A Class of Monotone Interpolation Schemes

PIOTR K. SMOLARKIEWICZ AND GEORG A. GRELL

National Center for Atmospheric Research, Boulder, Colorado 80307*

Received January 27, 1991; revised September 13, 1991

This paper discusses a class of monotone (nonoscillatory) interpolation schemes convenient for applications with a variety of problems arising in computational fluid dynamics. These interpolators derive from the flux-corrected-transport finite difference advection schemes. It is shown that any known dissipative advection algorithm may be implemented as an interpolation scheme. The resulting interpolation procedure retains the formal accuracy of the advection scheme and offers such attractive computational properties as preservation of a sign or monotonicity of the interpolated variable. The derived class of interpolators consists of schemes of different levels of accuracy, efficiency, and complexity reflecting a rich variety of available advection schemes. Theoretical considerations are illustrated with idealized examples and selected applications to atmospheric fluid dynamics problems. © 1992 Academic Press, Inc.

1. INTRODUCTION

A complaint about traditional polynomial-fitting methods used for interpolating scalar fields defined on a discrete mesh is that they often lead to spurious numerical oscillations in regions of steep gradients of the interpolated variables. In order to suppress computational noise, characteristic of quadratic and higher-order interpolation schemes, one often implements a smoothing procedure, or adopts a linear (first-order) interpolation technique. These, however, introduce excessive numerical diffusion that smears out sharp features of interpolated fields. A more advanced approach invokes the so-called shape-preserving interpolation, which incorporates appropriate constraints on the derivative estimates used in the interpolation schemes (cf., [1] for a review). In this paper we consider an alternate approach and supplement the existing shape-preserving interpolation procedures with a class of schemes derived from monotone advection algorithms.

Finite-difference schemes for solving the advection equation

$$\frac{\partial \phi}{\partial t} + \nabla \circ (\mathbf{v}\phi) = 0 \tag{1}$$

at an arbitrary point $(\mathbf{x}_i, t^n)$ of a discrete mesh in $\mathbf{R}^M \times \mathbf{R}^1$ (assume a scalar variable $\phi(\mathbf{x}, t)$ is advected with a flow field $\mathbf{v}(\mathbf{x}, t)$ solonoidal in M spatial dimensions), were often derived via Lagrangian arguments requiring interpolation of the advected field at the departure point $(\mathbf{x}_0, t^{n-1})$ of a parcel's trajectory arriving at $(\mathbf{x}_i, t^n)$ (e.g., [2–4]). However, such arguments are not necessary; and, to our knowledge, all advection algorithms can be (and the majority have been) derived, analyzed, and studied, in abstraction from Lagrangian interpretation and without invoking an explicit interpolation procedure. In particular, a variety of advanced, monotone advection algorithms (cf., [5, 6] for reviews) were developed in the spirit of flux-corrected-transport (FCT)—an Eulerian concept which imposes appropriate constraints on fluxes from higher-order conservative advection schemes to assure monotonicity of the transported variables [7–9]. Starting with a shape-preserving interpolator, one can derive a monotone advection algorithm *via* Lagrangian arguments [1]; however, given a variety of existing advection schemes with attractive properties, the inverse problem of designing a shape-preserving interpolator starting with a monotone transport scheme appears to be an intriguing alternative.

The central theoretical issue concerns the formal equivalence of the finite difference operators for interpolation and advection on discrete meshes. The general idea of such an equivalence is certainly not new. The arbitrary-Lagrangian-Eulerian (ALE) models for fluids often refer to such a relationschip in the rezoning (remapping) stage of the computational procedure [10, 11]. However, the arguments invoked are partially intuitive, which leaves a certain margin of uncertainty with respect to the exact versus the approximate aspects of the equivalence. The remapping prodecures of ALE models are strictly conservative and optionally monotone. Although powerful, they are complex and computationally expensive and thus recommended only for those applications where the exact conservation is essential [12]. Relaxing this constraint considerably simplifies the theoretical and practical aspects of the "advection-interpolation" equivalence; designing a strictly conservative interpolation operator requires analysis in terms of volume

* The National Center for Atmospheric Research is sponsored by the National Science Foundation.

0021-9991/92 $5.00

and surface integrals, whereas operators which conserve interpolated fields only with accuracy to the truncation error emerge from simple arguments in terms of contour integrations. The theoretical basis for the "advection-interpolation" equivalence exploited herein represents a special case of a more general theory of transport algorithms [13]. However, since the focus of that work was on different issues, the utility of the development therein for designing a class of interpolation schemes may not be immediately obvious.

The paper is organized as follows. Section 2 discusses mathematical foundations of the "advection-interpolation" equivalence. Section 3 summarizes general aspects of interpolation schemes derived from finite difference advection algorithms and discusses details of the implementation of FCT advection algorithms as monotone interpolators. Section 4 illustrates the theoretical considerations with examples of applications. A summary of our conclusions is presented in Section 5.

2. "ADVECTION-INTERPOLATION" EQUIVALENCE

Consider the following interpolation problem: Let us assume that a sufficiently continuous field $\psi\colon \mathbf{R}^M \to \mathbf{R}^1$ is known a priori in $\mathbf{x}_i$ points of a uniform mesh in $\mathbf{R}^M$. For clarity, we restrict ourselves to regular meshes with a constant grid interval $\mathbf{\Delta X} \equiv (\Delta X^1, \Delta X^2, ..., \Delta X^M)$ such that $\mathbf{x}_i = \mathbf{i} \circ \mathbf{\Delta X}$, although the results for irregular grids easily follow (cf., [13] for a discussion). The problem addressed is to provide an estimate of ψ, with certain desired accuracy, in a point of interest $\mathbf{x}_0$ noncoincident, in general, with any of $\mathbf{x}_i$'s. A traditional approach is to expand ψ in, say, the pth order Taylor sum about some $\mathbf{x}_i$ from a local neighborhood of $\mathbf{x}_0$ and provide adequate estimates of the derivatives using information available on the grid, or, equivalently, to evaluate ψ at $\mathbf{x}_0$, based on the assumption that ψ fits a pth order Lagrangian polynomial in between the grid points [4]. Consider, however, an alternate approach.

