$

Update increments and fluxes based on interfaces in x -direction:

for each i, j, k do

consider interface between cells $C_{i-1,j,k}$ and C_{ijk} :

$$U := u_{i-1/2,j,k}^{n+1/2}, \quad V := \frac{\Delta t}{\Delta x} v_{i-1/2,j,k}^{n+1/2}, \quad W := \frac{\Delta t}{\Delta x} w_{i-1/2,j,k}^{n+1/2}$$

$$R := q_{ijk} - q_{i-1,j,k}$$

if $U > 0$ then $I := i - 1$ else $I := i$

$$F_{i-1/2,j,k} := F_{i-1/2,j,k} + Uq_{ijk}$$

if method = 1 then end loop here

if $U > 0$ then $I := i$ else $I := i - 1$

if $V > 0$ then $J := j + 1$ else $J := j$

$$G_{I,J-1/2,k} := G_{I,J-1/2,k} - \frac{1}{2}UVR$$

if $W > 0$ then $K := k + 1$ else $K := k$

$$H_{I,J,K-1/2} := H_{I,J,K-1/2} - \frac{1}{2}UWR + \frac{1}{3}U|V|WR$$

if $V > 0$ then $\hat{J} := j + 1$ else $\hat{J} := j - 1$

$$H_{I,\hat{J},K-1/2} := H_{I,\hat{J},K-1/2} - \frac{1}{3}U|V|WR$$

if method = 2 then end loop here

$R :=$ limited version of R

$$S := \frac{1}{2}|U| \left(1 - \frac{\Delta t}{\Delta x}|U| \right) R$$

$$F_{i-1/2,j,k} := F_{i-1/2,j,k} + S$$

if method = 3 then end loop here

$$G_{i,J-1/2,k} := G_{i,J-1/2,k} + VS$$

$$G_{i-1,J-1/2,k} := G_{i-1,J-1/2,k} - VS$$

$$H_{i,J,K-1/2} := H_{i,J,K-1/2} + (1 - |V|)WS$$

$$H_{i,\hat{J},K-1/2} := H_{i,\hat{J},K-1/2} + |V|WS$$

$$H_{i-1,J,K-1/2} := H_{i-1,J,K-1/2} - (1 - |V|)WS$$

$$H_{i-1,\hat{J},K-1/2} := H_{i-1,\hat{J},K-1/2} - |V|WS$$

if method = 4 then end loop here

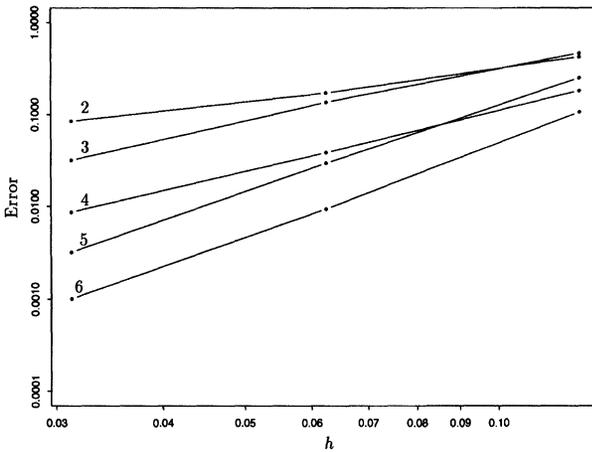
Update increments and fluxes based on interfaces in y -direction and

z -direction, as above but with velocities and fluxes appropriately switched.

Update q :

for each i, j, k do

$$q_{ijk}^{n+1} = q_{ijk} - \frac{\Delta t}{\Delta x} [F_{i+1/2,j,k} - F_{i-1/2,j,k} + G_{i,j+1/2,k} - G_{i,j-1/2,k} + H_{i,j,k+1/2} - H_{i,j,k-1/2}]$$



Order of accuracy		
Method	max-norm	1-norm
2	0.71	1.01
3	1.81	2.10
4	2.13	2.14
5	2.96	3.19
6	2.90	3.21

FIG. 11.1. Log-log plot of the 1-norm error vs. h for Methods 2-6 on the constant coefficient problem of Example 11.1, with $u = v = w = 1$.

