4-2002

Dynamic Grid Adaptation Using the MPDATA Scheme

John P. Iselin
Bucknell University

Joseph M. Prusa
Iowa State University

William J. Gutowski
Iowa State University, gutowski@iastate.edu

Dynamic Grid Adaptation Using the MPDATA Scheme

Abstract

A dynamic grid adaptation (DGA) scheme is developed using various combinations of the multidimensional positive definite advection transport algorithm (MPDATA) to show the applicability of DGA with the MPDATA scheme to solve advection problems. A one-dimensional model is used to show the effects of varying the number of grid points and the parameters that control the grid redistribution scheme, to determine a stability criteria for the scheme and to investigate the effect of several MPDATA options. A two-dimensional model is used to show the applicability of the scheme in multiple dimensions and to illustrate the effects of DGA in combination with MPDATA options. Diffusion errors are reduced by more than 90% using DGA when compared to static, uniformly spaced grid computations. Phase errors are reduced using certain MPDATA options by more than 25%.

Disciplines

Atmospheric Sciences | Climate | Mechanical Engineering

Comments

This article is from *Mon. Wea. Rev.*, **130**, 1026–1039. doi: http://dx.doi.org/10.1175/1520-0493(2002)130<1026:DGAUTM>2.0.CO;2. Posted with permission.

Dynamic Grid Adaptation Using the MPDATA Scheme

JOHN P. ISELIN

Department of Mechanical Engineering, Bucknell University, Lewisburg, Pennsylvania

JOSEPH M. PRUSA

Department of Mechanical Engineering, Iowa State University, Ames, Iowa

WILLIAM J. GUTOWSKI

Department of Geological and Atmospheric Sciences and Department of Agronomy, Iowa State University, Ames, Iowa

(Manuscript received 16 February 2001, in final form 10 September 2001)

ABSTRACT

A dynamic grid adaptation (DGA) scheme is developed using various combinations of the multidimensional positive definite advection transport algorithm (MPDATA) to show the applicability of DGA with the MPDATA scheme to solve advection problems. A one-dimensional model is used to show the effects of varying the number of grid points and the parameters that control the grid redistribution scheme, to determine a stability criteria for the scheme and to investigate the effect of several MPDATA options. A two-dimensional model is used to show the applicability of the scheme in multiple dimensions and to illustrate the effects of DGA in combination with MPDATA options. Diffusion errors are reduced by more than 90% using DGA when compared to static, uniformly spaced grid computations. Phase errors are reduced using certain MPDATA options by more than 25%.

1. Introduction

Widely varying spatial and temporal scales and the nonlinearity in atmospheric flows present significant modeling challenges. Large-scale features drive smaller-scale disturbances like fronts, thunderstorms, and tornadoes whose influence can cascade back up to larger scales. Fortunately, since small-scale phenomena are frequently distributed nonuniformly throughout the spatial domain it is not necessary to have a fine computational grid over the entire domain, and is an inefficient use of computational resources to do so. Dynamic grid adaptation (DGA) is one of several techniques that permit refining a computational grid in select spatial regions.

In this work the multidimensional positive definite advection algorithm (MPDATA) of Smolarkiewicz and Margolin (1998) is used in combination with the grid redistribution technique of Brackbill and Saltzman (1982). An extended Courant–Friedrichs–Lewy (CFL) stability criteria that is applicable to DGA schemes is developed and used to show that although the problem is cast from an Eulerian perspective the use of DGA yields a Lagrangian aspect to the numerical technique. One- and two-dimensional test models are used to show under what combinations of Courant number and grid resolution DGA is appropriate.

DGA is the modification of a computational grid in response to computed characteristics of the flow field in order to reduce the truncation error of the numerical scheme. Adaptive grid strategies can be split into gridpoint redistribution schemes and local grid refinement schemes (Kim and Thompson 1990). Gridpoint redistribution, which is the technique used in this work, allows a fixed number of grid points to move continuously during the simulation to increase the gridpoint density in regions where numerical error would otherwise be high and subsequently lower the gridpoint density in regions where the numerical error is low. In contrast, local grid refinement schemes insert and extract grid points into a static grid to increase the grid density in some regions and reduce it in others to achieve the reduction of truncation error.

DGA techniques have been used extensively in aerospace applications to resolve discontinuities in flow properties due to shock waves, slipstreams, and contact discontinuities (Hawken et al. 1991). Their use has been more limited in atmospheric modeling. Dietachmayer and Droegemeier (1992) used the variational approach of Brackbill and Saltzman (1982) to solve the viscous Burger's equation in one dimension. In two dimensions

Corresponding author address: Dr. John P. Iselin, Dept. of Mechanical Engineering, Bucknell University, Lewisburg, PA 17837.
E-mail: iselin@bucknell.edu

a frontogenesis problem, solid-body rotation of four cones, and the evolution of a buoyant thermal were calculated. This work was extended by Fiedler and Trapp (1993). Two- and three-dimensional buoyant thermals were computed in a stably stratified fluid. Prusa et al. (1996, 2001) used a coordinate mapping technique in a mesoscale model to move grid points in the simulation of propagation and breaking of high-altitude gravity waves. More recently, Smolarkiewicz and Prusa (2002) have generalized this mapping technique to a global model. Tomlin et al. (1997) used unstructured dynamic grids to simulate atmospheric reaction flow problems. Behrens (1996) used an unstructured grid with local grid refinement and a semi-Lagrangian approach in a parallel computing environment to compute the rotating slotted cylinder about a point (Zalesak 1979). More recently Bacon et al. (2000) have developed the operational Multiscale Environmental Model with Grid Adaptivity (OMEGA), which is an unstructured grid model that uses the gridpoint insertion/extraction technique.

Moving nested grids (Berger and Oliger 1984) is a combination of dynamic grid adaptation and local grid refinement. Finer Cartesian grids overlay coarser Cartesian grids. The nested grids move, change size, and are added or removed as directed by the attributes of the flow. Care must be taken in these models to deal with abrupt changes in grid resolution at the boundaries of the nested grids. Conservative interpolation is needed in order to establish the boundary conditions for the finer domains. Flow features of the finer grids are not allowed to propagate onto the coarser grids. DeMaria et al. (1992) used a nested finite-element model to perform hurricane tracking during the 1989 and 1990 hurricane seasons. Skamarock and Klemp (1993) used a moving nested grid as a form of dynamic grid adaptation. A numerical algorithm controlled the movement of the nested grids in two-dimensional simulations. Skamarock et al. (1994) used this method to study the three-dimensional evolution of long-lived squall lines. However, the movement of the grids was partially directed through user intervention. Finley et al. (1998a,b) used moving nested grids to model tornadoes. Although the nested grids had the ability to move, user interaction was required during the simulation to direct the grid movement.

The advantages of DGA over nested Cartesian grids are (i) no abrupt changes in grid resolution, (ii) smaller-scale features of the flow resolved in the area of grid clustering can influence the larger scales as occurs in nature, and (iii) the gridpoint movement is completely automated. However, it has yet to be shown whether DGA techniques can be made computationally competitive with nested grid techniques.