As a consequence of the Stokes' theorem

$$\psi(\mathbf{x}_0) - \psi(\mathbf{x}_i) = \int_C d\mathbf{x} \circ \nabla\psi(\mathbf{x}), \tag{2}$$

where C denotes an arbitrary contour connecting the point of interest $\mathbf{x}_0$ with an arbitrary $\mathbf{x}_i$ of the computational grid. Exploiting the arbitrariness of the contour selection in (2), we choose for C the line segment of the parametric equation

$$\mathbf{x}(\mathbf{x}_i, \tau) = -(\mathbf{x}_i - \mathbf{x}_0)\tau + \mathbf{x}_i, \tag{3}$$

where the parameter $\tau \in [0, 1]$. Implementing (3) in (2) gives

$$\phi(\mathbf{x}_i, 1) = \phi(\mathbf{x}_i, 0) - \int_0^1 (\mathbf{x}_i - \mathbf{x}_0) \circ \nabla\phi(\mathbf{x}_i, \tau)\, d\tau, \tag{4}$$

where

$$\phi(\mathbf{x}_i, \tau) \equiv \psi(\mathbf{x}(\mathbf{x}_i, \tau)). \tag{5}$$

Since for every fixed $\mathbf{x}_0$ and $\mathbf{x}_i$ the first element of the scalar product appearing under the integral in (4) is constant, (4) may be rewritten as

$$\phi(\mathbf{x}_i, 1) = \phi(\mathbf{x}_i, 0) - \int_0^1 \nabla \circ (\mathbf{U}\phi)(\mathbf{x}_i, \tau)\, d\tau, \tag{6}$$

where

$$\mathbf{U} = \mathbf{x}_i - \mathbf{x}_0. \tag{7}$$

Equation (6) represents the formal integral of the equation

$$\frac{\partial\phi}{\partial\tau} + \nabla \circ (\mathbf{U}\phi) = 0 \tag{8}$$

over the τ interval [0, 1] at $\mathbf{x}_i$ grid-point. In other words, (6) is a formal solution to the advection equation of the form (1) in which the free parameter τ plays the role of a time-independent variable, and the vector $\mathbf{U}$ defined in (7) plays the role of the velocity field. Therefore, the interpolation problem has been expressed as the equivalent advection problem. In principle, the formal solution (6) may be approximated using any known advection algorithm for (1). The truncation error of such an approximation represents the error of estimating ψ at the point of interest $\mathbf{x}_0$. In the following section we discuss the suitability and attractiveness of different advection schemes for implementation as interpolation algorithms in (6), as well as some particular adjustments of FCT advection schemes required for their efficient implementation as monotone interpolators. Before this, however, we would like to draw the reader's attention to certain theoretical aspects of the "advection-interpolation" equivalence.

One might wonder how (6) is related to the traditional interpolation approach which relies on the truncated Taylor expansion. Recall that the untruncated (exact) Taylor formula at the pth order of expansion of $\psi(\mathbf{x} + \mathbf{h})$ about $\mathbf{x}$ contains the $(p+1)$th remainder

$$R_{p+1}(\mathbf{x}, \mathbf{h}) = \left(\int_0^1 \frac{(1-s)^p}{p!} \mathscr{D}^{(p+1)}\psi(\mathbf{x} + s \cdot \mathbf{h})\, ds\right) \cdot \mathbf{h}^{(p+1)} \tag{9}$$

(Theorem VII.9.3 in [14]). Employing the untruncated zeroth-order Taylor expansion of $\psi(\mathbf{x}_0)$ about an arbitrary $\mathbf{x}_i$ results in

$$\psi(\mathbf{x}_0)=\psi(\mathbf{x}_i)+\left(\int_0^1 \nabla\psi(\mathbf{x}_i+s(\mathbf{x}_0-\mathbf{x}_i))\,ds\right)\cdot(\mathbf{x}_0-\mathbf{x}_i). \tag{10}$$

Including $-(\mathbf{x}_0-\mathbf{x}_i)$ under the integral and using the definitions (7), (5), and (3), reproduces (6). Thus, the conceptual difference between the traditional approach and the one advocated herein might be interpreted as a different exploitation of the Taylor formula. The traditional approach employs the *truncated*, say the pth order, Taylor expansion with adequate approximations of the derivatives, whereas the current approach assumes zeroth order, but *untruncated* Taylor expansion with a pth-order approximation of the first remainder. Apparently, instead of deriving (6) from (2), one could obtain it directly from the Taylor formula. The derivation of (6) from (2) is, however, more general; the degree of freedom associated with a contour selection in (3) leads to different versions of (6). In particular, selecting the contour consisting of a sequence of the line segments parallel to spatial coordinates leads to the alternate-direction (time-split) representation of (6).

3. INTERPOLATION SCHEMES DERIVED FROM ADVECTION ALGORITHMS

3.1. *General Aspects*

The formal solution (6) to the interpolation problem may be evaluated with the help of any known explicit advection algorithm for (1)

$$\phi_i^n=\phi_i^{n-1}-\mathscr{A}\mathscr{S}(\phi,\boldsymbol{\alpha},\mathcal{O}(\mathbf{x}_i,t^{n-1})), \tag{11}$$

where

$$\boldsymbol{\alpha}=\frac{\mathbf{v}\,\Delta t}{\Delta\mathbf{X}}\equiv\left(\frac{v^1\,\Delta t}{\Delta X^1},\frac{v^2\,\Delta t}{\Delta X^2},\ldots,\frac{v^M\,\Delta t}{\Delta X^M}\right) \tag{12}$$

is a local Courant number vector, and the finite difference operator $\mathscr{A}\mathscr{S}$ identifies an advection scheme defined by its particular dependence on the values of ϕ and $\boldsymbol{\alpha}$ available on the mesh in a neighborhood $\mathcal{O}$ of the $(\mathbf{x}_i, t^{n-1})$ grid point. (Since the arguments of $\mathscr{A}\mathscr{S}$ may be arbitrarily staggered both in time and space, any finite difference advection algorithm may be written in the symbolic form (11).)

