A Correction Method for Use in Multidimensional Time-Splitting Advection Algorithms: Application to Two- and Three-Dimensional Transport

A. CLAPPIER

Laboratoire de Pollution de l'Air et des Sols, Swiss Federal Institute of Technology, Lausanne, Switzerland

(Manuscript received 3 September 1996, in final form 24 April 1997)

ABSTRACT

Meteorological and air quality models rely on accurately solving the advection equation in two and three dimensions. While a number of methods have been developed, all suffer from the formation and growth of errors during the solution procedure. Here, a correction method is developed and applied to the piecewise parabolic method for use in multidimensional modeling. This method is a time-split, alternating direction method with a flux correction to account for diagonal advection. The correction removes over- and undershooting while maintaining the method's accuracy. The analysis also indicates that some methods will have errors that grow significantly in time, while the corrections developed minimize the problem. This analysis found that the buildup of errors was more pronounced in three-dimensional tests, suggesting that this is an import evaluation criteria for other advection algorithms as well.

1. Introduction

A variety of numerical algorithms have been developed to solve the transport equation in Eulerian grid models. One of the popular classes of algorithms are those based on using finite volume discretization successively, for example, Crowley (1968), Colella and Woodward (1984), Bott (1992), and Yamartino (1993). These schemes are used to calculate the advection of a state variable in one dimension, and for the last three of these schemes the authors employed a flux limiter to avoid the formation of local extrema (over- and undershooting). For multidimensional cases, these schemes are commonly used with the time-splitting algorithm described by Marchuk (1975). This method gives good results for the well-known test of the cone in a rotating wind field (Crowley 1968). However, for some other tests, when the monodimensional divergence of the flow field is not zero ($\partial u/\partial x \neq 0$, $\partial v/\partial y \neq 0$, or $\partial w/\partial z \neq 0$), the time-splitting algorithm creates over- and undershooting errors that increase with time (Easter 1993). For this reason, Bott (1993) added a correction term to his scheme. For the test of Staniforth et al. (1987) he found that undershooting did not increase with time but reached a maximum of 28% before decreasing to 5%. While a number of reviews have compared different schemes (Chock 1991; Chock and Winkler 1994; Odman and Russell 1991a,b; Odman et al. 1996), this article is intended to analyze the reasons for specific types of errors and develop natural correction methods to improve their accuracy with minimal computational penalty. This paper presents a correction method applied here to the piecewise parabolic method (PPM) (Colella and Woodward 1984). The correction method also avoids the over- and undershooting previously produced by the time-splitting approach.

Corresponding author address: Dr. Alain Clappier, Laboratoire de Pollution de l'Air et des Sols, Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland.
E-mail: alain.clappier@dgr.epfl.ch

2. Description of the 1D scheme: PPM

The four explicit schemes of Crowley (1968), Colella and Woodward (1984), Bott (1992), and Yamartino (1993) solve the transport equation in one dimension:

$$\frac{\partial a}{\partial t} + \frac{\partial ua}{\partial x} = 0. \tag{2.1}$$

For each of these schemes, the tracer a is defined as an average over each computational grid cell. The velocity u is defined at the faces between the different grid cells; for example, u_i is the velocity at the i face between cell $I - 1$ and I (Fig. 1). The average value of a at time $n + 1$ in the cell I is calculated as a function of a_I^n, the value of a at time n:

$$a_I^{n+1} = a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x) \tag{2.2}$$

with

$$\Delta(u\overline{a}^x) = u_{i+1}\overline{a}^x_{i+1} - u_i\overline{a}^x_i,$$

where $(\Delta t/\Delta x)u_i\overline{a}^x_i$ is the flux of a at the face i and $\overline{a}^x_i$

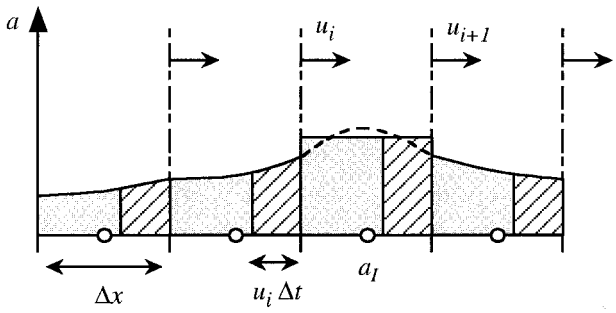

FIG. 1. Principle of the PPM algorithm. Piecewise parabolic interpolants, $a(x)$, are fit to conserve mass and give a continuous function in one dimension. If the fit leads to an internal maximum or minimum, it is replaced in that interval by a constant value. The solid line is the function $a(x)$ as used in the method. The dashed line is the polynomial fit used before correcting for the local maxima found in that interval.

represents the average value of a that crosses the face i during the time step Δt:

$$\overline{a}_i^x = \frac{1}{u_i \Delta t} \int_{-u_i \Delta t}^{0} a(x)\, dx, \tag{2.3}$$

where $a(x)$ is a polynomial interpolation of a over the grid. Note that the flux depends only on $a(x)$ in the upwind cell. In general, the overbar refers to averaging of the quantity over the domain that would cross the face. In general, the subscript i refers to the quantity evaluated at the i face.

The interpolation of a high-order polynomial can produce local extrema that can lead to over- and undershooting. The main differences between the schemes of Crowley (1968), Colella and Woodward (1984), Bott (1992), and Yamartino (1993) are the interpolation function they use and the method they employ to avoid local maxima and oscillations. In the PPM, Colella and Woodward interpolate the value for tracer a with continuous parabolic pieces. The parabola is replaced by a constant value or linear interpolation if it is found that a local maximum or minimum results from the parabolic fit (Fig. 1). This technique is explained in detail by Carpenter et al. (1990).

3. Analysis of errors made by advection algorithms in two directions

The advection equation in two dimensions is

$$\frac{\partial a}{\partial t} + \frac{\partial ua}{\partial x} + \frac{\partial va}{\partial y} = 0. \tag{3.1}$$

This can be solved by developing a method that accounts for advection in both directions in one step or, as is commonly done, by multiple application of the monodimensional algorithm in the different directions. Repeated application of the monodimensional algorithms results in the possibility that the equation can be solved in a variety of ways. As an example of a single-step algorithm, a first method consists of calculating the derivative in two directions and adding the two terms in order to calculate the value of the tracer at time $n + 1$.

Method 1:

$$a_I^{n+1} = a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x) - \frac{\Delta t}{\Delta x}\Delta(v\overline{a}^y). \tag{3.2}$$

This is a direct extension of PPM to two dimensions.

A second method, operator splitting, described by Marchuk (1975), consists of calculating a first value for the tracer, taking into account only the derivative in one direction. This new value is used to calculate the derivative in the second direction.