Although DGA can be implemented with other numerical schemes, MPDATA is particularly well suited to DGA because it is a positive-definite, second-order scheme that requires the velocities to be approximated at the half time step and is easily made monotone through flux-corrected transport. Many second-order schemes are non–sign preserving and not monotone due to their dispersive nature. The DGA technique clusters grid points in areas where high numerical errors occur, which are typically areas with high slopes and curvatures in the simulated field. Since the dispersive ripples created by most second-order schemes create false areas of high curvature, DGA has a tendency to move grid points away from physically realistic features and cluster points around these erroneous features. The sign-preserving nature of MPDATA eliminates spurious oscillations as long as the background of the advected field is zero, which is the case for many transport problems. If the background field is not zero the flux limiting method of Boris and Book (1973) is easily implemented, making the scheme monotone. Therefore, difficulties encountered while implementing most second-order schemes are eliminated when using MPDATA. Additionally, MPDATA is advantageous because the fluid velocity is evaluated at the half time step. When implementing DGA it is necessary to calculate the velocity of the computational grid point as it moves. In order to maintain the second-order accuracy of the scheme this velocity must be known to second order. Since the MPDATA scheme requires the velocity at the $n + 1/2$ time step a second-order approximation is easily obtained using a central difference between time steps n and $n + 1$.

During the course of this study one- and two-dimensional advection models were developed. The one-dimensional model, discussed in section 2, was used to examine the applicability of various MPDATA options in a DGA environment, to determine the effect of varying the number of grid points in a DGA–MPDATA model, and to investigate the effect of gridpoint movement on the stability criteria of the MPDATA scheme. The two-dimensional model, discussed in section 3, was used to investigate the applicability of DGA and MPDATA in a multidimensional case and to examine the effects of the various MPDATA options. Conclusions are drawn in section 4.

2. The one-dimensional model

A one-dimensional model was developed that simulated the passive advection of a Gaussian pulse by a constant velocity flow. Since a one-dimensional model required much less time to perform a calculation than a multidimensional model it was used to investigate the effects of varying the number of grid points and implementing the various MPDATA options. It was found that for the gridpoint redistribution scheme used, the gridpoint velocity could become so high the CFL condition was violated. For the one-dimensional case, a constraint on the gridpoint movement was developed that gives insight into how and why dynamic adaptive grids work and why they are useful. Sections 2a–e de-

scribe the MPDATA scheme in a one-dimensional DGA setting, the one-dimensional gridpoint redistribution scheme, the extended CFL stability criteria for DGA schemes, the specifics of the advection case, and the results of several numerical experiments, respectively.

a. MPDATA in one dimension

The MPDATA scheme uses the donor-cell scheme to obtain a first-order approximation at a new time level. Corrections to this are made by further applications of the donor-cell scheme using a pseudovelocity that is specifically designed to approximate the truncation error of the previous approximations. Therefore, the scheme is iterative in nature.

Transforming the one-dimensional advection equation from the physical space $q = q(x, t)$ into an integer computational space $q = q(\xi, \tau)$, where $t = \tau$ and $x = f(\tau, \xi)$, yields

$$\frac{\partial q}{\partial \tau} + \hat{u}\frac{\partial q}{\partial \xi} = 0, \tag{1}$$

where

$$\hat{u} = \left(\frac{\partial x}{\partial \xi}\right)^{-1}\left(u - \frac{\partial x}{\partial \tau}\right). \tag{2}$$

It is apparent that the velocity on the computational grid is simply the physical velocity less the speed of the grid point multiplied by a stretching factor $(\partial x/\partial \xi)^{-1}$. Note that even if the velocity in the physical problem is constant the velocity in the computational domain $\hat{u}$ will in general be variable since $\partial x/\partial \tau$ is a function of space and time.

In order to apply the MPDATA scheme the advection term in Eq. (1) must be in flux form, and since the computational velocity $\hat{u}$ is not constant with respect to space, Eq. (1) becomes

$$\frac{\partial q}{\partial \tau} + \frac{\partial(\hat{u}q)}{\partial \xi} = R, \tag{3}$$

where $R = q(\partial \hat{u}/\partial \xi)$ in this case.

MPDATA is developed by expanding the temporal and spatial finite differences using Taylor series, back substituting approximations of the governing equation to replace the time derivatives in the higher-order terms with spatial derivatives, and using a the donor-cell scheme to approximate the spatial derivative. This is developed in detail in Smolarkiewicz and Margolin (1998). This particular form of R yields a pseudovelocity:

$$\hat{u}^{(1)} \equiv \frac{1}{2}(\Delta\xi|\hat{u}| - \Delta\tau\hat{u}^2)\frac{1}{q}\frac{\partial q}{\partial \xi}. \tag{4}$$

This velocity has no physical meaning but is derived such that when used with the donor-cell scheme and subtracted from the first-order solution the result is a second-order solution. For a general R there is a $q(\partial \hat{u}/\partial \xi)$ term and a $-R$ term that appears in the pseudovelocity. However, in this particular case they identically cancel, meaning that this divergence term R only needs to be included in the first donor-cell iteration.

The MPDATA scheme can be extended to an arbitrary number of K iterations:

$$q^{(k+1)} = q_i^{(k)} - \{F[q_i^{(k)}, q_{i+1}^{(k)}, \hat{U}_{i+1/2}^{(k)}] - F[q_{i-1}^{(k)}, q_i^{(k)}, \hat{U}_{i-1/2}^{(k)}]\} + R_i^n \delta_{(k)0}$$
$$k = 0, 1, 2, \ldots, K, \tag{5}$$

where $q^{(0)} = q_i^n$, $\hat{U}^{(0)} = \hat{U} = (\Delta\tau/\Delta\xi)\hat{u}^n$, $q^{n+1} = q^{(K)}$, $\delta_{(k)0}$ is the Kronecker delta, and F is the donor cell flux:

$$F(q_L, q_R, U) = \max(U, 0)q_L - \min(U, 0)q_R. \tag{6}$$

The pseudovelocities at each iteration are defined as

$$\hat{U}^{(k)} = [|\hat{U}^{(k-1)}| - \hat{U}^{(k-1)}\hat{U}^{(k-1)}]\frac{\Delta\xi}{2q^{(k)}}\frac{\partial q^{(k)}}{\partial \xi}. \tag{7}$$

When $K = 1$, this collapses to the donor-cell scheme; $K = 2$ is the most basic MPDATA scheme. If additional iterations are taken ($K > 2$), the error decreases, but the scheme remains second-order accurate.

Smolarkiewicz and Margolin (1998) discuss several different options to the MPDATA scheme that are included in this work. The ''third-'' order accurate scheme includes corrections that make the scheme third-order accurate in the absence of velocity gradients. Although this scheme is still second-order accurate in the presence of velocity gradients, it has the effect of distributing the error in a more symmetric manner than schemes without these added terms. Margolin and Smolarkiewicz (1989) also develop the recursive pseudovelocity (RPV) option, which is a strictly two-pass scheme where the pseudovelocity of the second pass is derived using the summation of an infinite number of MPDATA iterations. A detailed explanation of these options in combination with DGA can be found in Iselin (1999).

b. One-dimensional gridpoint redistribution

Ideally, given a numerical method, the lowest-order error terms would be derived, assuming an uneven and time-dependent mesh. These error terms could then serve as a weight function that when equally distributed over the domain would result in a set of equations, which, when solved, would yield a desirable distribution of grid points. Analysis of a one-dimensional passive advection problem with centered space and time discretizations yields 13 unique leading error terms that are in general nonlinear. Due to the impracticality of using all of these terms and the inability of determining the most important ones, a more heuristic measure of the error is used than the truncation error of the scheme.