The convergent implementation of an advection algorithm (11) for the numerical evaluation of (6), or equivalently (8), necessarily requires the resulting Courant number

$$\tilde{\boldsymbol{\alpha}}=\frac{\mathbf{U}}{\Delta\mathbf{X}}\equiv\left(\frac{U^1}{\Delta X^1},\frac{U^2}{\Delta X^2},\ldots,\frac{U^M}{\Delta X^M}\right) \tag{13}$$

(recall that $\Delta\tau\equiv 1$) to be appropriately bounded; i.e.,

$$\|\tilde{\boldsymbol{\alpha}}\|\leqslant\mathscr{C}\in\mathbf{R}^1, \tag{14}$$

where the constant $\mathscr{C}$ depends on the algorithm employed and the norm selected. Choosing an arbitrary grid point $\mathbf{x}_i$ as

$$\mathbf{x}_i=[\mathbf{x}_0]\equiv\left(NI\left(\frac{x_0^1}{\Delta X^1}\right)\Delta X^1,\ NI\left(\frac{x_0^2}{\Delta X^2}\right)\Delta X^2,\ldots,NI\left(\frac{x_0^M}{\Delta X^M}\right)\Delta X^M\right), \tag{15}$$

where NI denotes the nearest integer value, ensures that

$$\underset{I}{\forall}\ |\tilde{\alpha}^I|\leqslant\frac{1}{2}, \tag{16}$$

which is sufficient for the stability of a majority of known finite-difference advection schemes in $M\leqslant 2$ spatial dimensions. Using definitions (15), (13), (7), and (5) the resulting interpolation algorithm may be compactly written as

$$\psi(\mathbf{x}_0)=\psi([\mathbf{x}_0])-\mathscr{A}\mathscr{S}\left(\psi,\frac{[\mathbf{x}_0]-\mathbf{x}_0}{\Delta\mathbf{X}},\mathcal{O}([\mathbf{x}_0])\right). \tag{17}$$

Although in principle any explicit advection scheme might be considered in (17), the multiple-time-level (e.g., centered-in-time) algorithms are not attractive for applications—since $\nabla\psi$ in (6) is not immediately available at intermediate points of the contour (3). In contrast, the class of the forward-in-time (dissipative) schemes is perfectly suited for (17). Note that the constancy of $\tilde{\boldsymbol{\alpha}}$ allows for straightforward, alternate-direction applications of one-dimensional schemes without degrading the formal accuracy of their constant coefficients limit (which is the alternative argument to that in the paragraph ending Section 2, for the exactness of the time-splitting). The latter has important implications for applications. First, in contrast to the arbitrary dimensional case, there is a great variety of methods suitable for integration of the one-dimensional, constant coefficient advection problem. Second, the boundedness of effective Courant numbers in (16) is sufficient for the stability of all one-dimensional dissipative schemes (known to the authors); this implies the stability of a time-split algorithm for an arbitrary M. Third, when compared to fully multidimensional algorithms, the alternate-direction implementation of a one-dimensional scheme has the virtue of considerable simplicity, especially when advanced monotone techniques are concerned. For illustration, we provide two elementary, one-dimensional examples of the interpolation algorithm (17) using common advection schemes.

Consider a finite-difference approximation in (17) using the popular, second-order-accurate Lax–Wendroff advection scheme [15] (alias Leith [2], Crowley [3])

$$\phi_0^1 = \phi_0^0 - \tfrac{1}{2}\beta(\phi_1^0 - \phi_{-1}^0) + \tfrac{1}{2}\beta^2(\phi_1^0 - 2\phi_0^0 + \phi_{-1}^0). \quad (18)$$

Identifying the Courant number β with $\tilde{\alpha} = ([x_0] - x_0)/\Delta X$ in (17), and accordingly with (3) and (5), ϕ_0^0 with $\psi([x_0])$, and ϕ_0^1 with $\psi(x_0)$, the Lax–Wendroff algorithm gives

$$\begin{aligned}\psi(x_0) = \psi([x_0]) &- \frac{1}{2}\frac{[x_0] - x_0}{\Delta X} \\ &\times (\psi([x_0] + \Delta X) - \psi([x_0] - \Delta X)) \\ &+ \frac{1}{2}\left(\frac{[x_0] - x_0}{\Delta X}\right)^2 \\ &\times (\psi([x_0] + \Delta X) - 2\psi([x_0]) + \psi([x_0] - \Delta X)),\end{aligned} \quad (19)$$

which is the second-order finite difference approximation to the second-order Taylor expansion of $\psi(x_0)$ about $[x_0]$,

$$\begin{aligned}\psi(x_0) = \psi([x_0]) &+ (x_0 - [x_0])\frac{\partial\psi}{\partial x}([x_0]) \\ &+ \frac{1}{2}(x_0 - [x_0])^2 \frac{\partial^2\psi}{\partial x^2}([x_0]) + O((x_0 - [x_0])^3).\end{aligned} \quad (20)$$

Similar examples, up to the tenth-order of accuracy of approximation, may be provided, employing formulae from pages 541–542 of the article by Tremback *et al.* [4].

Another special case (a counterexample) of finite-difference approximation in (17) employs the popular centered-in-time-and-space (leapfrog) advection scheme

$$\phi_0^1 = \phi_0^0 - \tfrac{1}{2}\beta(\phi_1^{1/2} - \phi_{-1}^{1/2}). \quad (21)$$

Similarly, as in the previous example, the Courant number β corresponds to $\tilde{\alpha}$, ϕ_0^0 to $\psi([x_0])$, and ϕ_0^1 to $\psi(x_0)$. This is, however, insufficient for a meaningful approximation in (17), since the leapfrog scheme requires $\phi_{\pm 1}^{1/2}$, corresponding to $\psi([x_0] - \frac{1}{2}\tilde{\alpha}\,\Delta X \pm \Delta X)$, which are not explicitly available on the computational grid. Therefore, the leapfrog scheme is not suitable for (17). In principle, one might approximate $\psi([x_0] - \frac{1}{2}\tilde{\alpha}\,\Delta X \pm \Delta X)$, using the information available on the grid, but this brings us back to direct implementation of the dissipative advection schemes. For instance, assuming that

$$\begin{aligned}&\psi([x_0] - \tfrac{1}{2}\tilde{\alpha}\,\Delta X + \Delta X) \\ &\quad\approx \psi([x_0]) + \tfrac{1}{2}(1 - \tfrac{1}{2}\tilde{\alpha}) \\ &\qquad\times (\psi([x_0] + \Delta X) - \psi([x_0] - \Delta X)) + \tfrac{1}{2}(1 - \tfrac{1}{2}\tilde{\alpha})^2 \\ &\qquad\times (\psi([x_0] + \Delta X) - 2\psi([x_0]) + \psi([x_0] - \Delta X))\end{aligned} \quad (22a)$$

and

$$\begin{aligned}&\psi([x_0] - \tfrac{1}{2}\tilde{\alpha}\,\Delta X - \Delta X) \\ &\quad\approx \psi([x_0]) - \tfrac{1}{2}(1 + \tfrac{1}{2}\tilde{\alpha}) \\ &\qquad\times (\psi([x_0] + \Delta X) - \psi([x_0] - \Delta X)) + \tfrac{1}{2}(1 + \tfrac{1}{2}\tilde{\alpha})^2 \\ &\qquad\times (\psi([x_0] + \Delta X) - 2\psi([x_0]) + \psi([x_0] - \Delta X))\end{aligned} \quad (22b)$$

reduces the resulting algorithm to a direct application of the Lax–Wendroff scheme; i.e., the current example reduces to the one discussed previously.