Example 11.2. The deformation flow of Examples 9.5 and 9.6 is extended to three dimensions by superimposing deformation in the x - y plane with deformation in the x - z plane. The velocities are

$$\begin{aligned}
 (11.2) \quad & u(x, y, z) = 2 \sin^2(\pi x) \sin(2\pi y) \sin(2\pi z)g(t), \\
 & v(x, y, z) = -\sin(2\pi x) \sin^2(\pi y) \sin(2\pi z)g(t), \\
 & w(x, y, z) = -\sin(2\pi x) \sin(2\pi y) \sin^2(\pi z)g(t).
 \end{aligned}$$

The time dependence $g(t)$ is given by (9.6). Again the flow reverses at time $T/2$ so that the initial data should be recovered at time T . In this test discontinuous initial data is used of the form

$$q(x, y, z) = \begin{cases} 1, & \text{if } x < 1/2, \\ 0, & \text{if } x \geq 1/2. \end{cases}$$

The interface at $x = 1/2$ deforms in a truly three-dimensional manner and should return to its initial location at time $T = 1.5$. In this test a $40 \times 40 \times 40$ grid was used with $\Delta t = 0.5 \Delta x$, giving a Courant number of 1. Method 4 with the superbee limiter was used. Figure 11.2(a) shows a contour plot of q on a typical cross section $z = 0.425$ ($k = 17$) at time $T/2$, the time of maximum deformation. The interface appears disconnected only because we are slicing through fingers that form from the initial flat interface.

Figure 11.2(b) shows the same cross section at time T , when the initial interface should have been recovered. Of course the smearing introduced during the deformation will not be eliminated as the flow reverses, so the resolution seen here seems quite good. Only mild overshoots and undershoots are observed. The final values of q lie between $-.008$ and 1.008 everywhere and the integral of q is conserved.

12. Conclusions. A hierarchy of methods for advection in incompressible flowfields have been developed, based on multidimensional wave propagation. On the basis of numerous tests, the following recommendations can be made. As a general purpose algorithm, Method 4 with the van Leer or MC limiter is robust and effective. For problems with discontinuous solutions, the superbee limiter may be preferable. Method 6 works very well on smooth data but a better approach to applying limiters is needed for discontinuous data.

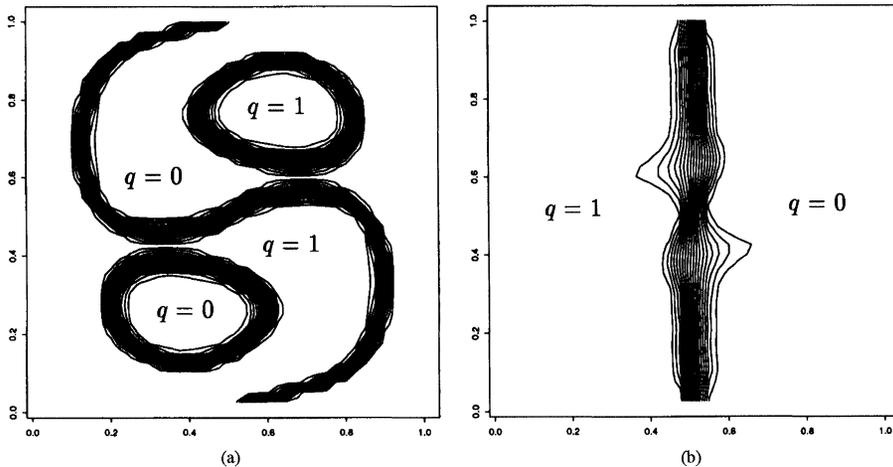


FIG. 11.2. Contour plots of cross sections at $z = 0.425$ for the three-dimensional deformation flow of Example 11.2 on a $40 \times 40 \times 40$ grid. Contour lines are at $q = 0.05j$, $j = 1, \dots, 19$. (a) Results at time $t = T/2$, when the initial interface at $x = 0.5$ is at maximum deformation. (b) The same slice at $t = T$, when the initial conditions should be recovered.

Fortran implementations of all of these algorithms are available by anonymous ftp from amath.washington.edu in the directory pub/rjl/programs/advection. These programs also include all of the flowfields and data needed to reproduce the results shown in this paper.

