Journal of Computational Physics **160,** 318–335 (2000)
doi:10.1006/jcph.2000.6465, available online at http://www.idealibrary.com on IDEAL®

Incremental Remapping as a Transport/ Advection Algorithm

John K. Dukowicz* and John R. Baumgardner†

Theoretical Division, Los Alamos National Laboratory, Group T-3, Mail Stop B216,
Los Alamos, New Mexico 87545
E-mail: *Duk@lanl.gov, †Baumgardner@lanl.gov

Received August 24, 1999; revised February 3, 2000

There are two fundamentally different strategies for solving the standard transport or continuity equation, corresponding to whether it is expressed as a partial differential equation or as an integral statement of conservation. The more common approach is to discretize the partial differential equation and to march the solution forward in time. The alternative method is to project cell volumes along Lagrangian trajectories as far forward or backward in time as desired, and then to remap the resulting density distribution onto some target mesh. This latter approach is known as remapping. Remapping has many advantages, not the least of which is that the time step is limited only by accuracy considerations, but it tends to be expensive and complex. In this paper we show that if the time step is made sufficiently short such that trajectories are confined to the nearest neighbor cells, then the remapping may be written as a flux-form transport algorithm, and it becomes nearly as simple and efficient as standard transport schemes. The resulting method, called incremental remapping, retains most of the advantages of general remapping. These include: (a) geometric basis for transport, (b) compatibility of associated tracer transport with simple tracer advection, i.e., retention of tracer monotonicity properties, and (c) efficient handling of multiple tracers since each additional tracer adds only a relatively small incremental cost.

Key Words: remapping; transport; advection; semi-lagrange.

1. INTRODUCTION

Let us consider the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0, \tag{1}$$

which is alternatively called the transport or advection equation, where $\mathbf{u}$ is the velocity

0021-9991/00

and ρ is typically the density as in compressible fluid dynamics, although it may be some other quantity depending on the application, such as layer thickness in a Boussinesq isopycnal ocean model, for example. The solution of this type of equation is perhaps the most ubiquitous problem in all fields of computational fluid dynamics because it represents a fundamental property of the flow such as conservation of mass or the conservation of volume, as the case may be. Associated with this is the equation for the transport of a passive scalar or a tracer,

$$\frac{\partial \rho T}{\partial t} + \nabla \cdot \rho \mathbf{u} T = 0, \tag{2}$$

where T is the concentration of the tracer per unit mass. Clearly, (1) and (2) together imply that

$$\frac{dT}{dt} = 0, \tag{3}$$

where $\frac{d}{dt} \equiv \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$ is the total derivative, indicating that T is conserved along trajectories or characteristics, as a tracer should be. There are many examples of such tracers. In situations when diffusion effects are negligible, two examples are nonreactive pollutants in the atmosphere and salinity in the ocean. Of course, the density ρ is also effectively a tracer in the important case when the flow is incompressible ($\nabla \cdot \mathbf{u} = 0$) since (1) is then equivalent to (3). Even when these equations do not strictly apply, i.e., when the right-hand sides are nonzero, the transport operators represented by the left-hand sides of these equations are important components of all conservation equations, and therefore the accurate solution of such equations is of fundamental importance in practice.

Equations (1) and (2) are partial differential equations that are typically solved for ρ and T by means of some time-explicit finite-difference or finite-element discretization scheme in which the solution is marched forward in time in a series of finite time steps. There are innumerable schemes for the solution of these equations, and the literature describing such schemes is so extensive that it would serve no purpose to give particular references. The classical schemes are well described in textbooks (e.g., [1–3]), and new schemes are constantly being devised. These schemes are generally characterized by the following:

(a) Conservation: The equations are associated with conservation laws and are in conservation form. The numerical schemes are therefore expressed in terms of fluxes across cell faces which automatically assures conservation.

(b) CFL condition: Numerical stability of these schemes on a finite grid generally requires that the time step be limited by a Courant–Friedrichs–Lewy condition of the form $\mathrm{Max}|\mathbf{u}|\Delta t/\Delta x \leq 1$.

(c) Discretization errors: The schemes may be of arbitrary order of accuracy. However, numerical solutions typically contain diffusion and/or dispersion errors. The diffusion errors in particular are cumulative and so they tend to grow over a large number of time steps. Much effort is devoted to upwinding methods and/or positivity or monotonicity-preserving methods which attempt to control dispersion errors.

(d) Compatibility error: Typically, Eqs. (1) and (2) are solved for ρ and ρT, respectively, and T is then obtained from the ratio of the two prognostic quantities. An important characteristic of a tracer satisfying (3) is that no new extrema in the concentration T are created since T is unchanged along trajectories; i.e., T satisfies a monotonicity property.

Although ρ and ρT may both be obtained using conservative and monotonicity-preserving schemes, there is no guarantee that the resulting tracer T will satisfy the tracer monotonicity property associated with (3). Although this lack of compatibility is also a discretization error, it is of a different type than the errors referred to in (c) above and is much less frequently discussed in the literature. Specific methods attempting to preserve compatibility are presented in Schär and Smolarkiewicz [4] and in VanderHeyden and Kashiwa [5], for example.

There is an alternative approach to the solution of these equations known as remapping. We observe that (1) is a conservation equation equivalent to

$$\frac{d}{dt}\int_{V_L(t)} \rho\, dV = 0, \tag{4}$$

where dV is the element of volume, and the integration is over an arbitrary finite Lagrangian volume $V_L(t)$ at time t, that is, a volume whose bounding surface moves with the local fluid velocity, or alternatively, a volume which always contains the same material particles. Correspondingly, (2) is equivalent to

$$\frac{d}{dt}\int_{V_L(t)} \rho T\, dV = 0. \tag{5}$$

These two equations represent the conservation property directly, and so in a sense they are more fundamental than Eqs. (1)–(2). The two types of equations may be related to each other by means of Reynolds' transport theorem [6].

Equations (4) and (5) indicate that the mass and the total tracer of a Lagrangian volume are conserved along trajectories. They may therefore be used to solve for the time evolution of ρ and T. Defining the mass and total tracer of the Lagrangian control volume to be

$$M(t) = \int_{V_L(t)} \rho\, dV, \qquad Q(t) = \int_{V_L(t)} \rho T\, dV, \tag{6}$$

respectively, a discretization of these two equations is simply given by

$$M(t^{n+1}) = M(t^n), \qquad Q(t^{n+1}) = Q(t^n), \tag{7}$$

where the superscript n stands for the current time and $n+1$ stands for some unspecified future time. Defining Lagrangian cell-average density to be

$$\bar{\rho}(t) = \int_{V_L(t)} \rho\, dV \bigg/ \int_{V_L(t)} dV = M(t)/V_L(t),$$

we obtain

$$\bar{\rho}^{n+1} = M(t^{n+1})/V_L(t^{n+1}) = M(t^n)/V_L(t^{n+1}) = \bar{\rho}^n V_L(t^n)/V_L(t^{n+1}), \tag{8}$$

where $V_L(t^{n+1})$ is the Lagrangian volume obtained by integrating trajectories to time t^{n+1}. Similarly, defining the cell-average tracer to be

$$\tilde{T}(t) = \int_{V_L(t)} \rho T\, dV \bigg/ \int_{V_L(t)} \rho\, dV = Q(t)/M(t),$$

we obtain

$$\tilde{T}^{n+1} = Q(t^{n+1})/M(t^{n+1}) = Q(t^n)/M(t^n) = \int_{V_L(t^n)} \rho^n T^n \, dV \Big/ \int_{V_L(t^n)} \rho^n \, dV. \quad (9)$$

Equation (9) is a remarkable result. It indicates that the new-time $\tilde{T}$ is obtained via an averaging operator over old-time values of density and tracer, and therefore has the property

$$\operatorname*{Min}_{V_L(t^n)}[T^n] \le \tilde{T}^{n+1} \le \operatorname*{Max}_{V_L(t^n)}[T^n], \quad (10)$$

provided $\rho^n > 0$ everywhere. This property is equivalent to compatibility as previously defined.