3.2. *FCT Interpolation Schemes*

Insofar as the linear, dissipative advection algorithms are concerned there is no particular gain in using (17). The first example in Section 3.1 indicates that the interpolation schemes derived from such algorithms may be obtained alternatively through more traditional arguments invoking truncated Taylor expansion, or equivalently, polynomial fitting [4]. However, where preservation of monotonicity and/or sign of the interpolated variable is concerned, formula (17) becomes a useful tool. For instance, sign-preserving interpolation may be easily achieved by direct implementation of the simple positive definite advection scheme of Smolarkiewicz [16]. Although the extended versions of this scheme (MPDATA family of algorithms [17, 18, 9, 19]) include strictly monotone approximations and appear to be competitive tools for evaluation of a general transport problem [9], they are not particularly attractive for applications as shape-preserving interpolators. In the special case of the constant coefficients limit, which is the case in (17), the FCT schemes built upon the high-order-accurate, constant-grid-flux dissipative algorithms of Tremback *et al.* [4] are excellent tools insofar as overall accuracy, efficiency, and simplicity are concerned [9, 13]. Consequently, they are especially attractive for applications with the interpolation formula (17). Since the FCT formalism has been reviewed recently [9], below we summarize the FCT procedure only to the extent necessary to draw the reader's attention to particular adjustments required for its efficient implementation with (17). Given the exactness of the alternate-direction representation of (17) it is sufficient to consider only one-dimensional FCT schemes. For compactness of numerical equations we shall use the same notation as in [9].

An FCT advection scheme may be compactly written as

$$\phi_i^{n+1} = \Phi_i^{n+1} - (\tilde{A}_{i+1/2} - \tilde{A}_{i-1/2}), \quad (23)$$

where Φ denotes a low-order, monotone solution to (1) and $\tilde{A}$ is the antidiffusive flux limited such as to ensure that the equations, which relates stability of the resulting scheme not to the magnitude of the flow field but to its variability along trajectories (29). In a variety of atmospheric applications, flows are sufficiently smooth to enable semi-Lagrangian integrations of (30) with Courant numbers exceeding unity even by orders of magnitude. This leads to considerable savings of the computational effort and improved accuracy of integrations (due to lesser accumulation of truncation error over the reduced number of the time steps required).

In meteorological literature, a common method of solving (29) invokes an iterative solution of the implicit midpoint rule

$$\mathbf{x}_0 - \mathbf{x}_\mathbf{i} = \mathbf{v}(\tfrac{1}{2}(\mathbf{x}_0 + \mathbf{x}_\mathbf{i}), \tfrac{1}{2}(t_0 + t))(t_0 - t), \qquad (31)$$

with the first guess $\mathbf{x}_0 = \mathbf{x}_\mathbf{i} + \mathbf{v}(\mathbf{x}_\mathbf{i}, t_0)(t_0 - t)$; the two iterations suffice for the second-order-accurate solution, providing the Courant number variation is appropriately bounded along the trajectory [20]. Since the midpoint $\frac{1}{2}(\mathbf{x}_0 + \mathbf{x}_\mathbf{i})$ appearing on the r.h.s. of (31), in general, does not coincide with a grid point, the velocity field must be interpolated to the midpoint's position. Therefore, the entire semi-Lagrangian algorithm for solving (30) requires interpolation procedures for both the velocity field and the advected scalar.

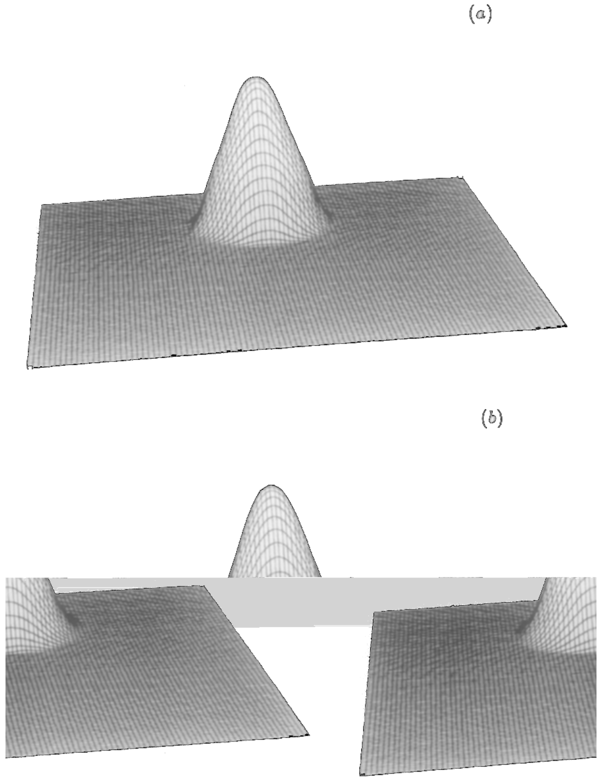

FIG. 1. The rotating cone test from [9, 16–19]. Both solutions are shown after six revolutions of the cone. The reference spikes in the upper-right and the lower-left corners represent the initial- and minus half of the initial-height of the cone placed on the large constant background. Plate (a) shows the semi-Lagrangian solution obtained with monotone interpolator; the number of the time-steps is 377 (the maximum Courant number is 10). Plate (b) displays the Eulerian solution obtained with monotone MPDATA scheme; the number of the time-steps is 3768 (the maximum Courant number is 1).

For illustration, we utilize the rotating cone test from [9, 16–19]; this test has been used to document the properties of MPDATA advection schemes. The two solutions in Fig. 1 are shown after six revolutions of the cone whose initial height is equal to the size of the reference spike in the upper right corner and whose radius at the base is equal to 15 grid intervals. The entire transported field consists of the cone and the large constant background (equal 25 initial heights of the cone). Figure 1a shows the semi-Lagrangian solution obtained with the interpolater (17) implementing the FCT version of the fourth-order-accurate dissipative scheme [4] (discussed in [9]). For comparison, Fig. 1b shows the Eulerian solution obtained with the non-oscillatory option of the third-order-accurate MPDATA scheme [19]. In the figures the two solutions are hardly distinguishable. Both solutions are monotone and excellently preserve the initial shape of the cone. There are, however, two important differences. In order to ensure stable integrations, the Eulerian scheme requires 3768 time steps, whereas the semi-Lagrangian scheme allows for a ten times larger temporal increment (the maximum Courant number in Fig. 1a is 10, which is close to the limit imposed by the convergence of iterations in (31)) and therefore requires one order of magnitude fewer time steps. Overall, the semi-Lagrangian solution is about six times "cheaper" than the Eulerian solution. The price to be paid for this gain in computational effort is the loss of exact conservation. The Eulerian solution conserves the integral of the transported field with accuracy to the round-off error, whereas the semi-Lagrangian solution exhibits a small, but not negligible, conservation error. The relative conservation error for the cone itself (i.e., disregarding a constant background which is free of error but falsely improves normalized accuracy measures) is about 1 % after the six revolutions of the cone.