When using DGA, it is not crucial to locate grid points in some precise location as long as reasonable resolution

occurs in the areas of interest. Thus as long as the model developer makes reasonable decisions on the grid redistribution criteria, dramatic improvements in numerical solutions are possible without increasing the number of grid points. There are several parameters required for the techniques described in this section and in section 3c that must be set by the user. However, it was found that the numerical performance with DGA is quite robust with regard to the precise values of these parameters. Reasonable performance is almost guaranteed as long as grid points cluster, no region is left void of grid points, and gridpoint movement is not too fast.

Following Dietachmayer and Droegemeier (1992) the sum of the absolute values of the first and second derivatives of the prognostic variable was used to construct the weighting function:

$$\hat{w} = \left|\frac{\partial q}{\partial x}\right| + L\left|\frac{\partial^2 q}{\partial x^2}\right|,$$

where L is a characteristic scale. In our simulations, we chose this scale to be the domain size. While other choices are possible (e.g., the integral scale for a turbulence) and may even be preferred in a specific application, they are problem specific and less straightforward to implement. A disadvantage of our choice for L was it often resulted in the second derivative dominating the value of the weighting function.

The weight function was based solely on the prognostic field variable at the old time step n. Since it was used to integrate forward in time from time step n to $n + 1$, the weight function was smoothed using a 1–2–1 filter to spread the influence of the gridpoint clustering out in front of the propagating phenomenon. Both the dependent and the independent variables were scaled between 0 and 1 so the scheme would work independently of the problem size.

In order to counter the dominance of the second derivative previously mentioned the scaled weight function was raised to the 1/4 power. This had a tendency to compress the local maxima closer to one, decreasing this dominance. To avoid the possibility that the weight function vanishes and thus a region void of grid points forms, an arbitrary positive value of one was added to the weight function. Thus the weight function was defined as

$$w_i = 1 + \lambda \hat{w}_i^{1/4}, \tag{8}$$

where λ is the grid stretching parameter specified by the user to control the extent of grid adaptation and $\hat{w}_i$ is the value of the smoothed and scaled weight function at grid point i.

Using calculus of variations Thompson et al. (1985) present a method of grid generation based on the minimization of a weighted gridpoint density. The global measure of the gridpoint density is

$$I = \int \frac{(\xi_x)^2}{w}\, dx, \tag{9}$$

where ξ_x represents the derivative of ξ and w is a function of x. Because ξ and w were chosen as functions of x the problem was solved without iterating. Integration of the resulting Euler–Lagrange equation for Eq. (9) equally distributes the weighted grid points over the domain:

$$\frac{d\xi}{dx} = \frac{w}{c}. \tag{10}$$

Equation (10) was solved by the use of numerical integration over the entire domain and $\hat{I} - 2$ one-dimensional interpolations, where $\hat{I}$ is the number of grid points in the domain.

c. The extended CFL condition for DGA methods

When a grid redistribution DGA scheme is used in conjunction with an explicit numerical scheme, violations of the CFL condition can occur if the gridpoint velocity relative to the flow is too large. This is evident from Eq. (2) where the apparent velocity is proportional to the grid speed. Although this problem was only encountered in the one-dimensional model it can occur in higher-dimensional models if the gridpoint velocity, relative to the fluid velocity, is large enough.

Yanenko et al. (1979) includes a Lagrangian transport term in the grid stretching criteria and Thompson et al. (1985) refer to grid points as ''observers'' that follow the flow. Therefore, others have had the conceptual idea that the DGA technique is a blend between Eulerian and Lagrangian references. The following analysis explicitly presents the Lagrangian aspect of the DGA technique.

By using Eq. (2) at the half time step and cell boundary the traditional CFL condition is generalized to an extended CFL stability condition:

$$\left|\frac{\Delta\tau}{\Delta\xi}\left(u - \frac{\partial x}{\partial \tau}\right)\frac{\partial \xi}{\partial x}\right| \leq 1. \tag{11}$$

Substituting a central difference about $x^{n+1/2}$ for $\partial x/\partial\tau$, noting that $\Delta\xi = 1$, and solving for x^{n+1} yields an expression for locations of x^{n+1} that will satisfy the extended CFL condition:

$$(x^n + \Delta\tau u) - \frac{\partial x}{\partial \xi} \leq x^{n+1} \leq (x^n + \Delta\tau u) + \frac{\partial x}{\partial \xi}. \tag{12}$$

This expression illustrates that there is a bounded region in which a grid point can move. The expression in parentheses is the first-order Lagrangian transport of the grid point by the fluid. Had a higher-order finite-difference approximation for $\partial x/\partial t$ been used this bracketed expression would be higher order. The width of the region is determined by the metric term $\partial x/\partial\xi$. If the grid point is not allowed to move, the expression reverts back to the typical expression for the CFL condition:

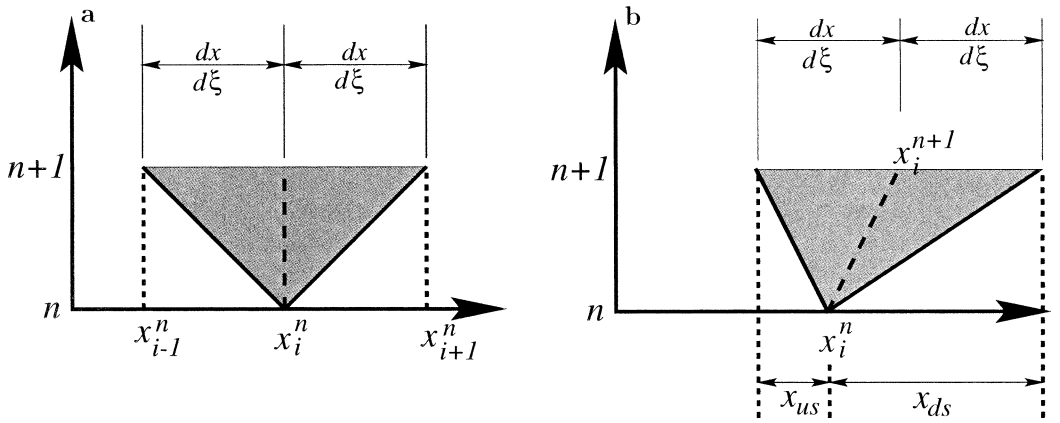

FIG. 1. (a) Zone of stability for a static grid; (b) zone of stability for a dynamic grid. If the advected distance $u\Delta t$ remains within the shaded cone, the scheme is stable. In (b) the x_{us} is the advection distance if the $u < 0$, and x_{ds} is the distance if $u > 0$.

$$\left|\Delta\tau\frac{\partial\xi}{\partial x}u\right| \leq 1, \tag{13}$$

where an approximation to $\partial\xi/\partial x$ is $1/\Delta x$.