Method 2:

$$a_I' = a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x)$$

$$a_I^{n+1} = a_I' - \frac{\Delta t}{\Delta y}\Delta(v\overline{a'}^y). \tag{3.3}$$

These two methods give different results, as is seen in two simple tests. In the first test, a is initialized as a

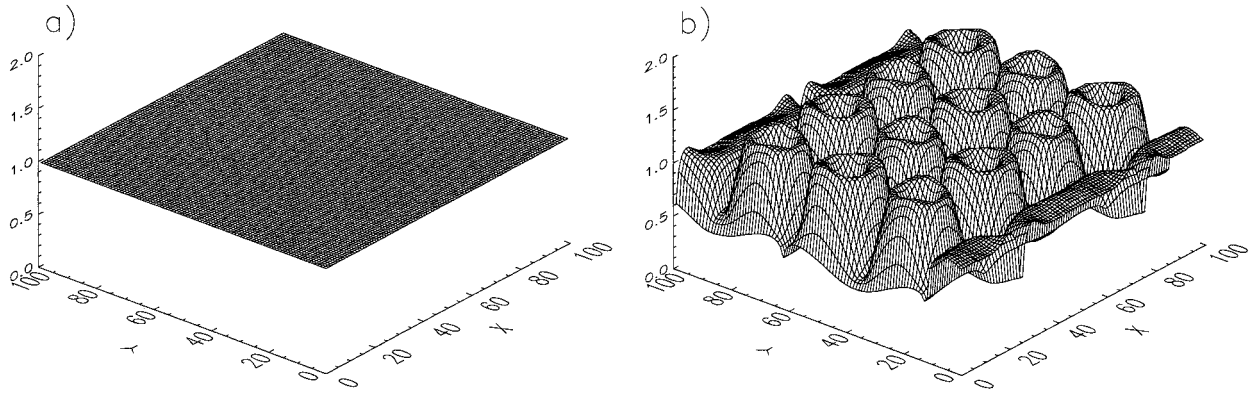

FIG. 2. Calculated distribution of a after 100 time steps of 1 s (Courant number is 1): (a) method 1 and (b) method 2.

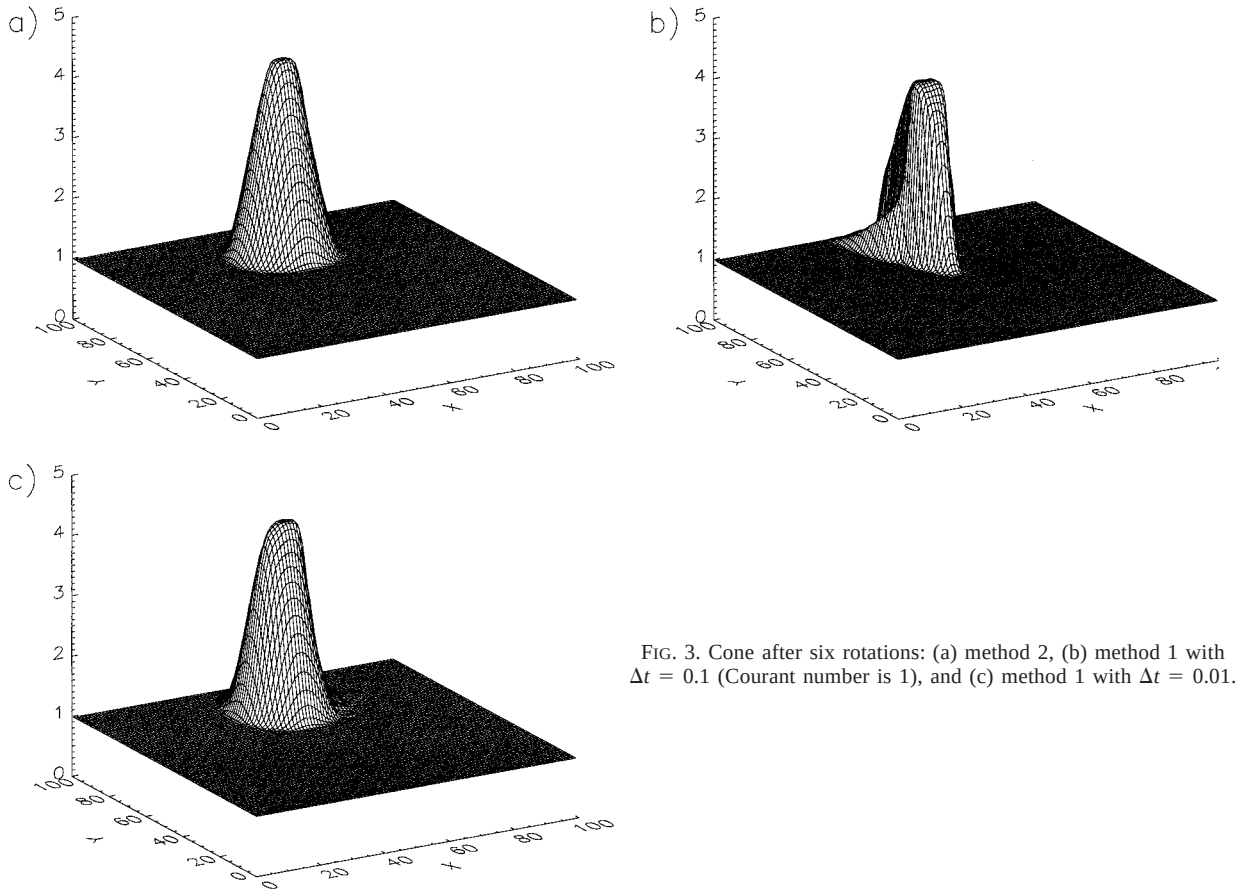

FIG. 3. Cone after six rotations: (a) method 2, (b) method 1 with $\Delta t = 0.1$ (Courant number is 1), and (c) method 1 with $\Delta t = 0.01$.

constant value and the flow field is the strong deformational flow field of the Staniforth et al. (1987) test:

$$u = \frac{8\pi}{25}\sin\left(\frac{\pi x}{25}\right)\sin\left(\frac{\pi y}{25}\right)$$

$$v = \frac{8\pi}{25}\cos\left(\frac{\pi x}{25}\right)\cos\left(\frac{\pi y}{25}\right).$$

The solution of this test is easy to find: a should stay constant in time and in space. Method 1 gives the exact solution, as is seen below: If a is constant, $\overline{a}^x = \overline{a}^y = a$, from (3.2) we obtain

$$a_I^{n+1} = a\left(1 - \frac{\Delta t}{\Delta x}\Delta u - \frac{\Delta t}{\Delta y}\Delta v\right) \tag{3.4}$$

due to the mass conservation of the wind field,

$$\frac{\Delta t}{\Delta x}\Delta u - \frac{\Delta t}{\Delta y}\Delta v = 0,$$

and therefore $a_I^{n+1} = a$.