Equations (8) and (9) may be interpreted and applied in two distinct ways:

(a) Forward trajectories: We will simplify the following by limiting ourselves to a discussion of the remapping of mass, noting that (4) is a special case of (5) with $T = 1$. The current grid with known cell masses and volumes is projected to the future time t^{n+1}. Equation (8) gives the new mean cell density on the new Lagrangian grid, denoted by the subscript L. Using these mean densities, we may reconstruct the distribution of density, $\rho_L^{n+1}(\mathbf{r})$, over the entire domain. The simplest reconstruction is to assume that the density is constant within each cell, $\rho_L^{n+1}(\mathbf{r}) = \overline{\rho}^{n+1}$, resulting in a first-order approximation. We now wish to transfer the new-time mass and mean density to some target grid, which typically will be the same as the current grid if one is interested in an “Eulerian” method. This is obtained as follows,

$$M_T^{n+1} = \int_{V_T} \rho_L^{n+1} \, dV, \quad (11)$$

and

$$\overline{\rho}_T^{n+1} = M_T^{n+1} \big/ V_T, \quad (12)$$

where the subscript T stands for the target grid. Equation (11) implies integration on the overlap of grids T and L.

(b) Backward trajectories: The target grid, which is the desired grid at time t^{n+1}, is projected back in time to the departure points of the trajectories at time t^n. This gives us the Lagrangian grid L at time t^n, i.e., $V_L(t^n)$. However, we have the mean densities at time t^n on our starting grid, which we assume for simplicity to be the same as the target grid. This density may be reconstructed to give the density field $\rho_T^n(\mathbf{r})$. Equations (7) and (8) are now interpreted to give

$$M_T^{n+1} = \int_{V_L(t^n)} \rho_T^n \, dV \quad (13)$$

and

$$\overline{\rho}_T^{n+1} = M_T^{n+1} \big/ V_T. \quad (14)$$

We note that (13) again implies integration on the overlap of grids T and L. Note that in this case the reconstruction takes place on the target grid, which may be somewhat simpler because the target grid is usually more regular than the Lagrangian grid.

We now summarize some characteristics of methods employing remapping:

(a) Conservation: This is inherent in the construction of these methods provided the reconstructed density and tracer satisfy

$$\int_V \rho\, dV = \bar{\rho} V, \qquad \int_V \rho T\, dV = \bar{\rho} V \tilde{T}, \tag{15}$$

when integrated over each cell, and $\bar{\rho}$, $\tilde{T}$ are the known or specified mean values in each cell that are the data used in the reconstruction. Note that this implies that $\tilde{T}$ is the density-weighted mean.

(b) CFL condition: There is no inherent time step limit. In principle, one can go forward or backward along trajectories as far as one wishes. However, there is a practical limit of the form $\text{Max}|\nabla \mathbf{u}|\Delta t < 1$ to prevent trajectory intersections [7]. There is a price to be paid for the potentially longer time step: grid L is in principle arbitrarily located with respect to the grid T, and to go from one to the other requires a costly search when the CFL condition is not satisfied.

(c) Discretization errors: Typically, remapping methods are either first- or second-order accurate, depending on whether the reconstructed density and tracer are constant or linear within a cell. First-order methods are inherently positivity and monotonicity preserving, in analogy with upwind or donor-cell schemes, but are very diffusive. Second-order schemes may be made positivity and monotonicity preserving by limiting the gradients of density and tracer within each cell in the reconstruction phase. Higher-order reconstruction methods might be possible at the expense of much greater complexity.

(d) Compatibility error: Compatibility is automatic for positivity- and monotonicity-preserving methods. We observe that the final mean tracer in the target cell is given by an averaging operator as in (9), provided $\rho > 0$. Therefore, if the reconstructed tracer within each cell satisfies a monotonicity constraint, $\text{Min}[\tilde{T}^n] < T^{n+1} < \text{Max}[\tilde{T}^n]$, among its neighbors, then the mean tracer $\tilde{T}^{n+1}$ will also satisfy this monotonicity constraint and hence it will satisfy compatibility, as defined previously.

(e) Multiple tracers: Much of the computation required for remapping is geometrical in nature and is common for each tracer. If this computation is reused, then the extra computational work for each additional tracer is very small. This is in contrast to standard transport or advection schemes where generally the entire algorithm must be repeated for each additional tracer.

Despite its many advantages, remapping is much less common than its alternative, i.e., flux-based difference schemes, primarily because it is generally more expensive and more complicated to implement. Nevertheless, it is very useful in certain applications, and methods have been found to overcome many of the difficulties. Fully general, second-order accurate methods in 2D are presented in [8, 9], a first-order 3D method is given in [10], and second-order 3D methods are given in [11, 12]. Methods based on multidimensional splitting utilizing 1D remapping are described in [13, 14]. In the present paper we are concerned with conservative remapping methods only. However, there exists a nonconservative analog of remapping, based on Eq. (3), under the broad heading of semi-Lagrangian methods (for a review, see [15]). There is an extensive literature, primarily confined to meteorological applications.

The present paper is based on the fact that remapping and advection or transport algorithms are equivalent in the limit of short time steps. If the time step of a remapping scheme is restricted such that trajectories are confined to the immediately neighboring cells (we call this an incremental remapping) then two major advantages accrue:

(1) There is no need to search for the cells of one mesh within another (generally a very costly procedure), and

(2) the algorithm is expressible in terms of cell-face fluxes, which are reused for neighboring cells, thereby saving computational work.

The resulting algorithm is analogous to the rezone phase ("phase three") of ALE methods [16], but it is much more accurate. It is exactly equivalent to a full-scale remapping, except for the relatively short Lagrangian displacement, and it therefore retains all the advantages of a remapping mentioned previously, such as automatic compatibility and economical handling of multiple tracers. Thus, at the price of a CFL-like restriction we are able to obtain a simplified remapping algorithm that appears to be a hybrid between advection and remapping, retaining the advantages of both methods.