Figure 1 illustrates the zone of stability for a static grid and one where the grid point is advected with the flow. If the distance a fluid parcel travels over a time step is outside this shaded region, the CFL condition is violated and an explicit scheme, like MPDATA, will become unstable. Note that in Fig. 1b this advection distance is not equal for upstream and downstream directions. If the grid point motion (to the right in Fig. 1) is in the same direction as the fluid motion, larger time steps, corresponding to larger Courant numbers, can be taken because the zone of stability is tilted in that direction. This downstream distance is indicated as x_{ds} in Fig. 1. However, if the gridpoint motion opposes the flow direction (to the left in Fig. 1), smaller time steps are required, since the upper corner of the zone tilts away from the flow direction. This upstream distance is indicated by x_{us}. If the gridpoint motion was completely determined by the flow advection, infinitely large time steps could be taken because the value of q at the grid point would be constant. Grid points are most free to take on this Lagrangian character only in the interior of the domain. Near the boundaries, the Eulerian character of the regular grid must manifest itself as grid points cannot cross the boundary of the domain.

When upstream motion of grid points is combined with close grid spacing the extended CFL condition is very restrictive. It was found that when the number of grid points in the one-dimensional model was less than 250, $\partial x/\partial\xi$ was large enough that with a reasonable time step the scheme was always stable. However, with larger numbers of grid points, $\partial x/\partial\xi$ became so small that almost any gridpoint movement caused a violation of the extended CFL condition.

Two different techniques that used Eq. (12) to ensure stability were attempted. The first technique limited the amount the grid points could move based on the most severe violation of the CFL condition. The second technique let the grid points move as directed by the grid generator and adjusted the time step so that the scheme would be stable. The first technique was successful while the second was not.

The first technique limited the movement of the grid points using the greatest violation of the CFL condition. This was done by calculating the maximum allowable movement of each grid point and identifying which exceeded this allowable movement by the greatest percentage. At each grid point, the fraction of the original movement allowed was calculated as

$$\phi_i = \min\left(\frac{\Delta x_{M_i}}{\Delta x_{A_i}}, 1\right), \tag{14}$$

where Δx_{M_i} is the maximum allowable distance so the CFL is not violated at point i and Δx_{A_i} is the unlimited distance at point i. To ensure that no grid points were allowed to cross, all of the gridpoint movements were limited by the smallest ϕ_i. Often a couple of iterations of the procedure were required to obtain an acceptable gridpoint redistribution.

The second method attempted to adjust the time step size in order to meet the CFL criteria simultaneously at all the grid points. The inequalities in Eq. (12) can be rearranged to yield a minimum and maximum allowable time step for stability. These minimum and maximum time steps were computed for each grid point. If the largest minimum time step was less than the shortest maximum time step, the technique worked. However, in practice, situations occurred where the minimum time step of one point was larger than the maximum time step of another point and it was not possible to simultaneously satisfy Eq. (12) for all grid points.

d. One-dimensional model description

The equation used to model the passive advection of the tracer is

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} = 0, \tag{15}$$

where q, u, t, and x represent a non-negative intensive tracer quantity, the fluid velocity, time, and space respectively. Equation (15) was solved on a domain $0 < x < 1$ using a scaled Gaussian pulse of the form

$$q(x) = \frac{1}{3\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x - 0.2}{0.03}\right)^2\right] \tag{16}$$

as an initial condition. The velocity field was set to a constant, $u = 1$. The integration continued until $t = 0.5$.

The locations of the cell boundaries were used as the physical points in the grid redistribution scheme rather than the cell centers since MPDATA requires the velocity values at cell boundaries. Additionally, the coordinates of the cell centers are simply the average of the cell boundary coordinates, whereas the opposite is not true. The first and the last cell boundaries were set to 0 and 1, respectively, and served as boundary conditions for the elliptic grid solver. Ghost cell centers were calculated by reflecting the adjacent cell center about the end cell boundary to provide locations for boundary values that could be transported into the domain.

The initial condition was created from an evenly spaced grid followed by 10 repeated applications of the grid generator and the initial condition (16).

The time step was held constant throughout the integration. It was determined as

$$\Delta t = \frac{s}{u} \min_{\forall i} \left(\left.\frac{dx}{d\xi}\right|_i^{n=0}\right), \tag{17}$$

where the metric term $dx/d\xi$ was approximated by a central difference expression and the s was a safety factor such that $0 < s \leq 1$. If the grid points were not permitted to move, this safety factor would be identical to the Courant number. However, since the grid points were permitted to move, the Courant number was based on the fluid velocity plus the grid speed.

The error in the numerical solution was evaluated using the L_2 error norm, which was defined as

$$E_{L_2} = \left\{\frac{1}{L}\int_0^L [\hat{q}(x, t) - q(x, t)]^2\, dx\right\}^{1/2}, \tag{18}$$

where $q(x, t)$ and $\hat{q}(x, t)$ are the numerical and analytical solutions, respectively. The trapezoidal rule was used to compute the L_2 norm.

e. One-dimensional experimental results

A series of computations was conducted using a static, uniform-resolution grid to test different combinations of either two iterations, three iterations, four iterations, or the RPV option, and the third-order terms of the MPDATA scheme. All variations without the third-order terms were second-order accurate. When the third-order-accurate terms were included, MPDATA was third-order accurate due to the constant velocity of this test case. The $\log_2$ of the L_2 error norm was used to evaluate the accuracy of the different schemes. As the number of grid points increases by multiples of two, this measure of the error drops by the order of the scheme. Our results agreed completely with the findings in Margolin and Smolarkiewicz (1989).

A set of computations using DGA were run while varying the MPDATA options. The percentage reduction of the L_2 error for a given combination of MPDATA options was defined as

$$\% \text{ reduction} = 100\frac{E_{\text{static}} - E_{\text{dynamic}}}{E_{\text{static}}}, \tag{19}$$

where E_{static} and E_{dynamic} are the L_2 errors associated the computations performed with the static, uniform-resolution grid and the dynamically adaptive grid, respectively. Figure 2 shows the percentage reduction of the L_2 error norm when the DGA technique is used compared to a static, evenly spaced grid. When a static, uniform-resolution grid is used and the fluid velocity is constant, the Courant number is a constant at all grid points. However, when the grid points are allowed to move, the Courant number is different at each grid point. The maximum Courant number based on the initial grid stretching and ignoring the grid speed was called the safety factor. When the safety factor was 0.8 or less, the error was reduced by 90% when the third-order correction terms are not used. The use of constant velocity field for these experiments had the following two effects. First, MPDATA with the third-order terms is actually third-order accurate in the absence of velocity gradients. Therefore, the improvements due to the DGA are less impressive in Fig. 2c than in the cases where the third-order terms were not used. Second, the advection equation is a linear equation when the velocity profile is linear and thus the numerical solution approaches the exact analytical solution as the Courant number approaches one. Naturally, in this limiting case the DGA results appear less impressive.