Method 2 produces a gradient in a that increases with the simulation time (Fig. 2b). This result can be explained starting from Eq. (3.3):

$$a_I^{n+1} = a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x) - \frac{\Delta t}{\Delta y}\Delta\left[v\overline{\left(a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x)\right)}^y\right]. \tag{3.5}$$

The last term of Eq. (3.5) can be developed as follows:

$$a_I^{n+1} = a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x) - \frac{\Delta t}{\Delta y}\Delta(v\overline{a}^y) + \frac{\Delta t^2}{\Delta y \Delta x}\Delta\left[v\Delta\left(u\overline{\overline{a}^x}^y\right)\right]. \tag{3.6}$$

If $\overline{a}^x = \overline{a}^y = a$, (3.6) gives

$$a_I^{n+1} = a\left(1 - \frac{\Delta t}{\Delta x}\Delta u - \frac{\Delta t}{\Delta y}\Delta v\right) + a\frac{\Delta t^2}{\Delta y \Delta x}\Delta(v\Delta u). \tag{3.7}$$

Equations (3.4) and (3.7) show that method 2 generates a supplemental term:

$$a\frac{\Delta t^2}{\Delta y \Delta x}\Delta(v\Delta u). \tag{3.8}$$

This term is zero if the flow field is monodimensionally divergence free (i.e., $\partial u/\partial x = \partial v/\partial y = 0$). If the flow

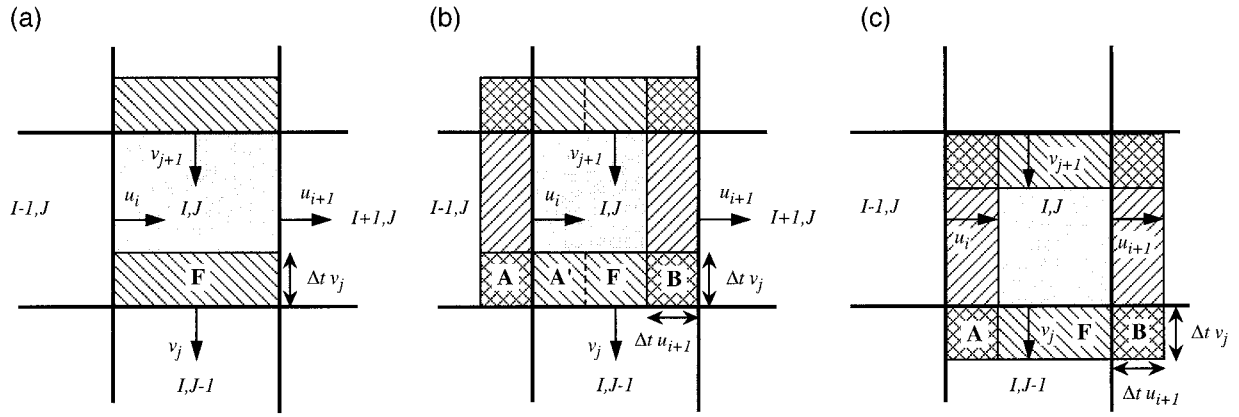

FIG. 4. Schematic of the transport of various sections of the domain when $\partial u/\partial x = 0$ and $\partial v/\partial y = 0$: (a) method 1, (b) method 2 before the time step, and (c) method 2 after the time step. Here, F is the area that will move from cell I, J to the cell $I, J - 1$; A is the area that will move from the cell $I - 1, J$ to the cell $I, J - 1$, and A′ (enclosed between the dashed line and the bold lines) is the area that will be occupied by A at the end of the time step; B is the area that will move from the cell I, J to the cell $I + 1, J - 1$. Method 1 takes into account only F. Using method 2 the area that goes into $I, J - 1$ is equal to the area F + A − B. At the end of the time step, the fact that A occupies the area A′ in F compensates for the loss of area B from F.

field is not monodimensionally divergence free, it will produce an artificial gradient in a. Due to this term, even a monotonic advection scheme will produce over- and undershooting. For this first test, method 1 gives the exact solution, but method 2 does not.

The second test is the classic cone in a solid body rotational flow field. The flow field velocities are defined over a 100 × 100 grid cell domain:

$$u = -\omega y \quad \text{and} \quad v = \omega x \quad \text{with} \quad \omega = 0.1.$$

The initial condition has a cone of radius 15 grids and height of 5, above a background of base value 1. The cone is centered at (50, 75). This test is often used to investigate the accuracy of advection schemes (Crowley 1968; Smolarkevicz 1984; Bott 1989, 1992). It is important to note that this rotational flow field is monodimensionally divergence free ($\partial u/\partial x = \partial v/\partial y = 0$).

During this test, the cone should rotate around the domain, keeping the same shape. Method 2 conserves the shape of the cone except the peak is truncated. Method 1 creates overshooting and distortion, which is greater for high Courant numbers (Fig. 3).

With the results of these two tests, we can conclude that method 2 generates supplemental terms that partially correct the solution of method 1, but it introduces an error when $\partial u/\partial x \neq 0$ or $\partial v/\partial y \neq 0$, which is the general case. Thus, the relative accuracy of the methods are highly dependent on the flow field and the choice of which is better is not clear.

4. Development of a correction factor for use in multidimensional modeling

To understand the origin of the errors in these two methods, we will analyze more precisely the differences between the two calculations.

Referring to Fig. 4, the flux, across the face j between cell I, J and $I, J - 1$ calculated using method 1 [see (3.2)], is equal to

$$f_j = \frac{\Delta t}{\Delta y} v_j \overline{a}^y_j, \tag{4.1}$$

where $\overline{a}^y_j$ represents the average of a over the area F [(2.3), Figs. 4 and 5].

Method 2 modifies this flux to add a correction term. Equation (3.5) gives

$$f_j = \frac{\Delta t}{\Delta y} v_j \left(\overline{a_I^n - \frac{\Delta t}{\Delta x} \Delta (u \overline{a}^x)}^y \right)_j. \tag{4.2}$$

This accounts for, in part, the flux of a into the cell from upwind cells. This flux can be also written as

$$f_j = \frac{\Delta t}{\Delta y} v_j \overline{a}^y_j - \frac{\Delta t^2}{\Delta y \Delta x} v_j u_{i+1} \overline{\overline{a}^x_{i+1}}^y + \frac{\Delta t^2}{\Delta y \Delta x} v_j u_i \overline{\overline{a}^x_i}^y, \tag{4.3}$$

where $\overline{\overline{a}^y_i}^x$ represents the average of a over the area A and $\overline{\overline{a}^x_{i+1}}^y$ that over B (Figs. 4 and 5).

Note, as shown in Fig. 4, the velocity in the Y direction is negative.

Comparing expressions (4.1) and (4.3), we can see that method 1 calculates only the flux coming from the cell I, J to the cell $I, J - 1$ (area F in Fig. 4),

$$f^1_j = \frac{\Delta t}{\Delta y} v_j \overline{a}^y_j, \tag{4.4}$$

and does not include any diagonal fluxes from cell $I - 1, J$ to $I, J - 1$ and from I, J to $I + 1, J - 1$. Here the superscript of f refers to the order of the method; that is, f^1_j is the first-order flux calculation.