2. INCREMENTAL BACKWARD-TRAJECTORY REMAPPING

In what follows, we will illustrate two-dimensional, backward-trajectory incremental remapping on a regular Cartesian mesh. This is for ease of presentation only; the basic ideas presented can be easily extended to more complicated alternatives. Let us assume that we wish to integrate

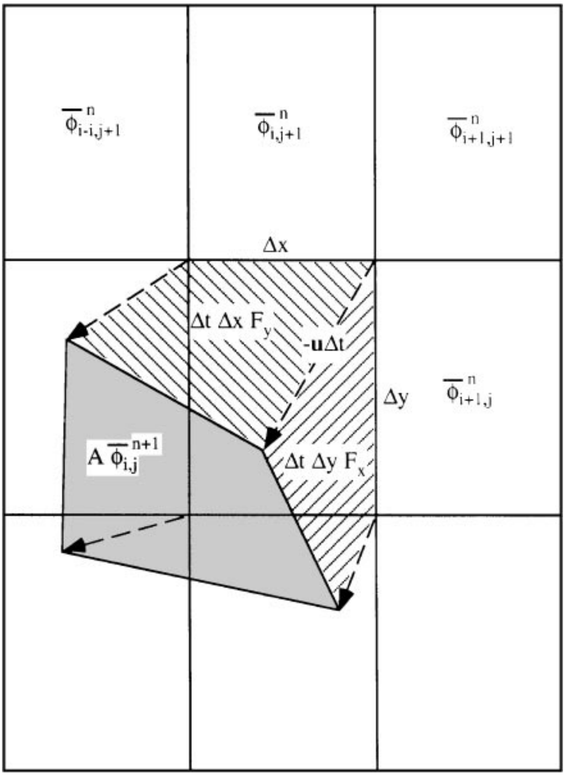
$$\frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{F} = 0, \tag{16}$$

where ϕ is the transported concentration, such as ρ, ρT, etc., and $\mathbf{F}$ is the corresponding flux, as in Eqs. (1) and (2). A discrete version of the integral form of this equation on a 2D Cartesian mesh may be expressed as

$$A\overline{\phi}^{n+1} - A\overline{\phi}^{n} + \Delta t(\Delta y \Delta_x F_x + \Delta x \Delta_y F_y) = 0, \tag{17}$$

where $\overline{\phi}$ is the cell area average of ϕ, $A\overline{\phi} = \int_A \phi \, dA$, $A = \Delta x \, \Delta y$ is the area of the cell, Δx, Δy are the cell dimensions, $\Delta t = t^{n+1} - t^n$ is the time step, (F_x, F_x) are the average normal fluxes on a cell face, and (Δ_x, Δ_y) are difference operators in the x- and y-directions, respectively. This is a standard flux-form discretization of (16), and various schemes differ only in the definition of the fluxes. Let us assume that velocities are defined on cell vertices. It then turns out that (17) also corresponds to a backward-trajectory remapping, as illustrated in Fig. 1.

It will be useful to review the discussion of the backward-trajectory remapping from the Introduction as it applies to this case. Consider Fig. 1. The components of the vertex velocities are all assumed to be positive in this case. The backward trajectories from the vertices of the target or home cell, labeled with indices (i, j), are indicated by dashed vectors. The Lagrangian cell, shown as the shaded quadrilateral, is that portion of the domain at time t^n that will arrive in the target cell at time t^{n+1}. It may therefore be called the departure cell. The total quantity of ϕ in the target cell (i, j) at time t^{n+1}, indicated by $A\overline{\phi}_{i,j}^{n+1}$, is given by the integral of ϕ^n over the four small quadrilaterals comprising the departure cell, i.e., those subcells overlapping cells (i, j), $(i-1, j)$, $(i-1, j-1)$, and $(i, j-1)$. However,



$\overline{\phi}^{n}_{i-i,j+1}$
$\overline{\phi}^{n}_{i,j+1}$
$\overline{\phi}^{n}_{i+1,j+1}$
Δx
$\Delta t\ \Delta x\ F_y$
$-\mathbf{u}\Delta t$
Δy
$\overline{\phi}^{n}_{i+1,j}$
$A\ \overline{\phi}^{n+1}_{i,j}$
$\Delta t\ \Delta y\ F_x$

FIG. 1. Backward-trajectory remapping of a typical two-dimensional cell, showing the departure cell, the arrival cell, and the fluxing areas associated with cell faces.

rather than having to compute these overlap integrals directly, we may alternatively obtain the same result in terms of cell-face fluxes. The net sum of partial area integrals is exactly equivalent to (17), where the fluxing areas such as $\Delta t \Delta x F_y$ are the integrals over the hatched areas shown in the figure. We observe that each fluxing area is associated with a cell face and that it is also required in connection with the departure cell of the neighboring cell; it is therefore reusable when computing the neighboring departure cell. This is a considerable economy, both in computational work and in programming logic.

2.1. *Fluxing Areas*

We will illustrate the computation of fluxes and fluxing areas by focusing on the fluxing area $\Delta t \Delta x F_y$ at the top or north face of the target cell. Fluxes associated with all other faces may be obtained by a suitable translation and/or rotation. In the following, we will simplify the discussion by referring to the target cell as the home cell (H), and neighboring cells and faces by compass directions, i.e., north (N), northeast (NE), etc. Consider Fig. 2 and the notation introduced therein. The west vertex is labeled 1, the east vertex is labeled 2, and the corresponding vertex displacements are the endpoints of backward trajectories: $\mathbf{x}_1 = -\mathbf{u}_1 \Delta t = (x_1, y_1)$ and $\mathbf{x}_2 = -\mathbf{u}_2 \Delta t = (x_2, y_2)$. These displacements are referred to local coordinates located at each vertex. The departure points are connected by a straight line that intersects the west and east faces at local coordinates y_a and y_b, respectively, and the north face at local coordinates x_a and x_b, as indicated in Fig. 2. The intersection points are

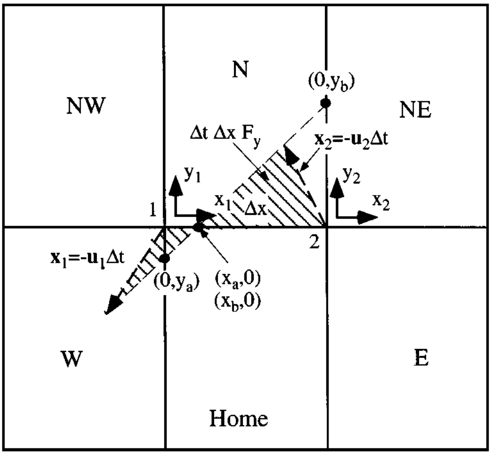


FIG. 2. The fluxing area on the north face of the arrival cell, and the associated notation. Note that the points $(x_a, 0)$ and $(x_b, 0)$ refer to exactly the same point, but with respect to different coordinates.

given by

$$\begin{aligned} y_a &= \frac{y_1\Delta x + (x_2y_1 - x_1y_2)}{\Delta x + (x_2 - x_1)}, & y_b &= \frac{y_2\Delta x + (x_2y_1 - x_1y_2)}{\Delta x + (x_2 - x_1)}, \\ x_a &= \frac{y_a\Delta x}{y_a - y_b}, & x_b &= \frac{y_b\Delta x}{y_a - y_b}; \ (x_a - x_b = \Delta x). \end{aligned} \tag{18}$$

Depending on the direction of the velocity at a given vertex, the corresponding trajectory departure point may be located in any one of the four neighboring cells surrounding the vertex. There are very many possibilities associated with the different orientations of the independent velocities at the two vertices. These possibilities may be summarized by dividing the fluxing area into four components:

$$\Delta t\Delta x F_y = F_{y1} + F_{y2} + F_{y3} + F_{y4}. \tag{19}$$

These components are classified largely according to the cells in which they are located, as illustrated in Fig. 3. Thus, for example, the fluxing area in Fig. 2 is composed of contributions from two components, F_{y1} and F_{y3}; F_{y1} originates from the W cell, and F_{y3} originates from the H and N cells. The remaining two components make no contribution. The four types of components are defined in the following. They exhaust all possibilities.