Note added in proof. The software package CLAWPACK (Conservation LAWS PACKAge) recently developed by the author extends this type of method to general nonlinear systems of conservation laws [24], [30]. In particular, this package can be applied to the advection equation and reduces to the method described here. This is easier to apply and modify than the original advection software mentioned above. Some further description of advection equations and examples can also be found in the User Notes [25], including conservative extensions to flowfields that are not incompressible and to curvilinear grids.

Acknowledgments. It is a pleasure to acknowledge Aaron Fogelson's contributions. His work on advection in the bloodstream [14] motivated this work and he also contributed valuable insights. In particular he suggested that transverse propagation of slopes should be further investigated, which led to improvements of the algorithm. I would also like to thank Dale Durran for suggesting several test problems and performing some comparisons. Several other people also read preliminary versions of this manuscript and made useful suggestions, including Chris Bretherton, Len Margolin, and Piotr Smolarkiewicz.

REFERENCES

- [1] J. B. BELL, P. COLELLA, AND H. M. GLAZ, *A second-order projection method for the incompressible Navier-Stokes equations*, J. Comput. Phys., 85 (1989), pp. 257–283.
- [2] J. B. BELL, P. COLELLA, AND L. H. HOWELL, *An efficient second-order projection method for viscous incompressible flow*, in Proc. 10th AIAA Conference on Computational Fluid Dynamics, Hawaii, 1991, pp. 360–367.
- [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 (1988), pp. 1–24.
- [4] J. B. BELL AND D. L. MARCUS, *A second-order projection method for variable-density flows*, J. Comput. Phys., 101 (1992), pp. 334–348.
- [5] J. P. BORIS AND D. L. BOOK, *Flux corrected transport I, SHASTA, a fluid transport algorithm that works*, J. Comput. Phys., 11 (1973), pp. 38–69.
- [6] A. BOTT, *Monotone flux limitation in the area-preserving flux-form advection algorithm*, Monthly Weather Rev., 120 (1992), pp. 2592–2602.