The supplemental terms of method 2 enable the method to account for the diagonal fluxes from the cell $I - 1, J$ to the cell $I, J - 1$ (area A in Fig. 4),

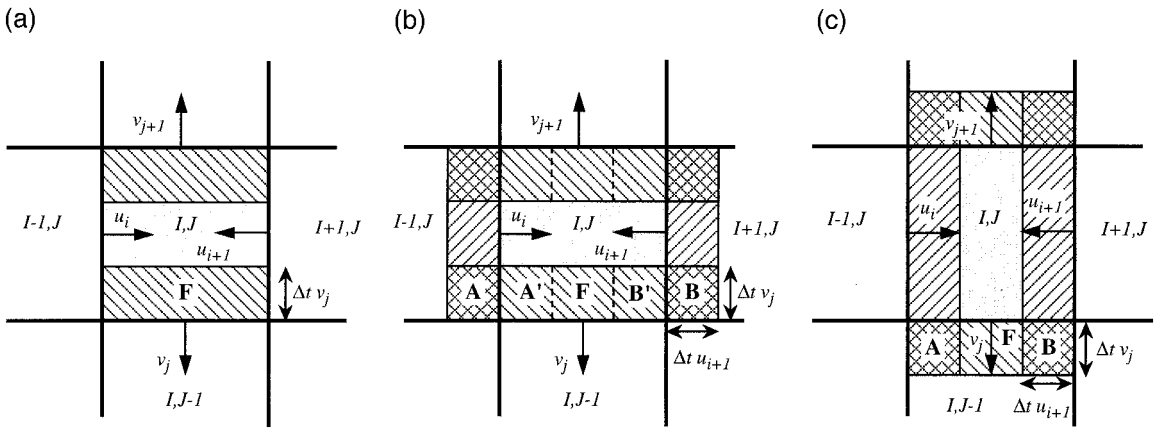

FIG. 5. Same as Fig. 4 but $\partial u/\partial x \neq 0$ and $\partial v/\partial y \neq 0$ ($u_i = -u_{i+1}$; $v_j = -v_{j+1}$) (a) method 1, (b) method 2 before the time step, and (c) method 2 after the time step. Now B is the area that will move from the cell $I + 1$, J to the cell I, $J - 1$ and B′ (enclosed between the dashed line and the bold lines) is the area that will be occupied by B at the end of the time step. Areas A and B will move into the same cell (I, $J - 1$). Using method 2, the area that goes into I, J is equal to F + A + B. At the end of the time step, the part of the area F occupied by A and B is equal to A′ + B′. Due to the fact that A′ + B′ is taken into account twice, the calculation of the average mass that crosses the face j is erroneous. The area that goes into I, $J - 1$ has to be equal to F + A + B − A′ − B′.

$$f^2_{j,i} = \frac{\Delta t^2}{\Delta y \Delta x} v_j u_i \overline{\overline{a}^x_i}^y, \tag{4.5}$$

and from the cell I, J to the cell $I + 1$, $J - 1$ (area B in Fig. 4),

$$f^2_{j,i+1} = \frac{\Delta t^2}{\Delta y \Delta x} v_j u_{i+1} \overline{\overline{a}^x_{i+1}}^y. \tag{4.6}$$

In this case, the supplemental terms are second order.

With method 2, the net flux at the face j (f_j) is equal to the flux coming from I, $J - 1$ (f^1_j) plus the flux coming from $I - 1$, J ($f^2_{j,i}$) minus the flux going to $I + 1$, $J - 1$ ($f^2_{j,i+1}$). If $\partial u/\partial x = 0$, the additional flux coming from $I - 1$, J compensates for the flux going to $I + 1$, $J - 1$ (Fig. 4). This explains why the results of method 2 are best when $\partial u/\partial x = \partial v/\partial y = 0$.

Consider next an example where $u_{i+1} = -u_i$. We can see from Fig. 5 that method 2 does not take into account the fact that $f^2_{j,i}$ and $f^2_{j,i+1}$ will replace a part of f^1_j. When $\partial u/\partial x \neq 0$, $f^2_{j,i+1}$ does not fully compensate for $f^2_{j,i}$.

The "lost" part of f^1_j is equal to

$$f^c_j = \frac{\Delta t^2}{\Delta y \Delta x} v_j \overline{a}^y_j u_{i+1} - \frac{\Delta t^2}{\Delta y \Delta x} v_j \overline{a}^y_j u_i, \tag{4.7}$$

which corresponds to areas A′ and B′ in Fig. 5. The superscript c refers to the correction needed for the lost flux. To correct this effect, the flux at the face j must be calculated with all the appropriate terms. This gives

$$f_j = \frac{\Delta t}{\Delta y} v_j \overline{a}^y_j - \frac{\Delta t^2}{\Delta y \Delta x} v_j u_{i+1} \overline{\overline{a}^x_{i+1}}^y + \frac{\Delta t^2}{\Delta y \Delta x} v_j u_{i+1} \overline{\overline{a}^x_i}^y + \frac{\Delta t^2}{\Delta y \Delta x} v_j \overline{a}^y_j u_{i+1} - \frac{\Delta t^2}{\Delta y \Delta x} v_j \overline{a}^y_j u_i, \tag{4.8}$$

which can be written as

$$f_j = \frac{\Delta t}{\Delta y} v_j \overline{a}^y_j - \frac{\Delta t^2}{\Delta y \Delta x} v_j \Delta\left(u \overline{\overline{a}^x}^y\right) + \frac{\Delta t^2}{\Delta y \Delta x} v_j \overline{a}^y_j \Delta u. \tag{4.9}$$

Using the same method, it is also possible to calculate

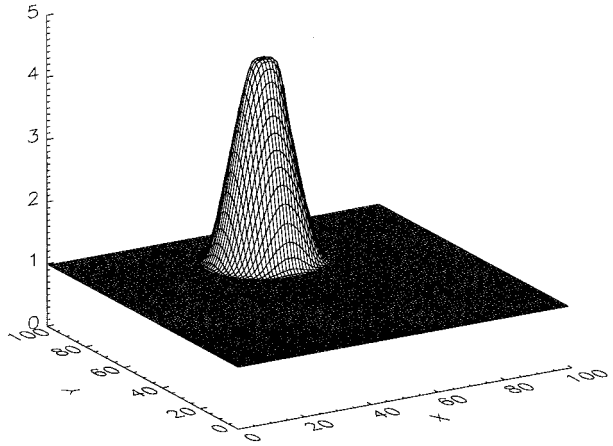

FIG. 6. Cone after six rotations: method 2′ with $\Delta t = 0.1$.

TABLE 1. Performance for the Crowley cone test at the end of six rotations. The error max and rms error are calculated as, respectively, $\max|a^6 - a^0|$ and $\{n^{-1}\Sigma[a^6 - a^0)/a^0]^2\}^{1/2}$, where a^0 is the analytical solution, a^6 the solution after six rotations, and n the number of points in the domain. The CPU time is normalized by the CPU needed to perform the test using method 1 (472.62 s on a SUN Sparc 10).

Rotating cone	Max	Min	Error max	rms error	CPU time
Method 1 $\Delta t = 0.1$	3.95	0.64	1.65	13.08	1
Method 1 $\Delta t = 0.01$	4.15	1.00	0.85	3.65	10
Method 2 $\Delta t = 0.1$	4.22	1.00	0.78	1.71	1
Method 2′ $\Delta t = 0.1$	4.22	1.00	0.78	1.71	1
Method 2″ $\Delta t = 0.1$	4.22	1.00	0.78	1.69	1

TABLE 2. Performance for the cube in a rotational shear flow test at the end of one cycle. The error max and rms error are calculated as, respectively, $\max|a^1 - a^0|$ and $\{n^{-1}\Sigma[(a^1 - a^0)/a^0]^2\}^{1/2}$, where a^0 is the analytical solution, a^1 the solution after one cycle, and n the number of points in the domain. The CPU time is normalized by the CPU needed to perform the test using method 1. (94.76 s on a SUN Sparc 10).