(a) F_{y1}: Corner contributions (E, W, NE, and NW cells) [Case: $y_ay_1 > 0$, $x_1 < 0$ and/or $y_by_2 > 0$, $x_2 > 0$; see Fig. 3a].

$$\begin{aligned} F_{y1} = {} & \frac{1}{4}(y_a - |y_a|)x_1\langle\phi\rangle_W - \frac{1}{4}(y_b - |y_b|)x_2\langle\phi\rangle_E \\ & - \frac{1}{4}(y_b + |y_b|)x_2\langle\phi\rangle_{NE} + \frac{1}{4}(y_a + |y_a|)x_1\langle\phi\rangle_{NW}, \end{aligned} \tag{20}$$

where $\langle\phi\rangle_W$ represents the area average of ϕ in the triangle in the W cell, for example. These averages are obtained from triangle area integrals of reconstructed ϕ distributions, as described in the next section. There are four possible triangles that can contribute, but at most two can contribute at any one time.

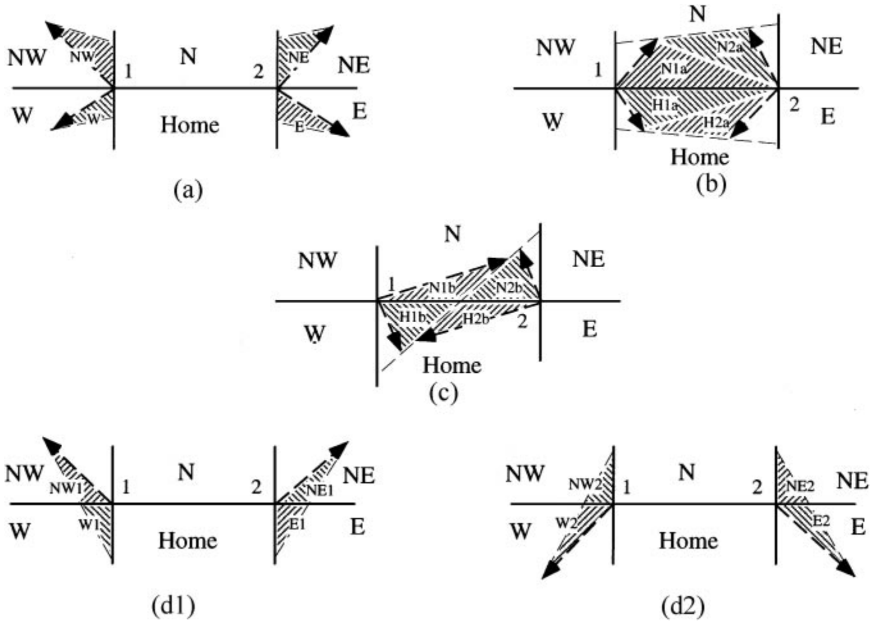


FIG. 3. The four possible contributions to the fluxing area on the north face of the arrival cell: (a) F_{y1}, (b) F_{y2}, (c) F_{y3}, and (d) F_{y4}.

(b) F_{y2}: Direct contributions (H and N cells) [Case: $y_a y_b \geq 0$, $x_a x_b \geq 0$; see Fig. 3b].

$$F_{y2} = -\frac{1}{4}(y_b + |y_b|)(\Delta x + \tilde{x}_2 - \tilde{x}_1)\langle\phi\rangle_{N2} - \frac{1}{4}(\tilde{y}_1 + |\tilde{y}_1|)\Delta x\langle\phi\rangle_{N1}$$
$$-\frac{1}{4}(y_b - |y_b|)(\Delta x + \tilde{x}_2 - \tilde{x}_1)\langle\phi\rangle_{H2} - \frac{1}{4}(\tilde{y}_1 - |\tilde{y}_1|)\Delta x\langle\phi\rangle_{H1}, \qquad (21)$$

where

$$\text{if } x_1 > 0 \ \{\tilde{x}_1 = x_1, \tilde{y}_1 = y_1\}, \qquad \text{else } \{\tilde{x}_1 = 0, \tilde{y}_1 = y_a\}$$
$$\text{if } x_2 > 0 \ \{\tilde{x}_2 = x_2, \tilde{y}_2 = y_2\}, \qquad \text{else } \{\tilde{x}_2 = 0, \tilde{y}_2 = y_b\}.$$

The first two terms in (21) represent the contribution from the quadrilateral possibly located in the N cell, divided into two triangles labeled N1a and N2a, and the remaining terms are from triangles labeled H1a and H2a from the quadrilateral possibly located in the H cell. Again, it is obvious that at most two of these four possible triangles can contribute at any one time.

(c) F_{y3}: Direct contributions (H and N cells) [Case: $y_a y_b < 0$, $x_a x_b < 0$; see Fig. 3c].

$$F_{y3} = -\frac{1}{4}(\tilde{y}_1 - |\tilde{y}_1|)x_a\langle\phi\rangle_{H1} - \frac{1}{4}(\tilde{y}_1 + |\tilde{y}_1|)x_a\langle\phi\rangle_{N1}$$
$$+\frac{1}{4}(\tilde{y}_2 - |\tilde{y}_2|)x_b\langle\phi\rangle_{H2} + \frac{1}{4}(\tilde{y}_2 + |\tilde{y}_2|)x_b\langle\phi\rangle_{N2}. \qquad (22)$$

These are contributions from the four possible triangles labeled N1b, N2b, H1b, and H2b in Fig. 3c. Again, at most two of these four possible triangles can contribute at any one time.

(d) F_{y4}: Corner contributions (E, W, NE, and NW cells) [Case: $y_a y_1 \le 0$, $x_1 < 0$ and/or $y_b y_2 \le 0$, $x_2 > 0$; see Figs. 3d1 and 3d2].

$$F_{y4} = \frac{1}{8}(x_a - |x_a|)\begin{bmatrix}(y_a - |y_a|)\langle\phi\rangle_{W1} - (y_1 - |y_1|)\langle\phi\rangle_{W2} \\ \quad + (y_a + |y_a|)\langle\phi\rangle_{NW2} - (y_1 + |y_1|)\langle\phi\rangle_{NW1}\end{bmatrix} - \frac{1}{8}(x_b + |x_b|)\begin{bmatrix}(y_b + |y_b|)\langle\phi\rangle_{NE2} - (y_2 + |y_2|)\langle\phi\rangle_{NE1} \\ \quad + (y_b - |y_b|)\langle\phi\rangle_{E1} - (y_2 - |y_2|)\langle\phi\rangle_{E2}\end{bmatrix}. \tag{23}$$

These summarize the possible contributions from the eight triangles shown in Figs. 3d1 and 3d2, but it is easy to see that at most two of these triangles can contribute at any one time.