The results discussed clearly illustrate the utility of the advocated approach. The reader interested in a comparative performance of the interpolators (17) employing different advection schemes is referred to [13], where a number of oscillatory and monotone algorithms have been tested in the context of semi-Lagrangian transport on the sphere.

4.2. *Local Mesh Refinement: Nested Grid Computations*

An accurate and efficient interpolation procedure constitutes a vital element of any local mesh refinement scheme that requires frequent mappings of fields between coarse solution (23) is free of local extrema absent in the low-order solution

$$\tilde{A}_{i+1/2} = \min(1, \beta_i^{\downarrow}, \beta_{i+1}^{\uparrow})[A_{i+1/2}]^{+} + \min(1, \beta_i^{\uparrow}, \beta_{i+1}^{\downarrow})[A_{i+1/2}]^{-}, \tag{24}$$

where

$$A_{i+1/2} \equiv FH_{i+1/2} - FL_{i+1/2}, \tag{25}$$

with FH and FL denoting fluxes from a high-order and a low-order advection scheme, respectively. $[\ ]^{+} \equiv \max(0, \ )$ and $[\ ]^{-} \equiv \min(0, \ )$ in (19) are the positive- and negative-part operators, respectively, and

$$\beta_i^{\uparrow} \equiv \frac{\phi_i^{\mathrm{MAX}} - \Phi_i^{n+1}}{A_i^{\mathrm{IN}} + \varepsilon}; \qquad \beta_i^{\downarrow} \equiv \frac{\Phi_i^{n+1} - \phi_i^{\mathrm{MIN}}}{A_i^{\mathrm{OUT}} + \varepsilon}, \tag{26a), (26b}$$

where A_i^{IN}, A_i^{OUT} are the absolute values of the total incoming and outgoing A-fluxes (25), from the ith grid box, respectively. ε is a small value, e.g., $\sim 10^{-15}$, allowing for efficient coding of β-ratios when A_i^{IN} or A_i^{OUT} vanish. The limiters ϕ_i^{MAX} and ϕ_i^{MIN} define monotonicity of the scheme (i.e., by design $\phi_i^{\mathrm{MIN}} \leqslant \phi^{n+1} \leqslant \phi_i^{\mathrm{MAX}}$), and they are traditionally specified [8] as

$$\phi_i^{\mathrm{MAX}} = \max(\phi_{i-1}^n, \phi_i^n, \phi_{i+1}^n, \Phi_{i-1}^{n+1}, \Phi_i^{n+1}, \Phi_{i-1}^{n+1}) \tag{27a}$$

$$\phi_i^{\mathrm{MIN}} = \min(\phi_{i-1}^n, \phi_i^n, \phi_{i+1}^n, \Phi_{i-1}^{n+1}, \Phi_i^{n+1}, \Phi_{i-1}^{n+1}). \tag{27b}$$

The reader interested in the formal derivation of a general FCT algorithm is referred to [9].

A shape-preserving interpolation scheme requires less restrictive monotonicity constraints than a conservative advection scheme. The minima over β ratios appearing in (24) ensure that the antidiffusive flux attributed to $i+1/2$ position on the grid does not contribute to generation of spurious extrema, either in gridbox i or in gridbox $i+1$. However, monotonicity of the interpolation scheme (17) only requires that $\phi_i^{n+1} = \psi(x_0)$ is free of spurious extrema. Consequently Eq. (24) may be replaced by

$$\tilde{A}_{i+1/2} = \min(1, \beta_i^{\downarrow})[A_{i+1/2}]^{+} + \min(1, \beta_i^{\uparrow},)[A_{i+1/2}]^{-}. \tag{24'}$$

Furthermore, since the effective flow field in (17) is constant, and therefore incompressible, the limiters in (27) may be simplified to

$$\phi_i^{\mathrm{MAX}} = \max(\phi_{i-1}^n, \phi_i^n, \phi_{i+1}^n, \Phi_i^{n+1},); \qquad \phi_i^{\mathrm{MIN}} = \min(\phi_{i-1}^n, \phi_i^n, \phi_{i+1}^n, \Phi_i^{n+1}), \tag{27'a), (27'b}$$

where the redundant dependence of the limiters on Φ_i^{n+1} has been retained to ensure strictly nonnegative values of the β ratios in (26) (cf., Section 3.1 in [9]). Since the low-order solution may always be written as an old value minus divergence of fluxes from the low-order scheme, the entire algorithm consisting of (23), (24′), (25), (26), and (27′) is in the form of a general advection scheme (11).

4. EXAMPLES OF APPLICATIONS

Further in this section we discuss examples of applications of the monotone interpolators to selected problems of atmospheric fluid dynamics. The performance of the algorithms is illustrated with the examples addressing two distinct classes of practical applications. The first example concerns the semi-Lagrangian advection of a passive scalar; it may be considered as a prototype for pollutant transport in a predetermined flow field. The second example deals with a problem of local mesh refinement within a context of a strongly stratified, hydrostatic flow past a three-dimensional hill.