Figure 3a depicts effects on the error of the DGA scheme of the number of smoothing passes of the weight function used in the grid generation procedure. These results were obtained using the RPV option without any third-order terms but are representative of any combination of the MPDATA options with DGA. When a small number of grid points was used, the effect of increasing the number of smoothing passes was detrimental to the solution, because the smoothing passes declustered the grid points. By comparing Figs. 3a and 3b it is apparent that when greater numbers of grid points were used, the smoothing passes reduced the number of times the gridpoint movements had to be

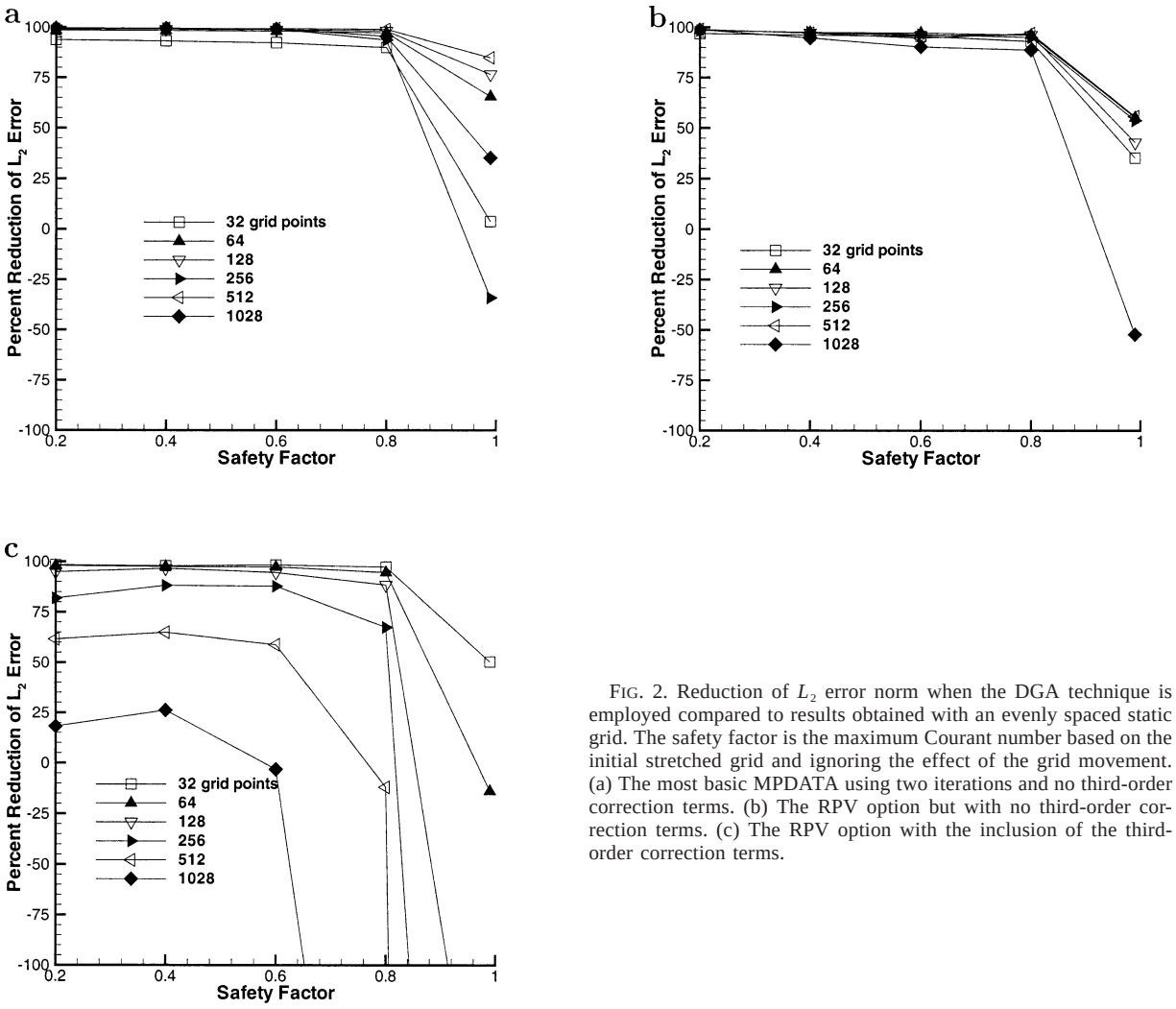

FIG. 2. Reduction of L_2 error norm when the DGA technique is employed compared to results obtained with an evenly spaced static grid. The safety factor is the maximum Courant number based on the initial stretched grid and ignoring the effect of the grid movement. (a) The most basic MPDATA using two iterations and no third-order correction terms. (b) The RPV option but with no third-order correction terms. (c) The RPV option with the inclusion of the third-order correction terms.

limited. Figure 3b shows the gridpoint limiting had a significant detrimental effect on the error. The best results were obtained when just enough smoothing passes were used to prevent the limiting of gridpoint movement.

Figure 4 compares the error as a function of computational cost for both codes. Although the DGA technique used fewer grid points, the equation for redistributing the grid points was solved at each time step. This added computational expense was nearly the computational savings that was achieved due to the method. When relatively few grid points were used, the slope of the curve in Fig. 4b was steeper (-5.1×10^{-7}) than the slope in Fig. 4a (-2.3×10^{-6}), indicating that when the number of grid points is small, increasing the number of points in the DGA code is more advantageous than adding grid points to the static grid code. However, as the grid points increase, the benefits diminish and eventually become liabilities. DGA exhibits 2.8-order accuracy for a small number of grid intervals ($16 \leq N \leq 64$), while the static grid exhibits 1.6-order accuracy for a small number of grid intervals ($32 \leq N \leq 128$). For large numbers of grid points ($N > 128$) these trends are reversed.

Note that these computational expenses are not representative of what would be expected from a full numerical weather prediction model using DGA. The percentage of CPU cycles taken by DGA in this case is significantly higher than would be taken in a full model. Although all of the DGA overhead is included in this case, only a hyperbolic advection equation is being solved. In a complete model an elliptic pressure equation would need to be solved in conjunction with hyperbolic equations for velocity components and tracer transport. Clearly the percentage overhead of DGA would be significantly reduced in this case. Even with this significant handicap, the DGA is competitive with the uniform static grid simulations.