Shear cube	Max	Min	Error max	rms error	CPU time
Method 1 $\Delta t = 0.2$	3.6	1.00	3.85	46.74	1
Method 1 $\Delta t = 0.02$	4.87	1.00	3.80	33.85	10
Method 2 $\Delta t = 0.2$	5.36	0.77	3.94	34.00	1
Method 2′ $\Delta t = 0.2$	4.98	1.00	3.79	31.05	1
Method 2″ $\Delta t = 0.2$	4.98	1.00	3.78	30.54	1

the flux for the face $j + 1$. The difference between the two fluxes gives

$$f_{j+1} - f_j = \frac{\Delta t}{\Delta y}\Delta\left[v\left(\overline{a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x) + \frac{\Delta t}{\Delta x}a_I^n\Delta u}^{\,y}\right)\right]. \quad (4.10)$$

TABLE 3. Results for the cube in a rotational shear flow test with values given at the middle of the cycle when errors are greatest. Differences are shown with respect to method 2″.

Shear cube	Max	Min	Max difference
Method 2	5.19	0.88	0.73
Method 2′	4.99	1.00	0.29

The value of a_I at time $n + 1$ can be calculated as the value of a_I at time n plus the sum of all the fluxes:

$$a_I^{n+1} = a_I^n - f_{i+1} + f_i - f_{j+1} + f_j. \quad (4.11)$$

Equations (4.11) and (4.10) give

$$a_I^{n+1} = a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x) - \frac{\Delta t}{\Delta y}\Delta\left[v\left(\overline{a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x) + \frac{\Delta t}{\Delta x}a_I^n\Delta u}^{\,y}\right)\right]. \quad (4.12)$$

The expression (4.12) corresponds to the expression

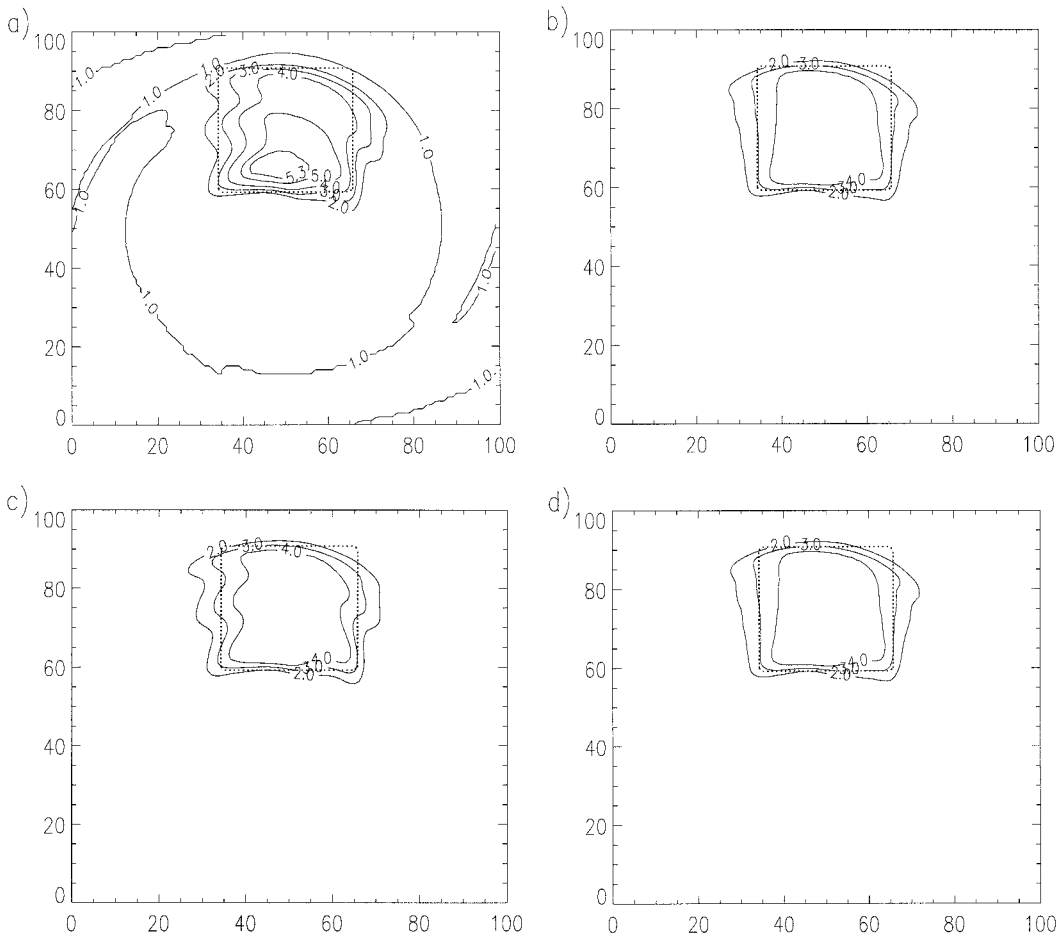

FIG. 7. Cube after one cycle: (a) method 2, (b) method 2″, and (c) method 2′. The analytical solution is where the original concentration distribution has a cube of height 5 (the dotted line).