4.1. *Semi-Lagrangian Advection*

The invariance of a scalar quantity

$$\psi(\mathbf{x}, t) = \psi(\mathbf{x}_0, t_0), \tag{28}$$

along a parcel's trajectory,

$$\mathbf{x}_0(t_0) = \mathbf{x}(t) + \int_t^{t_0} \mathbf{v}\, dt, \tag{29}$$

constitutes a basic element in a variety of fluid dynamics applications. The issue addressed is to approximate $\psi(\mathbf{x}, t)$ at all $\mathbf{x}_i$ points of a discrete mesh, assuming that at some earlier time t_0, ψ and the velocity field $\mathbf{v}$ are known at all $\mathbf{x}_i$'s. In other words, we are seeking an approximate solution to the advection problem

$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial t} + \mathbf{v} \cdot \nabla\psi = 0, \tag{30}$$

posed on the discrete grid. Approximate solutions to (30) via approximations to (29) and (28) are often referred to as semi-Lagrangian advection schemes [1]. A general concept of such a solution involves two distinct steps. The first step determines departure points $\mathbf{x}_0$'s of the trajectories arriving at $\mathbf{x}_i$'s through approximate solutions to (29), and the second step interpolates ψ's at $\mathbf{x}_0$'s. An apparent advantage of the semi-Lagrangian approach is that the restrictive Courant–Friedrichs–Lewy (CFL) stability criterion for the partial differential equation (30) is replaced by a weaker condition, characteristic of methods for ordinary differential and fine meshes. In order to minimize interpolation errors, the boundaries of a fine-meshed domain should avoid regions of strong gradients of the interpolated variables [21]. In applications addressing evolution of natural atmospheric flows, such a constraint is rarely satisfied as the fine meshes are often spawned according to a particular focus of interest (e.g., weather prediction in a certain region) rather than to criteria based upon flow structure or estimates of local truncation errors. Since concerns are usually with complex nonlinear flows, the computational noise due to communication between meshes arises at the boundaries as well as in the interior of the overlapping domains. Because of the nonlinearity, such a noise may affect the flow realization, or, in extreme cases, even lead to computational instability. A common cure for these difficulties is a smoothing (filtering, dissipating) of the interpolated variables. In this context, the implementation of the monotone interpolator in a mesh refinement scheme has the apparent advantage of suppressing spurious noise without introducing the excessive diffusion characteristic of traditional smoothers.

For illustration, we present the results of the two *multiple-grid* simulations of the idealized, free-slip flow of an inviscid, nonrotating, density-stratified fluid past a three-dimensional hill for Froude number $\lesssim 0.5$ ($\mathrm{Fr} = U/NH$, where U, H, and N are the flow speed, height of the obstacle, and the Brunt–Väisälä frequency, respectively). Salient features of such flows include the separation and reversal of the lower upwind stream, and the formation of intense vertically oriented vortices on the lee side of the hill [22–24]. The only difference between the two simulations is in the interpolation procedure employed in the mesh refinement scheme. The first experiment utilizes the interpolator (17) implementing the FCT version of the fourth-order-accurate dissipative scheme [4] (this interpolator has been already considered in Section 4.1), and the second experiment utilizes the same interpolator but without the FCT monotonicity adjustment. Since a goal of this section is to demonstrate potential advantages of the monotone interpolators, we shall comment only briefly on other physical and numerical aspects of the experiments performed.

In general terms, the experiments discussed herein are similar to those reported in [23, 24]. The bell-shaped hill

$$z(x, y) = H\left[1 + \left(\frac{x - x_0}{L}\right)^2 + \left(\frac{y - y_0}{L}\right)^2\right]^{-3/2}, \quad (32)$$

with the horizontal scale $L = 25 \times 10^3$ m and height $H = 0.12L$, is placed in the uniform flow ($U = 5\ \mathrm{ms}^{-1}$) with $\mathrm{Fr} = 0.15$. The relevant, hydrostatic equations of motion are integrated using a version of the Pennsylvania State University/National Center for Atmospheric Research community model [25, 26]. The model equations are cast in Lambert conformal map coordinates and the normalized pressure (sigma) coordinate in the vertical. The finite-difference formulation of the model equations employs centered-in-time-and-space approximations and a split-explicit time integration scheme [27] for efficient treatment of the fast gravity modes. The boundary conditions incorporate the free-slip/rigid-lid and the free-surface assumptions at the bottom and top boundaries of the model, respectively. The sponge layer in the upper portion of the model absorbs vertically propagating gravity waves, and the relaxation scheme [28] is employed at lateral boundaries of the model domain. The two-way-interactive mesh refinement scheme allows for an arbitrary number of overlapping and translating rectangular grids aligned with the model coordinates; the mesh refinement ratio of the temporal and spatial grid increments is common for all meshes.

The current experimental setup employs three horizontally nested, stationary grids with the refinement ratio equal to 3. All three meshes are centered with respect to the hill's origin but the center of finest mesh is shifted downstream by $1.4L$. In the horizontal, the model domain (the coarsest mesh) covers $28.8L \times 28.8L$; the first and second nested grids cover the subdomains of $9.6L \times 9.6L$ and $3.2L \times 3.2L$, respectively. All three meshes are resolved with 48×48 grid intervals. The vertical domain (common to all three meshes) is covered with 16 sigma layers of interval $\Delta\sigma$ varying from 0.05 in the lower part of the model to 0.1 at the top which coincides with $\sim 3.5H$. The gravity wave absorber fills the top $\sim 1.2H$ portion of the model domain.

The characteristic features of the flow (lee vortices and upwind separation) are established after dimensionless time $T \equiv tU/L \approx 2$ following the initialization. Figures 2 and 3 compare the surface flows from the two experiments after $T \approx 4.5$, when the flow is approximately steady (only the results from the two inner meshes are displayed). Insofar as the gross features of the flow are concerned (Figs. 2a and 3a), the two solutions are similar, and they are both in qualitative agreement with the results reported in [23, 24]. Inasmuch as the two simulations employ slightly different numerics, it is not surprising that the two realizations of the flow differ in details (Figs. 2b and 3b). Although it is difficult to assess a physical significance of the differences, the advantages of using monotone interpolators in the mesh refinement scheme are apparent. In the simulation with an oscillatory interpolator (Fig. 2), there is a clear notion of a noise development at the inflow/outflow boundaries of the finest mesh (Fig. 2b), whereas the flow simulated with a monotone interpolator passes through the boundaries in a smooth and orderly fashion. The analysis of the pressure fields' histories (not shown) indicates that the noise amplifies with time in the simulation with the oscillatory interpolator. This tendency of growth indicates potential "trouble" in a longer time simulation and a need for some