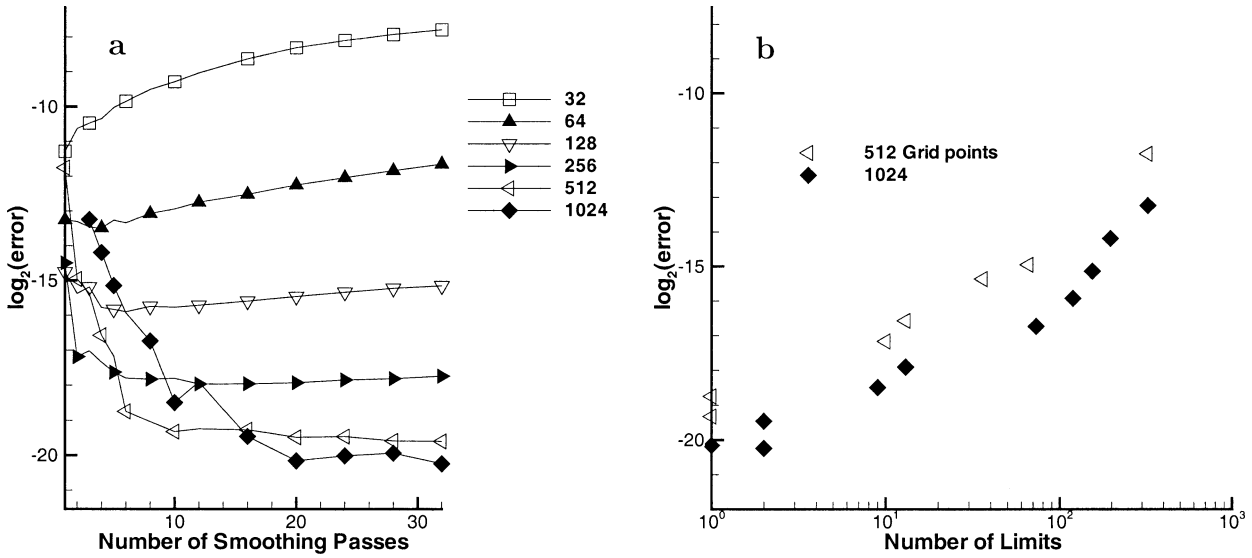

FIG. 3. The effects of limiting the gridpoint movement. The options used were RPV, no third-order terms, $\lambda_1 = 70$, and $\lambda_2 = 0$. The error used was the L_2 error. The legend indicates the number of grid points.

3. The two-dimensional model

A two-dimensional advection model was developed that solved the rotating cone problem. The purpose of this model was to evaluate the use of DGA–MPDATA in multiple dimensions. The results of the two-dimensional code more clearly illustrate the benefits of the different MPDATA options in combination with the DGA technique than does the one-dimensional model. Section 3a describes the rotating cone problem. Section 3b highlights the aspects of the MPDATA scheme in two dimensions that are not obvious extensions of the one-dimensional version. The grid redistribution scheme is significantly different in two dimensions than in one dimension. These differences are described in section 3c. In section 3d the results of several experiments are discussed.

a. Two-dimensional model description

The two-dimensional model was developed to simulate the advection of a passive tracer through the transformed equation:

$$\frac{\partial q}{\partial \tau} + \hat{u}\frac{\partial q}{\partial \xi} + \hat{v}\frac{\partial q}{\partial \eta} = 0, \tag{20}$$

where

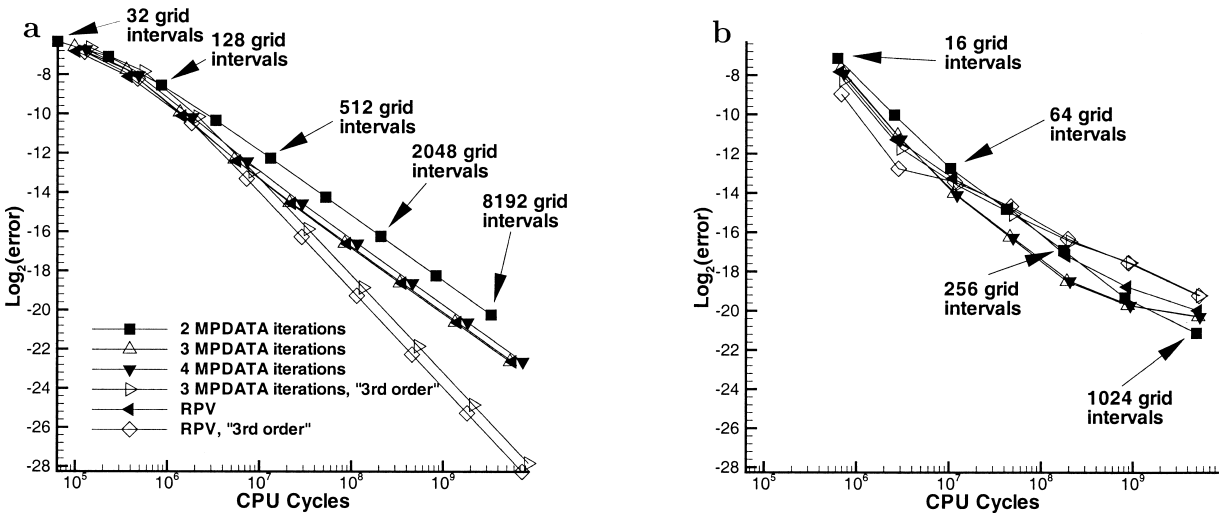

FIG. 4. The L_2 error vs CPU cycles when $\lambda = 70$ and safety factor = 0.6: (a) static uniform grid with 32, 64, 128, . . . , 8192 grid intervals and (b) dynamic grid, $\lambda = 70$ and 16, 32, 64, . . . , 1024 grid intervals.

$$\hat{u} = \frac{\partial \xi}{\partial t} + u\frac{\partial \xi}{\partial x} + v\frac{\partial \xi}{\partial y}, \tag{21}$$

$$\hat{v} = \frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y}, \quad \text{and} \tag{22}$$

$$\frac{\partial}{\partial t} = \frac{1}{J_2}\left[\frac{\partial}{\partial \tau} + \left(\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \tau} - \frac{\partial y}{\partial \eta}\frac{\partial x}{\partial \tau}\right)\frac{\partial}{\partial \xi} + \left(\frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \tau} - \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \tau}\right)\frac{\partial}{\partial \eta}\right],$$

$$\frac{\partial}{\partial x} = \frac{1}{J_2}\left(\frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta}\right),$$

$$\frac{\partial}{\partial y} = \frac{1}{J_2}\left(-\frac{\partial x}{\partial \eta}\frac{\partial}{\partial \xi} + \frac{\partial x}{\partial \xi}\frac{\partial}{\partial \eta}\right) \tag{23}$$

have been used to rewrite the two-dimensional version of Eq. (15) into transformed coordinates. The two-dimensional Jacobian is

$$J_2 = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}. \tag{24}$$

Equation (20) was solved for the two-dimensional rotating cone problem in a square bounded by $0 \le x \le 1$ and $0 \le y \le 1$. The value of the prognostic variable q was initialized as

$$q = \max\left[4 - \frac{4}{0.15}\sqrt{(x - 0.5)^2 + (y - 0.75)^2},\ 0\right]. \tag{25}$$

The velocity field was a temporally constant prescribed vortex:

$$u = -2\pi(y - 0.5) \qquad v = 2\pi(x - 0.5). \tag{26}$$

b. MPDATA in two dimensions

As with the one-dimensional case, MPDATA requires the equations to be in flux form. Following Eq. (3), Eq. (20) is rewritten as

$$\frac{\partial q}{\partial \tau} + \frac{\partial (q\hat{u})}{\partial x} + \frac{\partial (q\hat{v})}{\partial y} = R, \tag{27}$$

where the divergence term R is

$$R = q\left(\frac{\partial \hat{u}}{\partial \xi} + \frac{\partial \hat{v}}{\partial \eta}\right). \tag{28}$$