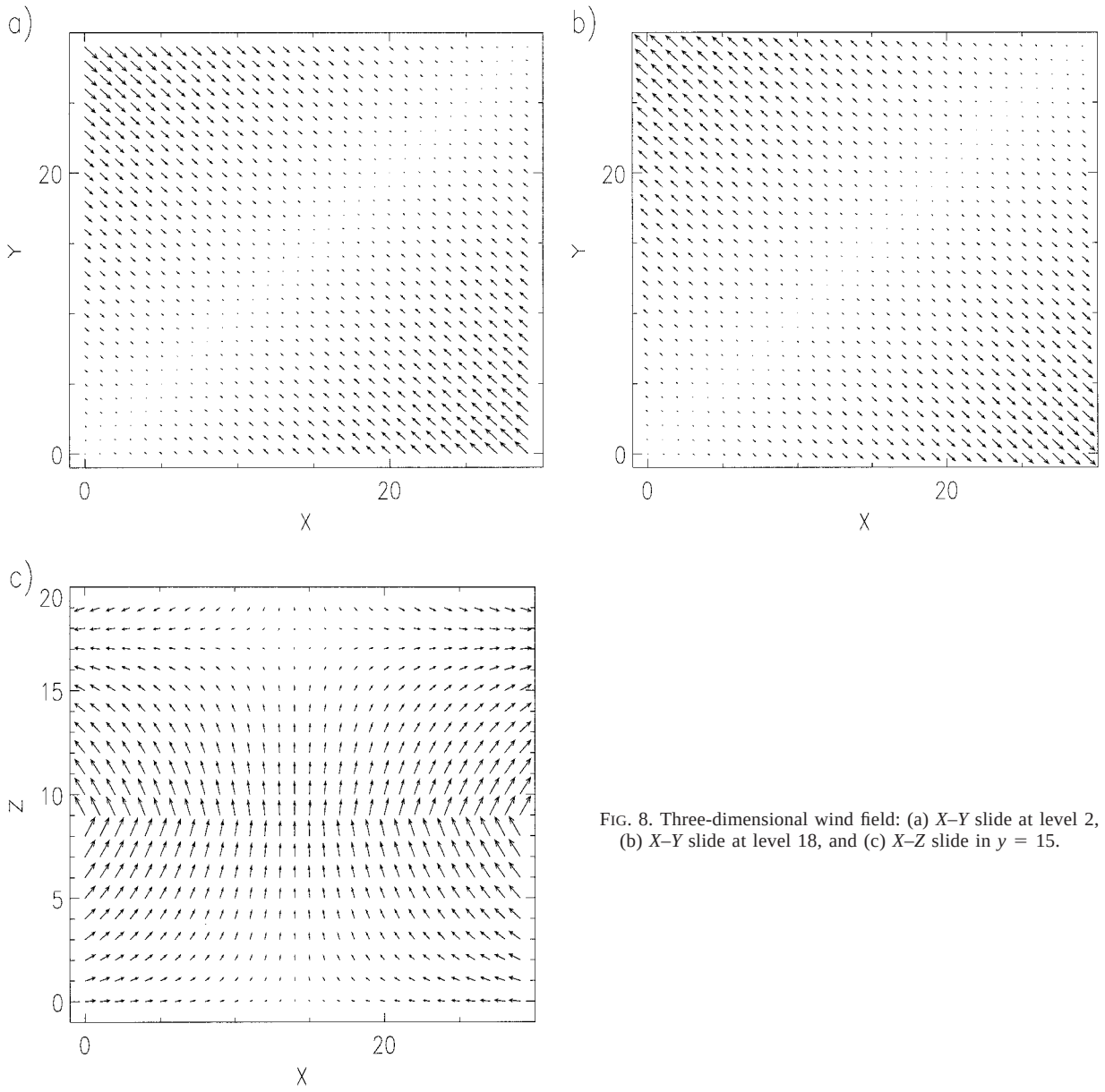

FIG. 8. Three-dimensional wind field: (a) *X–Y* slide at level 2, (b) *X–Y* slide at level 18, and (c) *X–Z* slide in $y = 15$.

(3.5) with a correction term. Equation (3.5) was produced by method 2 [Eq. (3.3)]. As for (3.5), it is possible to calculate (4.12) using a splitting method that gives method 2′ shown below:

$$a_I' = a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x)$$

$$a_I^c = a_I' - a_I^n\frac{\Delta t}{\Delta x}\Delta u$$

$$a_I^{n+1} = a_I' - \frac{\Delta t}{\Delta y}\Delta\left(v\overline{a^c}^y\right). \qquad (4.13)$$

The use of a_I^c to calculate the flux in the Y direction introduces the correction term needed to avoid the formation of an artificial gradient in the deformational flow field test.

5. Two-dimensional tests

Method 2′ gives the analytical solution for the test of the deformational flow field, but now it also gives the same results as method 2 for the rotating cone test (Fig. 6). Method 2′ does not need significantly more CPU time than method 2 because the calculation of a_I^c is very fast (Table 1).

A better test between methods 2 and 2′ can be done using a third test (Odman et al. 1996). In this test, the flow field is a parabolic rotating field:

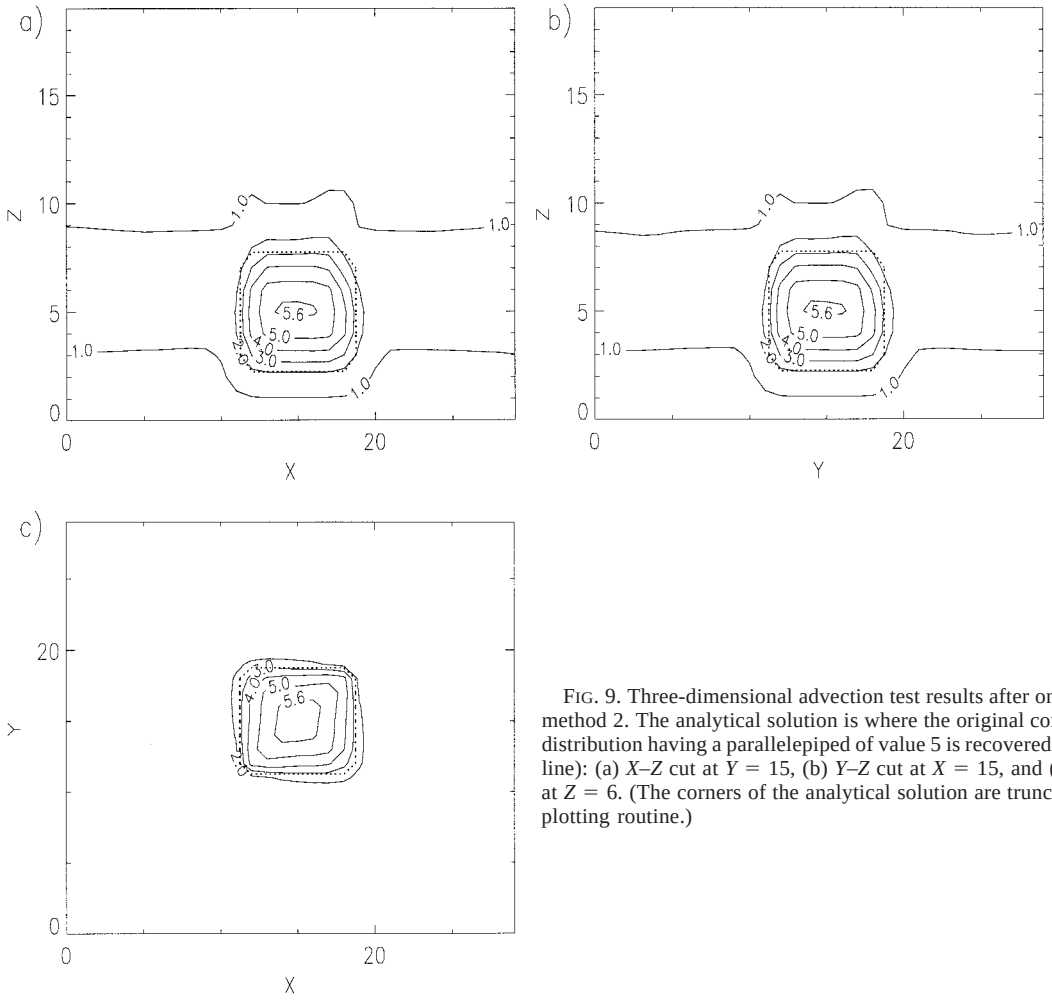

FIG. 9. Three-dimensional advection test results after one cycle for method 2. The analytical solution is where the original concentration distribution having a parallelepiped of value 5 is recovered (the dotted line): (a) *X–Z* cut at $Y = 15$, (b) *Y–Z* cut at $X = 15$, and (c) *X–Y* cut at $Z = 6$. (The corners of the analytical solution are truncated by the plotting routine.)

$$u = -2\omega y\left(1 - \frac{\sqrt{x^2 + y^2}}{R}\right)$$

$$v = -2\omega x\left(1 - \frac{\sqrt{x^2 + y^2}}{R}\right),$$

with $\omega = 0.1$ and $R = 50$.