The calculation of all these contributions may be organized in various alternative ways for computational convenience and efficiency. Many cases are mutually exclusive and careful programming should take this into account. Table I lists the 20 possible triangles associated with the N face of the H cell, and indicates the 8 triangle evaluations that might actually be required in the code implementing the present algorithm. For example, the logical conditions eliminate all but 1 of the 3 triangles that may possibly be located in the NW cell. Of course,

TABLE I
Evaluation of Contributions from the Triangles Shown in Fig. 3

Triangle evaluated	Cell label	Triangle label	Selecting logical condition
1	NW	NW	$(y_a > 0$ and $y_1 > 0)$ and $x_1 < 0$
		NW2	$(y_a > 0$ and $y_1 < 0)$ and $x_1 < 0$
		NW1	$(y_a < 0$ and $y_1 > 0)$ and $x_1 < 0$
2	W	W	$(y_a < 0$ and $y_1 < 0)$ and $x_1 < 0$
		W1	$(y_a < 0$ and $y_1 > 0)$ and $x_1 < 0$
		W2	$(y_a > 0$ and $y_1 < 0)$ and $x_1 < 0$
3	NE	NE	$(y_b > 0$ and $y_2 > 0)$ and $x_2 > 0$
		NE2	$(y_b > 0$ and $y_2 < 0)$ and $x_2 > 0$
		NE1	$(y_b < 0$ and $y_2 > 0)$ and $x_2 > 0$
4	E	E	$(y_b < 0$ and $y_2 < 0)$ and $x_2 > 0$
		E1	$(y_b < 0$ and $y_2 > 0)$ and $x_2 > 0$
		E2	$(y_b \ge 0$ and $y_2 < 0)$ and $x_2 > 0$
5	H	H1a	$y_a y_b \ge 0$ and $y_a + y_b < 0$
		H1b	$y_a y_b < 0$ and $\tilde{y}_1 < 0$
6	H	H2a	$y_a y_b \ge 0$ and $y_a + y_b < 0$
		H2b	$y_a y_b < 0$ and $\tilde{y}_2 < 0$
7	N	N1a	$y_a y_b \ge 0$ and $y_a + y_b > 0$
		N1b	$y_a y_b < 0$ and $\tilde{y}_1 > 0$
8	N	N2a	$y_a y_b \ge 0$ and $y_a + y_b > 0$
		N2b	$y_a y_b < 0$ and $\tilde{y}_2 > 0$

Note. Triangle evaluation in the code implementing the present algorithm is organized according to the cell in which each triangle is located. Many of the triangles are mutually exclusive. Therefore, only 8 of the possible 20 triangles need to be evaluated, as selected by the above logical conditions.

the average number of triangle evaluations per cell face is much smaller and depends on the velocity field; i.e., a uniform velocity field, not aligned with the grid, would require only 3 triangle evaluations per face.

2.2. *Time Step*

The time step is determined by accuracy and consistency considerations, and not by stability considerations. It is required that (a) backward trajectories do not extend beyond the nearest neighbor cells, (b) the trajectories do not cross, and (c) the departure cell is not inverted and has positive area. These conditions may be too complicated to implement in general. A more restrictive but much simpler condition that satisfies these requirements is

$$|u|\Delta t/\Delta x \leq 1/2, \qquad |v|\Delta t/\Delta y \leq 1/2 \tag{24}$$

at each vertex.

2.3. *Reconstruction*

We observe that ϕ may be either ρ or ρT. For a second-order-accurate remapping, both ρ and T must be linear functions of position reconstructed from known mean values $\bar{\rho}$, $\tilde{T}$ on the grid [8]. Based on conservation considerations, we require that integrals of density and the product of density and tracer over the cell recover the total mass and the total tracer:

$$\begin{aligned} \int_A \rho\, dA &= \bar{\rho} A, \\ \int_A \rho T\, dA &= \bar{\rho}\tilde{T} A. \end{aligned} \tag{25}$$

This implies that the distributions in each cell must be

$$\begin{aligned} \rho(\mathbf{r}) &= \bar{\rho} + \alpha\langle\nabla\rho\rangle \cdot (\mathbf{r} - \bar{\mathbf{r}}), \\ T(\mathbf{r}) &= \tilde{T} + \beta\langle\nabla T\rangle \cdot (\mathbf{r} - \tilde{\mathbf{r}}), \end{aligned} \tag{26}$$

where $\mathbf{r}$ is the position vector, $\bar{\mathbf{r}} = \int_A \mathbf{r}\, dA / \int_A dA$ is the cell centroid, $\tilde{\mathbf{r}} = \int_A \rho\mathbf{r}\, dA / \int_A \rho\, dA$ is the cell center of mass, $\langle\nabla\rho\rangle$, $\langle\nabla T\rangle$ are centered estimates of the gradients of density and tracer within the cell, respectively, and $\alpha, \beta (0 \leq \alpha, \beta \leq 1)$ are limiting coefficients that enforce monotonicity. The gradients may be any centered estimates; this results in second-order accuracy when $\alpha, \beta = 1$ (on a uniform orthogonal grid, as in the examples in Section 3, these are simply divided mean differences along grid lines, i.e., $\langle\rho\rangle_x = (\rho_{i+1} - \rho_{i-1})/(x_{i+1} - x_{i-1})$). Monotonicity is ensured by a two-dimensional adaptation [8] of van Leer limiting [17],

$$\alpha = \mathrm{Min}[1, \alpha_{\min}, \alpha_{\max}], \tag{27}$$

where

$$\begin{aligned} \alpha_{\min} &= \mathrm{Max}[0, (\bar{\rho}_{\min} - \bar{\rho})/(\rho_{\min} - \bar{\rho})], \\ \alpha_{\max} &= \mathrm{Max}[0, (\bar{\rho}_{\max} - \bar{\rho})/(\rho_{\max} - \bar{\rho})], \end{aligned}$$

and $\bar{\rho}_{\min}, \bar{\rho}_{\max}$ are minimum and maximum values of $\bar{\rho}$ among the nearest neighbors; $\rho_{\min}, \rho_{\max}$ are minimum and maximum values of ρ within the cell, calculated from (26) with $\alpha = 1$ (because the distribution is linear, these are among the four values at the vertices). The limiting coefficient β is obtained analogously from the $\tilde{T}$ distribution on the mesh.

The distributions in (26) are linear and the product ρT is quadratic; therefore we need to integrate quadratic functions. The Appendix gives a simple three-point integration formula that is exact for quadratics on arbitrary triangles. All the flux contributions may be calculated using this rule. Since the coefficients of the tracer distribution are constant within a cell, only integrals of $\phi = \{\rho, \rho x, \rho y\}$ over each triangle are needed and they may be reused for each tracer. Furthermore, integrals of $\phi = \{x, y, x^2, xy, y^2\}$ over each cell are required to obtain cell centroids and centers of mass; these are purely geometrical quantities that may be precalculated and stored. The integration of quadratic functions may be avoided but at a sacrifice of accuracy and/or compatibility. In this case compatibility may be preserved by evaluating the tracer integral in (25) using a constant density distribution, but with resulting loss of accuracy.