The major differences for the two-dimensional version of MPDATA are additional spatial terms that accounts for the orthogonal direction and the cross derivatives. Therefore, the pseudovelocity components for two-dimensional case are

$$\hat{U}^{(k+1)} = (|\hat{U}^{(k)}| - \hat{U}^{(k)}\hat{U}^{(k)})\frac{\Delta \xi}{2q^{(k)}}\frac{\partial q^{(k)}}{\partial \xi} - \hat{U}^{(k)}\hat{V}^{(k)}\frac{\Delta \eta}{2q^{(k)}}\frac{\partial q^{(k)}}{\partial \eta},$$

$$\hat{V}^{(k+1)} = (|\hat{V}^{(k)}| - \hat{V}^{(k)}\hat{V}^{(k)})\frac{\Delta \eta}{2q^{(k)}}\frac{\partial q^{(k)}}{\partial \eta} - \hat{U}^{(k)}\hat{V}^{(k)}\frac{\Delta \xi}{2q^{(k)}}\frac{\partial q^{(k)}}{\partial \xi}, \tag{29}$$

where

$$\hat{U}^{(k)} = \frac{\Delta \tau}{\Delta \xi}\hat{u}^{(k)}, \qquad \hat{V}^{(k)} = \frac{\Delta \tau}{\Delta \eta}\hat{u}^{(k)}, \tag{30}$$

When $k = 0$, the physical velocity is used; otherwise, k represents the iteration level. The spatial grid location for these calculations is $(i + 1/2, j)$ and $(i, j + 1/2)$ for the $\hat{u}$ and $\hat{v}$ pseudovelocities, respectively. MPDATA algorithm in two dimensions for a moving reference frame becomes

$$q_i^{(k+1)} = q_i^{(k)} - \{F[q_{i,j}^{(k)}, q_{i+1,j}^{(k)}, \hat{U}_{i+1/2,j}^{(k)}] - F[q_{i-1,j}^{(k)}, q_{i,j}^{(k)}, \hat{U}_{i-1/2,j}^{(k)}]\} - \{F[q_{i,j}^{(k)}, q_{i,j+1}^{(k)}, \hat{V}_{i,j+1/2}^{(k)}] - F[q_{i,j-1}^{(k)}, q_{i,j}^{(k)}, \hat{V}_{i,j-1/2}^{(k)}]\} + R_i^n \delta_{(k)0}$$

$$\text{where } k = 0, \ldots, K, \tag{31}$$

where $q^{(0)} = q^n$ and $q^{n+1} = q^{(K)}$.

c. Two-dimensional gridpoint redistribution

Following the one-dimensional case, the weight function was defined using the first and second derivatives of q (excluding the cross derivatives). During application of the two-dimensional grid distribution component it was found that extremely high values of the weight function could occur near the boundaries. These extremely high weights caused the grid to stretch to this boundary point and ignore the remainder of the domain. Therefore, the weight function was limited to be no larger than three standard deviations above the average value of the unsmoothed, unscaled weight function. The weight function was smoothed using

$$\breve{w}_{i,j}^{(0)} = \breve{w}_{i,j}, \tag{32}$$

$$\breve{w}_{i,j}^{p} = \frac{1}{8}[4\breve{w}_{i,j}^{(p-1)} + \breve{w}_{i-1,j}^{(p-1)} + \breve{w}_{i+1,j}^{(p-1)} + \breve{w}_{i,j-1}^{(p-1)} + \breve{w}_{i,j+1}^{(p-1)}] \tag{33}$$

$$\text{where } p = 1, P$$

$$\breve{w}_{i,j}^{n} = \breve{w}_{i,j}^{(P)}, \tag{34}$$

where (p) represents the iteration level, $\breve{w}$ is the limited but unsmoothed weight function, $\breve{w}^{(p)}$ is the partially smoothed weight function, and $\breve{w}^n$ is the final smoothed weight function. The i, j subscripts represented the ξ

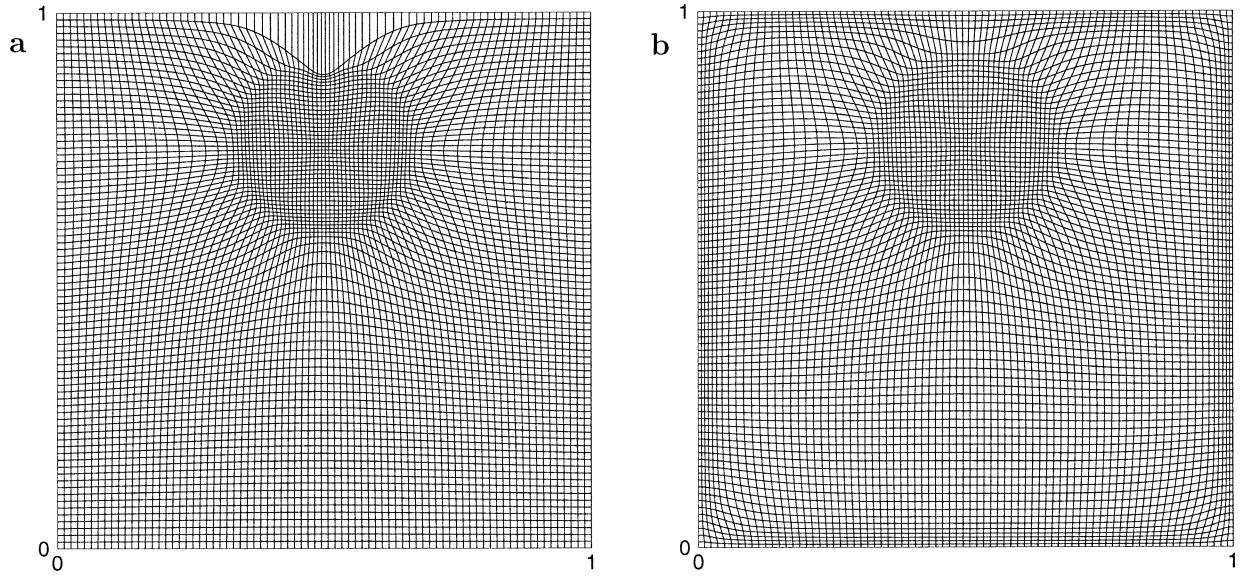

FIG. 5. Gridpoint distributions with and without boundary clustering: 80 × 80 grid points with $\lambda_v = 4.0$, $\lambda_o = 0.0$, and $K = 4$ smoothing passes. (a) Poor quality grid. (b) Improved quality grid.

and η indices, respectively; the (p) superscript represented the iteration level; and the (n) superscript represented the time level.

Like the one-dimensional case the weight function was scaled between 0 and 1, raised to a fractional power (1/2 in this case) to reduce the difference between the significant peaks of the weight function, and added to one to eliminate any zero values:

$$w^n_{i,j} = 1 + \sqrt{\hat{w}^n_{i,j}}, \tag{35}$$

where $\hat{w}$ is the limited, smoothed, and scaled weight function.

Following the grid redistribution scheme of Brackbill and Saltzman (1982) integrals that control the grid distribution using the weight function, orthogonality of the grid, and grid smoothness were set up. Variational calculus was then used to minimize a weighted sum of these measures, with λ_v and λ_o being user-specified coefficients of the grid smoothness and orthogonality terms, respectively. See the appendix for details.