Inside the domain with a background value of 1, a cube with a base of 30×30 points and a value of 5 is placed at the location coordinates (50, 75). This cube is transported for one-half cycle, and the midpoint ($r = R/2$) makes one revolution, distorting the cube. Then the flow field is reversed and the distorted cube is transported in the other direction. After this calculation, the cube should come back to its initial position and shape with a minimum of deformation.

In this test, method 2 shows strong overshooting: the maximum value after one cycle of 100 time steps is 5.36 instead of 5, the analytical result. Method 2′ does not produce any over- or undershooting. However, Fig. 7 shows that method 2′ generates errors.

In method 2′, the one-dimensional calculation is always applied with the same sequence: *X* first, then *Y*. Marchuk (1975) showed that if the sequence is reversed each $\Delta t/2$, the time splitting method becomes second-order accurate. In method 2″, the one-dimensional calculation is done first in the *X* direction, then in the *Y*, then another time for *Y*, and finally for *X*. This will be referred to as the time-split, alternating direction approach and is applied in these tests with correction. Figure 7 shows that method 2″ gives better results (Table 2).

The maximum difference (0.29) between the two methods appears at the middle of the simulation when the distorted cube starts to reverse (Table 3).

From these tests, we can conclude that method 2″, the time-split, alternating sequence, corrected version, is the best compromise between accuracy and computational time. It should be noted that, while this analysis has been applied to the PPM approach, it can be extended to others as well. The correction developed is physically natural and readily implemented with little computational penalty.

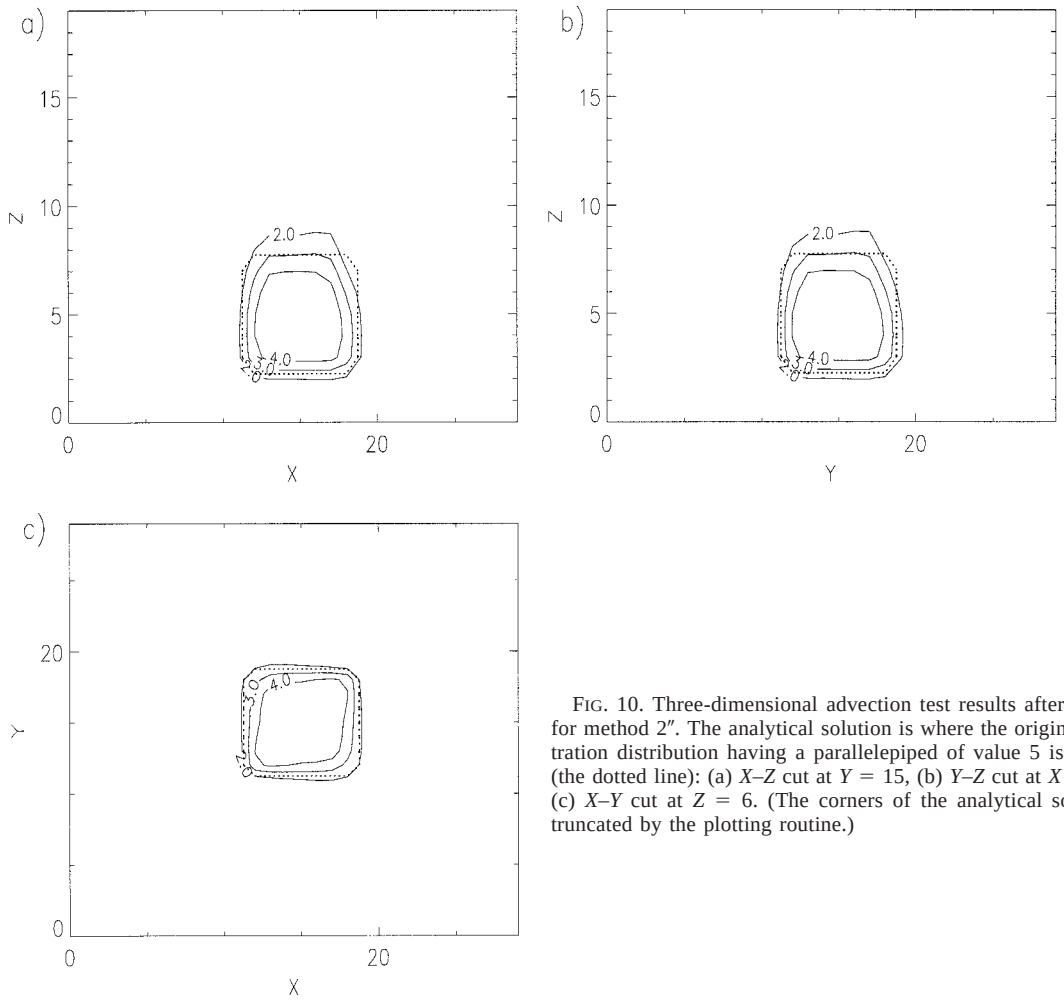

FIG. 10. Three-dimensional advection test results after one cycle for method 2″. The analytical solution is where the original concentration distribution having a parallelepiped of value 5 is recovered (the dotted line): (a) *X–Z* cut at $Y = 15$, (b) *Y–Z* cut at $X = 15$, and (c) *X–Y* cut at $Z = 6$. (The corners of the analytical solution are truncated by the plotting routine.)

6. Extension to three-dimensional advection

To solve the advection equation in three dimensions,

$$\frac{\partial a}{\partial t} + \frac{\partial ua}{\partial x} + \frac{\partial va}{\partial y} + \frac{\partial wa}{\partial z} = 0, \tag{6.1}$$

the time-splitting procedure of method 2″ can be extended. In this case the calculation sequence is

$$\begin{aligned} a_I' &= a_I^n - \frac{\Delta t}{\Delta x}\Delta(u\overline{a}^x) \\ a_I^c &= a_I' - a_I^n\frac{\Delta t}{\Delta x}\Delta u \\ a_I'' &= a_I' - \frac{\Delta t}{\Delta y}\Delta\left(v\overline{a^c}^y\right) \\ a_I^c &= a_I'' - a_I^n\frac{\Delta t}{\Delta y}\Delta v \\ a_I^{n+1} &= a_I'' - \frac{\Delta t}{\Delta z}\Delta\left(w\overline{a^c}^z\right). \end{aligned} \tag{6.2}$$