3. EXAMPLE CALCULATION

We carry out a simple test problem on a square domain of size 128 × 128 units, subdivided into a uniform grid of 128 × 128 cells. The velocity field corresponds to uniform clockwise angular rotation about the center of the domain such that the velocity is unity at the midpoints of the sides. The time step is $\Delta t = 0.65$. This exceeds the suggestion (24) for the time step but there is no danger of trajectory crossing because the velocity field is regular. The grid vertex displacements are calculated exactly in order to avoid spurious divergence errors. The computation is continued for 621 time steps, which is approximately a full period of revolution.

We consider two different density and tracer distributions. In the first experiment, the density and the tracer are given by a cylindrical distribution of unit amplitude and 20 units in diameter, centered 42 units directly north from the center of rotation, and surrounded by an ambient value of 0.1 for the density and zero for the tracer. In other words, the density and tracer distributions have the same shape and are superimposed over one another. This is perhaps the most demanding situation for the purpose of checking the compatibility properties of the algorithm. The results are shown in Fig. 4 for the density and Fig. 5 for the tracer. We observe that both the density and tracer distributions are free from oscillations of any kind. Both distributions retain unit amplitude over the duration of the experiment. The density very quickly acquires a slightly diffused outline and it propagates almost unchanged thereafter. This is as expected for a second-order remapping algorithm [8]. However, the corresponding result for the tracer is quite remarkable. We note that initially the tracer has values of either zero or one. Therefore, if any tracer diffuses out of the central distribution, no matter how minuscule an amount, it carries along a unit value. This is why, if we were to set the external density to zero, we would observe a large spreading patch of unit tracer value as the cylinder propagates around. The nonzero value of density outside the cylinder reduces this tendency because the admixture of a tiny amount of nonzero tracer is diluted over a large amount of material. Nevertheless, the diameter of the tracer cylinder appears to grow because of this effect, as the tracer is diffused out of the initial cylindrical distribution. Therefore, these effects are all as one would expect and compatibility is perfectly obeyed. Of course, total mass and total tracer are conserved by construction.

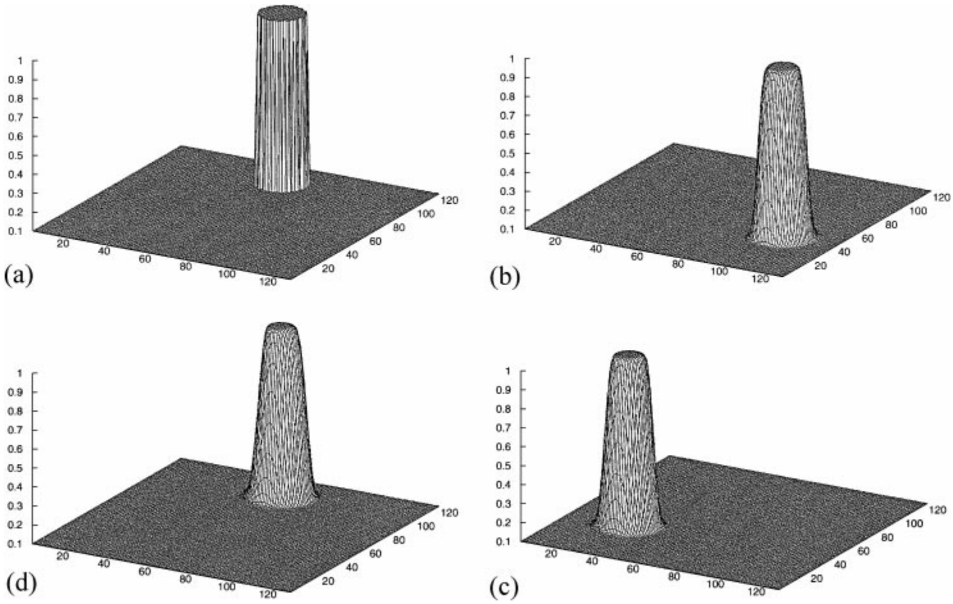


FIG. 4. The propagation of an initially cylindrical density distribution, advected with a uniformly rotating velocity field (n is the time step number): (a) initial distribution, $n = 0$; (b) one-third revolution, $n = 207$; (c) two-thirds revolution, $n = 414$; (d) full revolution, $n = 621$.

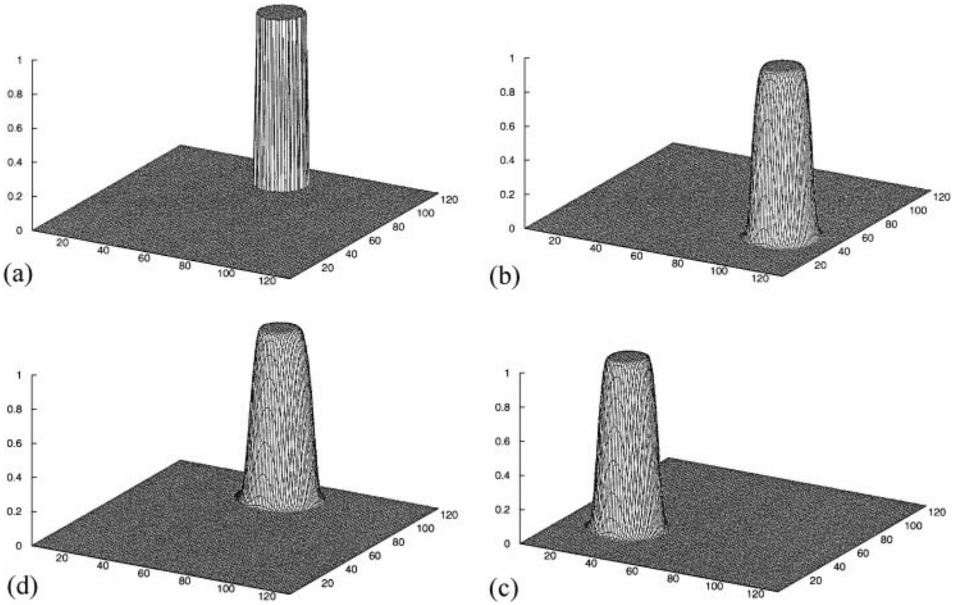


FIG. 5. The propagation of an initially cylindrical tracer distribution, exactly superimposed on the density distribution of Fig. 4, and advected with a uniformly rotating velocity field: (a) initial distribution, $n = 0$; (b) one-third revolution, $n = 207$; (c) two-thirds revolution, $n = 414$; (d) full revolution, $n = 621$.