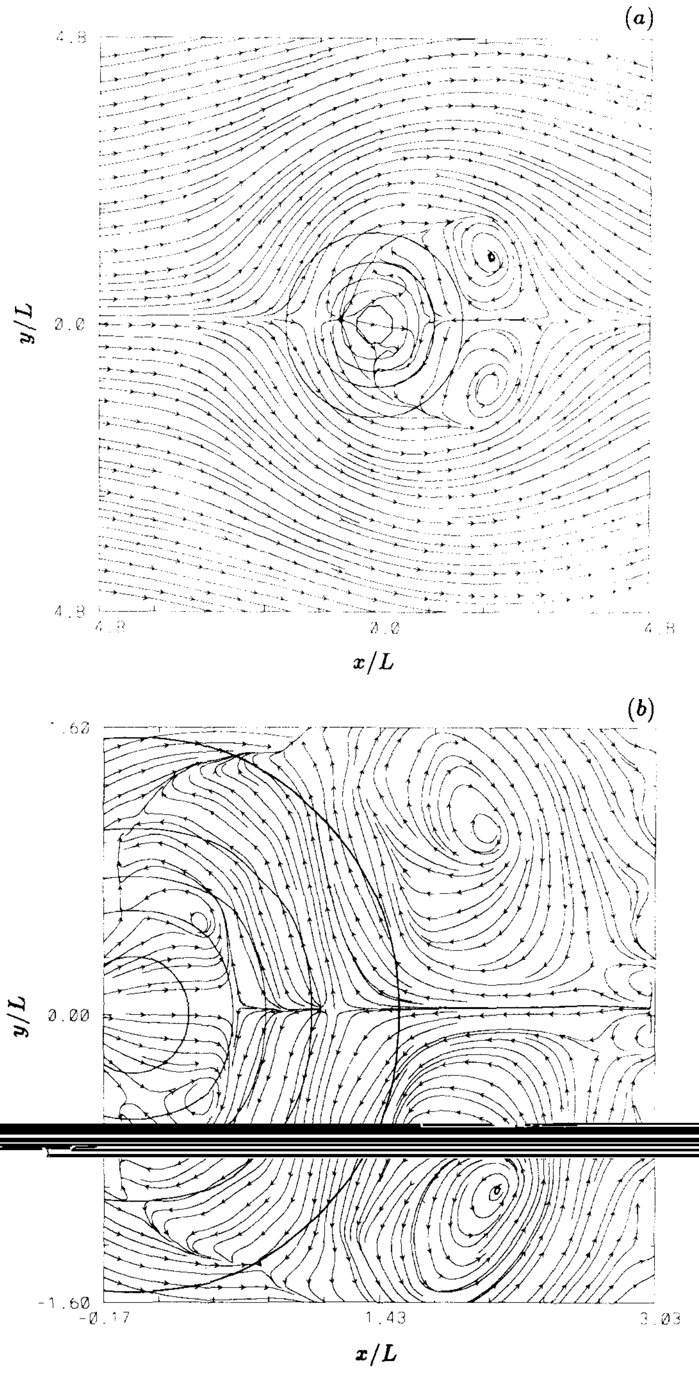

FIG. 2. Quasi-steady surface streamlines for the experiment with low Froude number flow past a bell-shaped hill using an oscillatory interpolator in the mesh refinement scheme. The two panels show the flows within the first and the second nested subdomains of the model. The contour interval of the obstacle's height is $\frac{1}{6}H$.

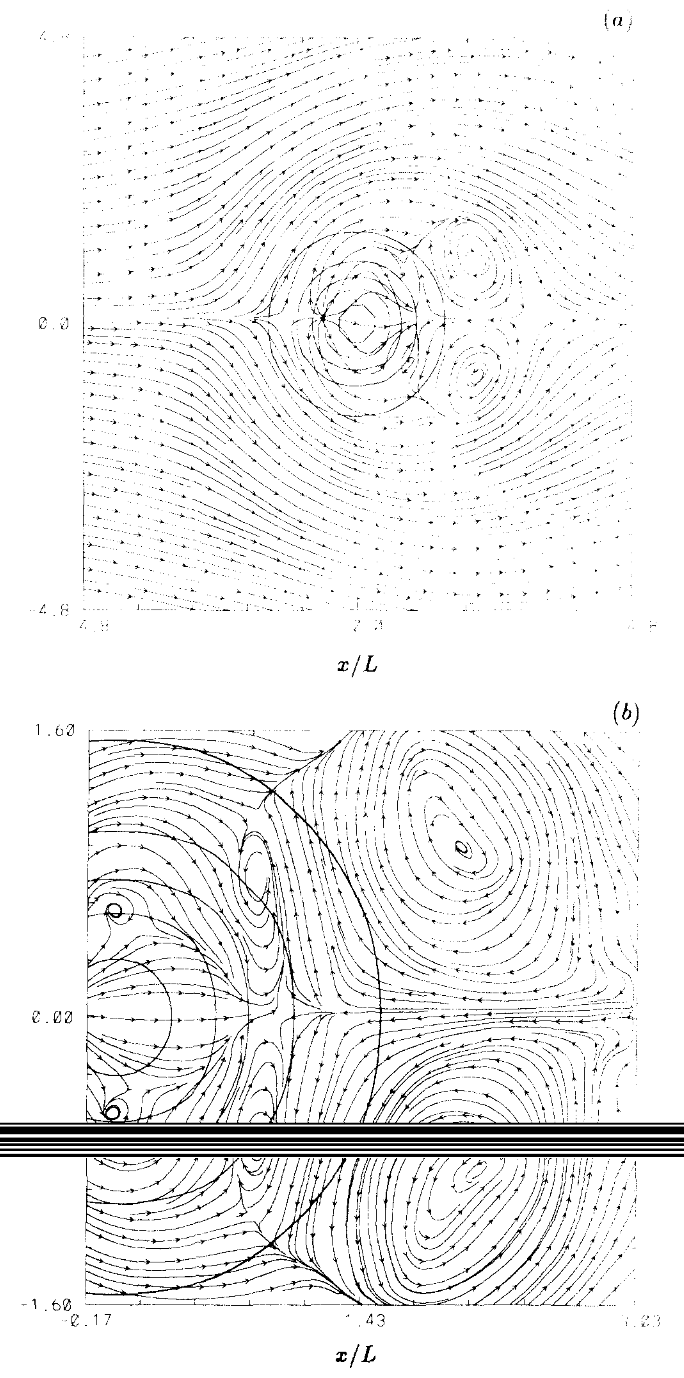

FIG. 3. As in Fig. 2 but using a monotone interpolator in the mesh refinement scheme.

additional smoothing in the mesh refinement scheme. Since the degree of smoothing required is, in general, problem dependent and unknown a priori, the implementation of the monotone interpolator in the mesh refinement scheme is a preferable alternative.

The advantages of the monotone interpolators become even more apparent in applications addressing natural atmospheric flows. Therein, the error growth depends on a variety of physical parameterizations (e.g., water substance phase-change processes), some of which are very sensitive to small-scale disturbances. In such applications, traditional smoothers exhibit a tendency for excessive smearing of many of the realistic features propagating into or through the interacting meshes.