In Section 3 it was shown that the method 2 generates the supplemental term (3.8) in 2D. In 3D, a similar demonstration can show that method 2 generates more supplemental terms:

$$a\frac{\Delta t^2}{\Delta y\Delta x}\Delta(v\Delta u), \qquad a\frac{\Delta t^2}{\Delta z\Delta x}\Delta(w\Delta u),$$

$$\text{and} \quad a\frac{\Delta t^3}{\Delta z\Delta y\Delta x}\Delta[w\Delta(v\Delta u)]. \tag{6.3}$$

Due to these terms, more errors are produced in 3D. In order to quantify these errors, method 2″ and the uncorrected method 2, with alternating directions, were applied to a new three-dimensional test case. In this case, the modeling domain is a parallelepiped with grid dimensions of $30 \times 30 \times 20$ height, and the background value is 1. Inside the domain, a smaller parallelepiped of dimension $6 \times 6 \times 4$ high grids with a value of 5 is centered at (15, 15, 6) grids. The flow field is a diagonal stagnation flow in the first 10 levels in the *X–Y* plane (Fig. 8), which reverses in the top 10 levels. Math-

ematically, for the bottom 10 levels, the flow field is given as a parabolic law:

$$r = (x - y)/\sqrt{2}$$

$$\text{if} \quad r > 0 \qquad u = -Ar^2 - Br \quad \text{and} \quad v = Ar^2 + Br$$

$$\text{if} \quad r < 0 \qquad u = Ar^2 - Br \quad \text{and} \quad v = -Ar^2 + Br,$$

with

$$A = \frac{1}{r_{\min} - r_{\max}}\left(\frac{u_{\min}}{r_{\min}} - \frac{u_{\max}}{r_{\max}}\right)$$

$$B = \frac{1}{1/r_{\min} - 1/r_{\max}}\left(\frac{u_{\min}}{r_{\min}^2} - \frac{u_{\max}}{r_{\max}^2}\right)$$

$$u_{\min} = 0.05 \quad \text{and} \quad u_{\max} = 1$$

$$r_{\min} = \sqrt{2} \quad \text{and} \quad r_{\max} = \sqrt{450},$$

$w = 0$ at $z = 0$ (the bottom), and $w(x, y, z)$ elsewhere, is found from solving the three-dimensional continuity equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0.$$

The cube is transported for 14 time steps, then the flow is reversed, and transport is followed in the other direction for 14 time steps. The cube should be recovered. Method 2, that is, the alternating direction method without correction, leads to a maximum value of 5.66 instead of 5 and a minimum value of 0.899 instead of 1 (Fig. 9). Method 2′, that is, with the correction, shows no over- or undershooting, with the maximum predicted of 4.99 (Fig. 10).

It is interesting to notice that the maximum of 5.36 generated in 2D by method 2 was obtained after 100 time steps. In 3D, the maximum of 5.66 is reached after only $2 \times 14 = 28$ time steps.

7. Discussion and summary

The above analysis developed a natural and effective correction for use in multidimensional transport modeling in two and three dimensions. One of the important results is that over- and undershooting is more severe in three-dimensional tests. This suggests that further testing of transport algorithms should include three-dimensional tests (with nonmonodimensionally divergence free flow fields) because these algorithms are frequently being applied in three dimensions. The three-dimensional test developed here was able to expose the severe error growth found when using one of the methods.

It should be noted that these are very severe tests of transport algorithms and are intended to expose any weakness in and bring about possible improvements to various schemes. In a typical application, the flow fields are usually much smoother, generally decreasing the errors resulting from the transport calculation. An important point identified here is that some methods lead to error terms that grow, continuously, in time, while others do not. Further, some ostensibly accurate methods can lead to significant overshooting, but there are natural corrections to such problems.

The above analysis and tests show that a very straightforward correction term can be applied to operator split methods to improve performance. The correction does not significantly impact computational time. This analysis also suggests that the type of errors generated by different approaches can be identified a priori, and similar approaches can be used to develop correction factors. The method suggested here, from two- and three-dimensional tests, as being most robust, while maintaining computational efficiency, involves operator splitting and alternating direction, with a new correction factor for multidimensional modeling.

Acknowledgments. The author would like to thank Dr. A. G. Russell for many discussions and his help in preparing this article. In addition, thanks go to Drs. R. Yamartino and M. T. Odman for useful discussions. This work was supported by the Office Fédérale de l'Education et de la Sciences (OFES), the Swiss Fond National (FN), and the Swiss Federal Institute of Technology (EPFL).

REFERENCES

Bott, A., 1989: A positive definite advection scheme obtained by nonlinear renormalization of the advection fluxes. *Mon. Wea. Rev.*, **117,** 1006–1015.

——, 1992: Monotone flux limitation in the area-preserving flux-form advection algorithm. *Mon. Wea. Rev.*, **120,** 2592–2602.

——, 1993: The monotone area-preserving flux-form advection algorithm: Reducing the time-splitting error in two-dimensional flow fields. *Mon. Wea. Rev.*, **121,** 2637–2641.

Carpenter, R. L., K. K. Droegemeier, P. R. Woodward, and C. E. Hane, 1990: Application of the piecewise parabolic method (PPM) to meteorological modeling. *Mon. Wea. Rev.*, **118,** 586–612.

Chock, D. P., 1991: A comparison of numerical methods for solving the advection equation III. *Atmos. Environ.*, **25A,** 853–871.

——, and S. L. Winkler, 1994: A comparison of advection algorithms coupled with chemistry. *Atmos. Environ.*, **28,** 2659–2675.

Colella, P., and P. R. Woodward, 1984: The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comput. Phys.*, **54,** 174–201.

Crowley, W. P., 1968: Numerical advection experiments. *Mon. Wea. Rev.*, **96,** 1–11.

Easter, R. C., 1993: Two modified versions of Bott's positive-definite numerical advection scheme. *Mon. Wea. Rev.*, **121,** 297–304.

Marchuk, G. I., 1975: *Method of Numerical Mathematics.* Springer-Verlag, 316 pp.

Odman, M. T., and A. G. Russell, 1991a: Multiscale modelling of pollutant transport and chemistry. *J. Geophys. Res.*, **96,** 7363–7370.

——, and ——, 1991b: A multiscale finite element pollutant transport scheme for urban and regional modeling. *Atmos. Environ.*, **25A,** 2385–2394.

——, J. G. Wilkinson, L. A. McNair, A. G. Russell, C. L. Ingram, and M. R. Houyoux, 1995: Horizontal advection solver uncer-

tainty in the urban airshed model. Report to the California Air Resources Board No. 93-722, 103 pp. [Available from MCNC–North Carolina Supercomputing Center, P.O. Box 12889, 3021 Cornwallis Road, Research Triangle Park, NC 27709-2889.]

Smolarkiewicz, P. K., 1984: A fully multidimensional positive definite advection transport algorithm with small implicit diffusion. *J. Comput. Phys.,* **54,** 325–362.

Staniforth, A., J. Côté, and J. Pudykiewicz, 1987: Comments on ''Smolarkiewicz's deformational flow.'' *Mon. Wea. Rev.,* **115,** 894–900.

Yamartino, R. J., 1993: Nonnegative, conserved scalar transport using grid-cell-centered, spectrally constrained blackman cubics for applications on a variable-thickness mesh. *Mon. Wea. Rev.,* **121,** 753–763.