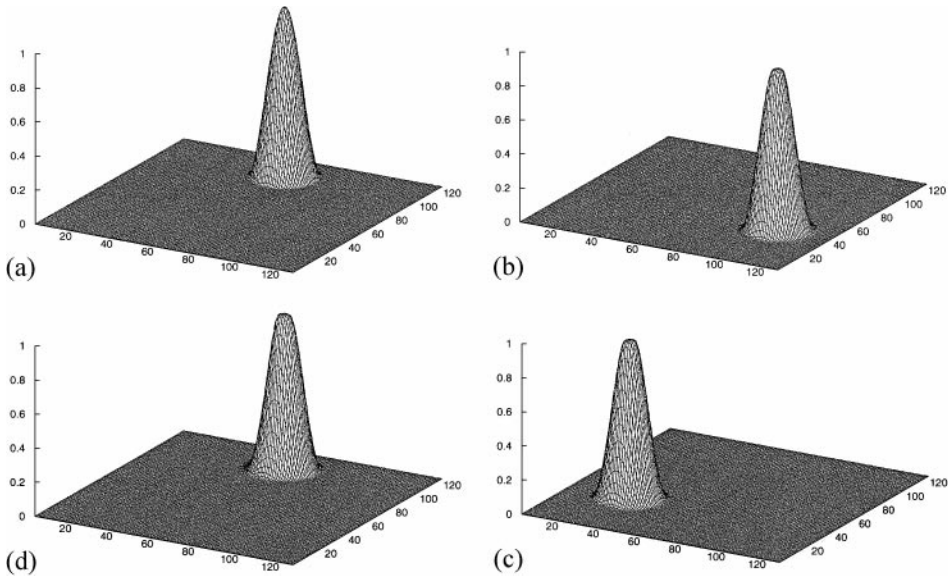


FIG. 6. The propagation of an initial cosine-bell density distribution, advected with a uniformly rotating velocity field: (a) initial distribution, $n = 0$; (b) one-third revolution, $n = 207$; (c) two-thirds revolution, $n = 414$; (d) full revolution, $n = 621$.

In the second experiment, the density and the tracer are initially zero everywhere except for a cosine-bell distribution of unit amplitude, 16 units in diameter halfway up, and 32 units in diameter at the base, and centered 42 units directly north from the center of rotation. In other words, it is a similar experiment except that the distribution is much better resolved by the mesh. The results are shown in Fig. 6 for the density and Fig. 7 for the tracer. We

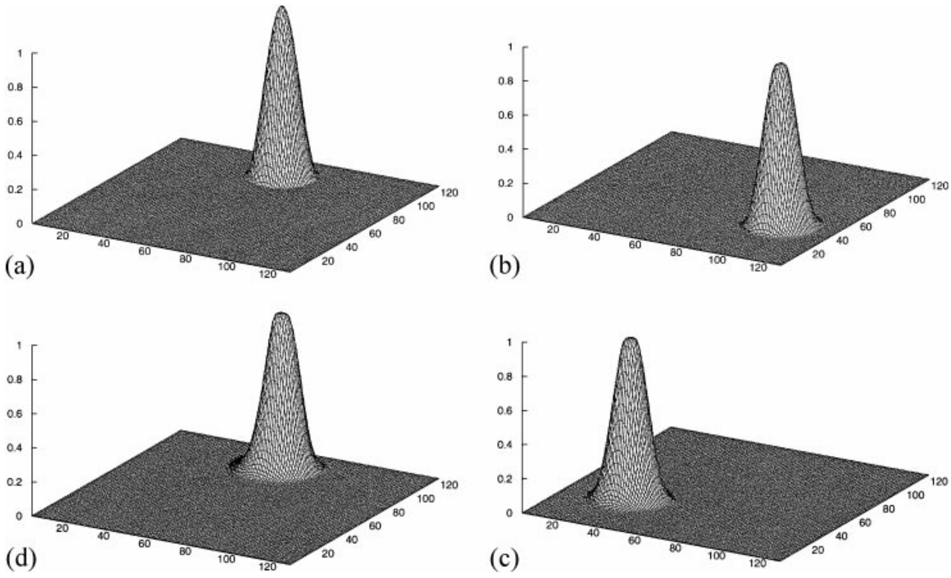


FIG. 7. The propagation of an initial cosine-bell tracer distribution, exactly superimposed on the density distribution of Fig. 6, and advected with a uniformly rotating velocity field: (a) initial distribution, $n = 0$; (b) one-third revolution, $n = 207$; (c) two-thirds revolution, $n = 414$; (d) full revolution, $n = 621$.

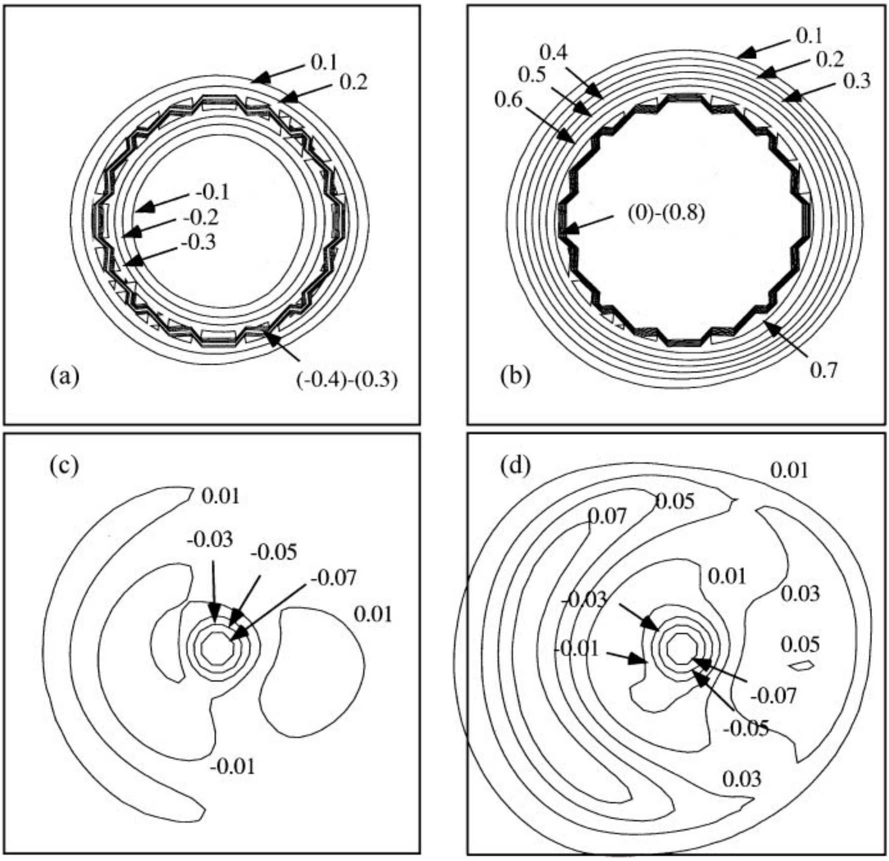


FIG. 8. Error contours inside a window, $43 \leq x \leq 81$, $87 \leq y \leq 125$, centered about the mean location of the distributions at $n = 621$: (a) density, cylindrical initial distribution; (b) tracer, cylindrical initial distribution; (c) density, cosine-bell initial distribution; (d) tracer, cosine-bell initial distribution.

observe that the cosine-bell distribution is propagated with little change. The amplitude of the density is reduced to 0.909 and the maximum tracer value is reduced to 0.910. We also observe that the tracer distribution is slightly broadened at the base, in accordance with the previous discussion.

Since the initial distributions should be propagated without change, it is relatively easy to compute the error. Figure 8 shows contour plots of the error for the two experiments at the final time. The error for the cylindrical distribution is quite large, as is to be expected for a discontinuous distribution, but it is confined to a relatively narrow region. What is remarkable is that the density and tracer distributions have preserved their symmetry. The error for the cosine-bell distribution is an order of magnitude smaller. The density error is largely concentrated at the peak of the distribution, as is generally observed for monotonicity-preserving schemes, which tend to suffer from peak clipping. The tracer error is also small, although the tracer distribution is broadened somewhat in the wake due to the tracer diffusion effect described above.