- [7] A. J. CHORIN, *Numerical solution of the Navier-Stokes equations*, Math. Comp., 22 (1968), pp. 745–762.
- [8] P. COLELLA, *Multidimensional upwind methods for hyperbolic conservation laws*, J. Comput. Phys., 87 (1990), pp. 171–200.
- [9] P. COLELLA AND P. WOODWARD, *The piecewise-parabolic method (PPM) for gas-dynamical simulations*, J. Comput. Phys., 54 (1984), pp. 174–201.
- [10] W. P. CROWLEY, *Numerical advection experiments*, Monthly Weather Rev., 96 (1968), pp. 1–11.
- [11] J. DOUGLAS, JR. AND T. F. RUSSELL, *Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures*, SIAM J. Numer. Anal., 19 (1982), pp. 871–885.
- [12] J. K. DUKOWICZ AND J. D. RAMSHAW, *Tensor viscosity method for convection in numerical fluid dynamics*, J. Comput. Phys., 32 (1979), pp. 71–79.
- [13] R. C. EASTER, *Two modified versions of Bott's positive-definite numerical advection scheme*, Monthly Weather Rev., 121 (1993), pp. 297–304.
- [14] A. L. FOGELSON, *A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting*, J. Comput. Phys., 56 (1984), pp. 111–134.
- [15] F. H. HARLOW AND J. E. WELCH, *The MAC method: a computing technique for solving viscous, incompressible, transient fluid flow problems involving free surfaces*, Phys. Fluids, 8 (1965), pp. 2182–2189.
- [16] A. HARTEN, *High resolution schemes for hyperbolic conservation laws*, J. Comput. Phys., 49 (1983), pp. 357–393.
- [17] A. HARTEN AND G. ZWAS, *Self-adjusting hybrid schemes for shock computations*, J. Comput. Phys., 9 (1972), p. 568.
- [18] P. D. LAX AND B. WENDROFF, *Systems of conservation laws*, Comm. Pure Appl. Math., 13 (1960), pp. 217–237.
- [19] ———, *Difference schemes for hyperbolic equations with high order of accuracy*, Comm. Pure Appl. Math., 17 (1964), pp. 381–398.
- [20] C. E. LEITH, *Numerical simulation of the Earth's atmosphere*, Methods Comput. Phys., 4 (1965), pp. 1–28.
- [21] B. P. LEONARD, *Universal limiter for transient interpolation modeling of the advective transport equations: the ULTIMATE conservative difference scheme*, NASA Technical Memorandum 100916, NASA Lewis, 1988.
- [22] ———, *Positivity-preserving numerical schemes for multidimensional advection*, NASA Technical Memorandum 106055, ICOMP-93-05, NASA Lewis, 1993.
- [23] B. P. LEONARD, M. K. MACVEAN, AND A. P. LOCK, *Positivity-Preserving Numerical Schemes for Multidimensional Advection*, NASA Technical Memorandum 106055, ICOMP-93-05, NASA Lewis, 1993.
- [24] R. J. LEVEQUE, *CLAWPACK software*, available from netlib.att.com in netlib/pdes/claw or on the Web at the URL <ftp://amath.washington.edu/pub/rjl/programs/clawpack.html>.
- [25] ———, *CLAWPACK User Notes*, available from netlib.att.com in netlib/pdes/claw/doc or on the Web at the URL <ftp://amath.washington.edu/pub/rjl/programs/clawpack.html>.
- [26] ———, *High resolution finite volume methods on arbitrary grids via wave propagation*, J. Comput. Phys., 78 (1988), pp. 36–63.
- [27] ———, *Hyperbolic conservation laws and numerical methods*, Von Karman Institute for Fluid Dynamics Lecture Series, Von Karman Institute for Fluid Dynamics, Rhode-St. Genese, Belgium, 1990, pp. 1–137.
- [28] ———, *Numerical Methods for Conservation Laws*, Birkhäuser-Verlag, Berlin, 1990.
- [29] ———, *Simplified multi-dimensional flux limiter methods*, in Numerical Methods for Fluid Dynamics Vol. 4, M. J. Baines and K. W. Morton, eds., Oxford University Press, 1993, pp. 175–190. (<ftp://amath.washington.edu/pub/rjl/papers/reading92.ps.Z>).
- [30] ———, *CLAWPACK—a software package for solving multi-dimensional conservation laws*, in Proc. 5th Internat. Conf. Hyperbolic Problems, 1994. (<ftp://amath.washington.edu/pub/rjl/papers/hyp94.ps.Z>).
- [31] P. LIN AND K. W. MORTON, *Characteristic Galerkin Schemes for Scalar Conservation Laws in Two Space Dimensions I: Formulation*, Tech Report 92/1, Oxford University Computing Laboratory, 1992.
- [32] P. LIN, K. W. MORTON, AND E. SÜLLI, *Characteristic Galerkin schemes for scalar conservation laws in two and three space dimensions*, SIAM J. Numer. Anal., to appear.
- [33] R. LISKA, personal communication, Czech Technical University in Prague, 1995.
- [34] R. W. MACCORMACK, *The effects of viscosity in hypervelocity impact cratering*, AIAA Paper 69-354, 1969.
- [35] K. W. MORTON, *Generalised Galerkin methods for hyperbolic problems*, Comput. Methods Appl. Mech. Engrg., 52 (1985), pp. 847–871.
- [36] K. W. MORTON AND P. K. SWEBY, *A comparison of flux-limited difference schemes and characteristic Galerkin methods for shock modelling*, J. Comput. Phys., 73 (1987), pp. 203–230.
- [37] J. E. PILLIOD, JR. AND E. G. PUCKETT, *Second-order volume-of-fluid algorithms for tracking material interfaces*, preprint, 1993.
- [38] Y. B. RADVOGIN, *Quasi-monotonous multidimensional differential schemes with second order accuracy*, Soviet Academy of Sciences, preprint 1991. (in Russian)
- [39] P. L. ROE, *The Use of the Riemann Problem in Finite-Difference Schemes*, Lecture Notes in Physics 141, Springer-Verlag, Berlin, New York, 1981.