5. SUMMARY

The practice of computational fluid dynamics often requires accurate as well as nonoscillatory interpolation procedures. Such procedures, often referred to as shape-preserving interpolation, may be employed to design monotone advection transport algorithms [1]. This paper poses an inverse problem. A variety of monotone advection schemes with attractive properties has been developed over the last two decades abstracting from any arguments that invoke explicit interpolation procedure [5, 6]. Our goal is to provide a formalism allowing the exploitation of these advection algorithms as shape-preserving interpolators. The central theoretical issue concerns a formal equivalence of the advection and interpolation operators on discrete meshes. The general idea of such an equivalence has often been referred to in the rezoning (remapping) stage of ALE models for fluids [10, 11]; however, the arguments invoked were partially intuitive. Releasing the exact conservation constraint of the ALE models simplifies the theoretical and practical aspects of the equivalence. Through elementary arguments exploiting either the Stokes theorem or the untruncated Taylor formula at zeroth order of expansion, one may show that the solution to an interpolation problem can be expressed as a formal integral of the advection equation (1). As a consequence, the interpolating operator on a discrete mesh may be represented by an advection scheme (formula (17)), in which the local Courant number vector is replaced by the normalized displacement between a grid point and a point of interest to the interpolation procedure. The accuracy of the resulting interpolation scheme is that of the advection scheme employed.

Among a variety of available advection schemes, only the dissipative (forward-in-time) algorithms are suitable for practical applications. Since the effective velocity field is constant for each point of interest to the interpolation procedure, these advection schemes retain their formal accuracy of the constant coefficients limit. Constancy of the effective velocity in an arbitrary-dimensional problem allows for a straightforward alternate-direction implementation of one-dimensional advection schemes without introducing errors characteristic of the time-split advection procedures in variable flows. The exactness of the time-splitting may be also shown by means of more general arguments invoking Stokes' theorem.

Insofar as the linear dissipative advection schemes are concerned, there is no particular gain from such an exercise, as the resulting interpolators may be alternately derived with the help of more traditional arguments invoking either the truncated Taylor formula or Lagrangian polynomial fitting [4]. However, when the preservation of monotonicity and/or sign of the interpolated variable is essential, then the interpolation formula (17) becomes a useful tool. For instance, a sign-preserving second-order-accurate interpolation may be easily achieved using a simple scheme [16], whereas a variety of monotone (and sign-preserving) interpolation schemes of different overall accuracy and complexity levels may be generated using FCT versions of dissipative schemes [4]. An efficient implementation of the FCT advection schemes requires minor adjustments (simplifications) in the traditional formulae.

The identified class of monotone interpolation schemes offers simple, powerful, and convenient tools for those applications where the exact conservation constraint is not essential. The examples of applications to selected, diverse problems of atmospheric fluid dynamics demonstrate the utility of the advocated approach.

ACKNOWLEDGMENTS

The authors are grateful to Len Margolin and William Skamarock for their personal reviews of the manuscript. The discussions with Tzvi Gal-Chen are also gratefully acknowledged. Special thanks go to Hope Hamilton for her editorial assistance.

REFERENCES

1. P. J. Rasch and D. L. Williamson, *SIAM J. Sci. Stat. Comput.* **11**, 656 (1990).
2. C. E. Leith, in *Methods in Computational Physics*, Vol. 4, edited by B. Alder, S. Ferenbach, and M. Rotenberg (Academic Press, New York, 1965).
3. W. P. Crowley, *Mon. Weather Rev.* **96**, 1 (1968).
4. C. J. Tremback, J. Powell, W. R. Cotton, and R. A. Pielke, *Mon. Weather Rev.* **115**, 540 (1987).
5. P. K. Sweby, *SIAM J. Numer. Anal.* **21**, 995 (1984).
6. S. T. Zalesak, in *Advances in Computer Methods for Partial Differential Equations*, Vol. VI, edited by R. Vichnevetsky and R. Stepleman, Publ. IMACS (Baltzer, Basel, 1987).
7. J. P. Boris and D. L. Book, *J. Comput. Phys.* **11**, 38 (1973).
8. S. T. Zalesak, *J. Comput. Phys.* **31**, 335 (1979).
9. P. K. Smolarkiewicz and W. W. Grabowski, *J. Comput. Phys.* **86**, 355 (1990).
10. J. K. Dukowicz, *J. Comput. Phys.* **54**, 411 (1984).
11. J. K. Dukowicz and J. W. Kodis, *SIAM J. Sci. Stat. Comput.* **8**, 305 (1987).
12. J. K. Dukowicz, Los Alamos National Laboratory, Los Alamos, NM 87545, personal communication (1990).
13. P. K. Smolarkiewicz and P. J. Rasch, *J. Atmos. Sci.* **48**, 793 (1991).
14. K. Maurin, *Analysis Part I: Elements* (Reidel, Boston, 1976).
15. R. D. Richtmyer and K. W. Morton, *Finite Difference Methods for Initial Value Problems* (Wiley-Interscience, New York, 1967).
16. P. K. Smolarkiewicz, *Mon. Weather Rev.* **111**, 479 (1983).
17. P. K. Smolarkiewicz, *J. Comput. Phys.* **54**, 325 (1984).
18. P. K. Smolarkiewicz and T. L. Clark, *J. Comput. Phys.* **67**, 396 (1986).
19. L. G. Margolin and P. K. Smolarkiewicz, Lawrence Livermore National Laboratory Report No. UCID-21866, 1989 (unpublished).
20. H. C. Kuo and R. T. Williams, *Mon. Weather Rev.* **118**, 1278 (1990).

21. W. Skamarock, J. Oliger, and R. Street, *J. Comput. Phys.* **80**, 27 (1989).

22. C. R. Hunt and W. H. Snyder, *J. Fluid Mech.* **96**, 671 (1980).

23. P. K. Smolarkiewicz and R. Rotunno, *J. Atmos. Sci.* **46**, 1154 (1989).

24. P. K. Smolarkiewicz and R. Rotunno, *J. Atmos. Sci.* **47**, 1498 (1990).

25. R. A. Anthes and T. T. Warner, *Mon. Weather Rev.* **106**, 1045 (1978).

26. R. A. Anthes, E. Y. Hsie, and Y. H. Kuo, National Center for Atmospheric Research Technical Note No. NCAR/TN-282 + STR. 1987 (unpublished).

27. R. Madala, in *Finite-Difference Techniques for Vectorized Fluid Dynamics Calculations*, edited by D. Book (Springer-Verlag, New York, 1981).

28. D. J. Perkey and C. W. Kreitzberg, *Mon. Weather Rev.* **104**, 744 (1976).