Timing is machine and problem dependent. However, a rough estimate of the cost yields $660 + 75 * \text{NT}$ floating point operations/cell, where NT is the number of tracers, for the most common case of three triangle evaluations per face. This is approximately consistent with an actual measurement on a single processor of an SGI Origin 2000 that showed that it takes between seven and eight tracers to double the execution time of the single tracer experiments described above. This emphasizes the low cost of additional tracers in this algorithm. Of course, the relative cost will be dependent on how efficiently the geometric and tracer dependent parts of the algorithm are coded.

4. DISCUSSION

A new method has been described for the solution of transport equations that combines the best features of a discrete flux-based scheme and an integral remapping. The resulting method is simpler and cheaper than a general remapping, and retains all its advantageous properties except for the ability to take an unrestricted time step. Notable among these is the compatibility property, which allows the method to deal with tracers accurately. This is particularly important in applications where the variable corresponding to density undergoes large variations and may even vanish. Examples include phase density in the case of multiphase fluid models, ice compactness in the case of sea ice models, and layer thickness in the case of isopycnal ocean models. Unless compatibility is assured, the associated tracer, such as temperature or salinity, may become negative or may assume some other unphysical value, and the code may fail as a result.

Many applications involve a large number of tracers. Examples include models of the transport of atmospheric pollutants with a very large number of constituents, and sea ice models that contain a large number of sea ice categories. The present method is advantageous in these cases because the incremental cost of adding each additional tracer is small.

Parenthetically, the present method improves on the multidimensional accuracy of existing methods. Many multidimensional monotone advection or transport schemes make use of a linear combination of a low-order monotone scheme and a high-order antidiffusive scheme. The low-order scheme is typically an upwind or donor-cell scheme. Unfortunately, standard extensions of the one-dimensional donor-cell scheme are not truly multidimensional. In 1D, the donor cell scheme may be interpreted as the remapping of cell-constant quantities. This is not the case in multi-dimensional versions of the donor-cell algorithm because the associated truncation error terms are not tensor invariant (e.g., see [18]); in other words, they lack the so-called “corner-coupling” terms. This error is masked in practice by the very large diffusion associated with the donor-cell algorithm. The low-order part of the present algorithm, i.e., that part associated with the leading constant terms on the right-hand side of the reconstruction equations (26), specifically includes contributions from corner cells and is therefore expected to be fully tensor invariant. This means that the present algorithm, or at least the low-order part of it, is at last the proper generalization to multiple dimensions of the donor-cell algorithm.

Finally, the present method has a rather obvious extension to three dimensions, although at a large increase in complexity due to a greatly increased number of cases arising from the subdivision of fluxing volumes into constituent tetrahedra.

APPENDIX: AREA INTEGRALS OVER ARBITRARY TRIANGLES

In a Cartesian plane, $\mathbf{r} = (x, y)$, consider an arbitrary triangle $\mathcal{T}$ defined by vertices $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$. We wish to evaluate the area integral

$$I = \int_{\mathcal{T}} f(\mathbf{r})\, dx\, dy$$

over the triangle $\mathcal{T}$, where $f(\mathbf{r})$ is an arbitrary function of position. Then

$$I = Af(\bar{\mathbf{r}}); \qquad \bar{\mathbf{r}} = \frac{1}{3}(\mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_3), \tag{A1}$$

is exact for linear functions, and

$$\begin{gathered} I = \frac{1}{3}A[f(\mathbf{r}_a) + f(\mathbf{r}_b) + f(\mathbf{r}_c)]; \\ \mathbf{r}_a = \frac{1}{6}(4\mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_3), \quad \mathbf{r}_b = \frac{1}{6}(\mathbf{r}_1 + 4\mathbf{r}_2 + \mathbf{r}_3), \quad \mathbf{r}_c = \frac{1}{6}(\mathbf{r}_1 + \mathbf{r}_2 + 4\mathbf{r}_3), \end{gathered} \tag{A2}$$

is exact for quadratic functions, where A is the triangle area given by

$$A = \frac{1}{2}|(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)|.$$

ACKNOWLEDGMENT

This work was made possible by the support of the DOE CCPP (Climate Change Prediction Program) program.

REFERENCES

1. R. Peyret and T. D. Taylor, *Computational Methods for Fluid flow* (Springer-Verlag, New York, 1983).
2. R. J. LeVeque, *Numerical Methods for Conservation Laws* (Birkhauser Verlag, Basel, 1992).
3. P. J. Roache, *Fundamentals of Computational Fluid Dynamics* (Hermosa, Albuquerque, 1998).
4. C. Schär and P. K. Smolarkiewicz, A synchronous and iterative flux-correction formalism for coupled transport, *J. Comput. Phys.* **128**, 101 (1996).
5. W. B. VanderHeyden and B. A. Kashiwa, Compatible fluxes for van Leer advection, *J. Comput. Phys.* **146**, 1 (1998).
6. P. A. Thompson, *Compressible-Fluid dynamics* (McGraw–Hill, New York, 1972), p. 16.
7. P. K. Smolarkiewicz and J. A. Pudykiewicz, A class of semi-lagrangian approximations for fluids, *J. Atmos. Sci.* **49**, 2082 (1992).
8. J. K. Dukowicz and J. W. Kodis, Accurate conservative remapping (rezoning) for arbitrary Lagrangian–Eulerian computations, *SIAM J. Statist. Comput.* **8**, 305 (1987).
9. D. S. Miller, D. E. Burton, and J. S. Oliviera, *Efficient Second Order Remapping on Arbitrary Two Dimensional Meshes*, UCRL-ID-123530, Lawrence Livermore National Laboratory, Livermore, CA, 1996.
10. J. Grandy, Conservative remapping and region overlays by intersecting arbitrary polyhedra, *J. Comput. Phys.* **148**, 433 (1999).
11. J. K. Dukowicz and N. T. Padial, *REMAP3D: A Conservative Three-Dimensional Remapping Code*, LA-12136-MS, Los Alamos National Laboratory, Los Alamos, NM, 1991.
12. S. J. Mosso, D. E. Burton, A. K. Harrison, E. J. Linnebur, and B. K. Swartz, *A Second Order, Two- and Three-Dimensional Remap Method*, LA-UR-98-5353, Los Alamos National Laboratory, Los Alamos, NM, 1998.
13. L. M. Leslie and R. J. Purser, Three-dimensional mass-conserving semi-Lagrangian scheme employing forward trajectories, *Mon. Wea. Rev.* **123**, 2551 (1995).
14. B. P. Leonard, A. P. Lock, and M. K. MacVean, Conservative explicit unrestricted-time-step multidimensional constancy preserving advection schemes, *Mon. Wea. Rev.* **124**, 2588 (1996).

15. A. Staniforth and J. Côté, Semi-Lagrangian integration schemes for atmospheric models-A review, *Mon. Wea. Rev.* **119**, 2206 (1991).

16. C. W. Hirt, A. A. Amsden, and J. L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, *J. Comput. Phys.* **14**, 227 (1974).

17. B. Van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, *J. Comput. Phys.* **32**, 101 (1979).

18. J. K. Dukowicz and J. D. Ramshaw, Tensor viscosity method for convection in numerical fluid dynamics, *J. Comput. Phys.* **32**, 71 (1979).