- [40] P. L. ROE, *Fluctuations and signals—a framework for numerical evolution problems*, in *Numerical Methods for Fluid Dynamics*, K. W. Morton and M. J. Baines, eds., Academic Press, New York, San Diego, CA, 1982, pp. 219–257.
- [41] ———, *Some Contributions to the Modeling of Discontinuous Flows*, *Lecture Notes in Applied Mathematics*, 22, Springer-Verlag, Berlin, New York, 1985, pp. 163–193.
- [42] ———, *Linear advection schemes on triangular meshes*, CoA Report No. 8720, Cranfield Institute of Technology, Cranfield, England, 1987.
- [43] P. L. ROE AND D. SIDILKOVER, *Optimum positive linear schemes for advection in two and three dimensions*, *SIAM J. Numer. Anal.*, 29 (1992), pp. 1542–1568.
- [44] R. B. ROOD, *Numerical advection algorithms and their role in atmospheric transport and chemistry models*, *Rev. Geophys.*, 25 (1987), pp. 71–100.
- [45] J. SALTZMAN, *Monotonic Difference Schemes for the Linear Advection Equation in Two and Three Dimensions*, Los Alamos Report LA-UR-87-2479, Los Alamos, NM, 1987.
- [46] ———, *An unsplit 3-D upwind method for hyperbolic conservation laws*, *J. Comput. Phys.*, (1994), pp. 153–168.
- [47] C. SCHULZ-RINNE, *The Riemann Problem for Two-Dimensional Gas Dynamics and New Limiters for High-Order Schemes*, Ph.D. thesis, ETH-Zürich, 1993.
- [48] J. S. SCROGGS AND F. H. M. SEMAZZI, *A Conservative Semi-Lagrangian Method for Multi-Dimensional Fluid Dynamics Applications*, *Numer. Methods. Partial Differential Equations*, 11 (1995), pp. 445–452.
- [49] P. K. SMOLARKEWICZ AND W. W. GRABOWSKI, *The multidimensional positive definite advection transport algorithm: Nonoscillatory option*, *J. Comput. Phys.*, 86 (1990), pp. 355–375.
- [50] P. K. SMOLARKEWICZ AND G. A. GRELL, *A class of monotone interpolation schemes*, *J. Comput. Phys.*, 101 (1992), pp. 431–440.
- [51] P. K. SMOLARKEWICZ, *The multi-dimensional Crowley advection scheme*, *Monthly Weather Rev.*, 110 (1982), pp. 1968–1983.
- [52] ———, *A fully multidimensional positive definite advection transport algorithm with small implicit diffusion*, *J. Comput. Phys.*, 54 (1984), pp. 325–362.
- [53] P. K. SMOLARKEWICZ AND T. L. CLARK, *The multidimensional positive definite advection transport algorithm: Further development and applications*, *J. Comput. Phys.*, 67 (1986), pp. 396–438.
- [54] P. K. SMOLARKEWICZ AND L. G. MARGOLIN, *On forward-in-time differencing for fluids: Extension to a curvilinear framework*, *Monthly Weather Rev.*, 121 (1993), pp. 1847–1859.
- [55] A. STANFORTH AND J. CÔTÉ, *Semi-Lagrangian integration schemes for atmospheric models—a review*, *Monthly Weather Rev.*, 119 (1991), pp. 2206–2223.
- [56] J. C. STRIKWERDA, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole, 1989.
- [57] P. K. SWEBY, *High resolution schemes using flux limiters for hyperbolic conservation laws*, *SIAM J. Numer. Anal.*, 21 (1984), pp. 995–1011.
- [58] J. A. TRANGENSTEIN, *An Unsplit Godunov Method for Three-Dimensional Polymer Flooding*, preprint, 1993.
- [59] B. VAN LEER, *Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme*, *J. Comput. Phys.*, 14 (1974), pp. 361–370.
- [60] ———, *Towards the ultimate conservative difference scheme IV. A new approach to numerical convection*, *J. Comput. Phys.*, 23 (1977), pp. 276–299.
- [61] G. YEH AND J. R. CHANG, *An exact peak capturing and oscillation free scheme to solve advection-dispersion transport equations*, *Water Resources Res.*, 28 (1992), pp. 2937–2951.
- [62] S. T. ZALESAK, *Fully multidimensional flux corrected transport algorithms for fluids*, *J. Comput. Phys.*, 31 (1979), pp. 335–362.
- [63] ———, *A preliminary comparison of modern shock-capturing schemes: Linear advection*, in *Advances in Computer Methods for Partial Differential Equations*, VI, R. Vichnevetsky and R. S. Stepleman, eds., IMACS, 1987, pp. 15–22.