Unlike the one-dimensional case where the equations could be solved using one numerical integration over the entire domain followed by an interpolation for each interior grid point, this set of equations was solved iteratively. Using central finite differences to represent the equations the system becomes block pentadiagonal where each block is a two-by-two matrix that represents the x and y location values for a $(\xi_{i,j}, \eta_{i,j})$ location. The entire system was solved using successive overrelaxation (SOR) with each individual two-by-two block solved using Cramer's rule. The convergence criteria was set to 2% of the unstretched grid cell width so that it was independent of the the number of grid points. In the experiments that follow in section 3d, 50–60 SOR iterations were required to generate the initial grid and typically 10 iterations were required to redistribute the grid points at a time step $n + 1$ since the grid at time step n was used as the initial guess for SOR.

Since $w_{i,j}$ is a function of the physical coordinates, the weight function was updated between iterations using

$$w^{k+1} = w^k + \frac{\partial w^k}{\partial x}(x^{k+1} - x^k) + \frac{\partial w^k}{\partial y}(y^{k+1} - y^k), \tag{36}$$

where Eqs. (23) were used to interchange the independent and dependent variables of the metric terms.

In order to produce acceptable grids when large numerical errors would occur near the boundary it was necessary to allow the boundary grid points to move parallel to the boundary. However, it was found that they could cluster inappropriately closely, producing cells with large aspect ratios as shown in Fig. 5a. This problem was solved as seen in Fig. 5b by specifying the boundary point weights to be 80% of the maximum weight function before smoothing. The smoothing spread the influence of this weight to the interior and the boundary cell areas were kept small. It also provided a more fundamental need of clustering points at the boundaries in anticipation of fine-scaled phenomena entering the domain. Although this could not happen in the rotating cone case, the redistribution technique ultimately was designed to be used with a limited area model and would need to account for such inflow conditions. A potentially more elegant solution to this problem would have been to include a measure of the grid aspect ratio in addition to the spacing, smoothness, and orthogonality criteria in the development of the grid redistribution scheme. The option was not implemented because adverse grid aspect ratios were not observed at

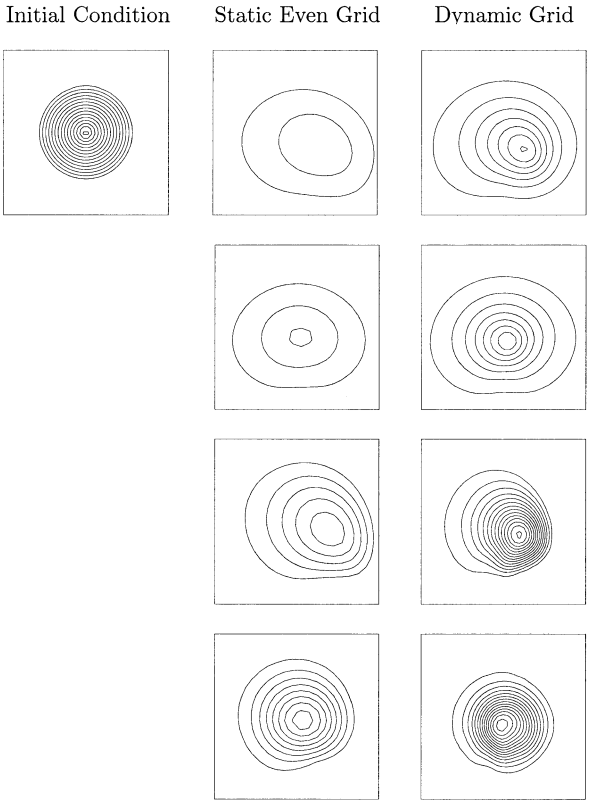

FIG. 6. Results of the rotating cone problem after six revolutions: $0.25 \leq x \leq 0.75$ and $0.5 \leq y \leq 1.0$ in each panel. (upper left) The initial condition, (middle column) results using the static MPDATA scheme, and (right column) results using the dynamic MPDATA scheme. The results in the first row were obtained using two MPDATA iterations and no third-order terms. The results in the second row were obtained using two MPDATA iterations and third-order terms. The results in the third row were obtained using the RPV option without third-order terms. The results in the fourth row were obtained using the RPV option and third-order terms.

other locations, significant computational expense is required to add another grid adaptation criterion, and boundary clustering would have been needed anyway.

d. Two-dimensional experimental results

A series of computations were performed involving six complete revolutions of the cone on a 41×41 grid with different combinations of the static and dynamic grids, inclusion and exclusion of the third-order terms, and different numbers of MPDATA iterations or alternatively the RPV option. Figure 6 shows contour plots of the initial condition and representative samples of the computations. The dynamic grid parameters were set to $\lambda_o = 5.0$ and $\lambda_v = 0$, four smoothing passes, and a safety factor $s = 0.6$. As with the one-dimensional grid when DGA was used, the initial grid was generated by 10 iterations of initializing the grid with q and then redistributing the grid points. The time step was held constant throughout the integration and was based on the initial grid spacing. In agreement with the findings of Margolin and Smolarkiewicz (1989) the effect of additional iterations and the third-order correction terms are a reduction in the diffusion error and a more symmetric distribution of the error, respectively. In contrast to the one-dimensional results, the added effect of the dynamic grid is a dramatic decrease in the diffusion error in all cases.

Figure 7 illustrates the L_2, diffusion, and phase errors for this series of computations. The diffusion error was calculated as

$$E_d = \left| \max_{i,j} (\hat{q}_{i,j}) - \max_{i,j} (q_{i,j}) \right| \tag{37}$$

and shows the expected reduction with diminishing returns when additional iterations are taken. The inclusion of the third-order terms had a tendency to reduce the diffusion error in all cases, so it may be concluded that although the third-order terms do not actually reduce the scheme's order, except in the restrictive case of zero velocity gradient, it has the benefit of lowering not only the phase error but the diffusion error as well. However, the leading cause of the reduction of diffusion error is the use of DGA. In all cases the diffusion error was reduced by a minimum of 25%, by 70% or more in all but two cases, and by 96% in one case.

Figure 7b shows the phase error. The phase error was calculated by taking the magnitude of the distance from the location of the maximum value of q of the final solution from the corresponding location of the initial condition represented on the initial grid. The most dramatic improvements came from the inclusion of the third-order correction terms. When the third-order option was not used, DGA reduced the the phase error. Note that when the third-order option was used in conjunction with the static grid computations with three or more iterations that the phase error was exactly zero, whereas the same calculations with the dynamic grid had a nonzero phase error. Zero phase errors in static grid computations occur when the maximum values of the computed solution and the initial condition occur at the same grid point. In contrast, zero phase error in a DGA computation can only occur when the spatial coordinates of the grid points where the maximum values of the initial condition and final solution occur are the same. The likelihood that these coordinates are exactly the same to machine precision is so small that the existence of some phase error is almost guaranteed in the DGA computations.

Figure 7c shows the L_2 error norm, which may be thought of as showing the combined effect of the diffusion and phase error. This was calculated as

$$E_{L_2} = \left\{ \frac{1}{A} \int_A [\hat{q}(x, y, t) - q(x, y, t)]^2 \, dA \right\}^{1/2}, \tag{38}$$

where $\hat{q}(x, y, t)$ is the analytic solution and $q(x, y, t)$ is the computed solution and the integration was per-

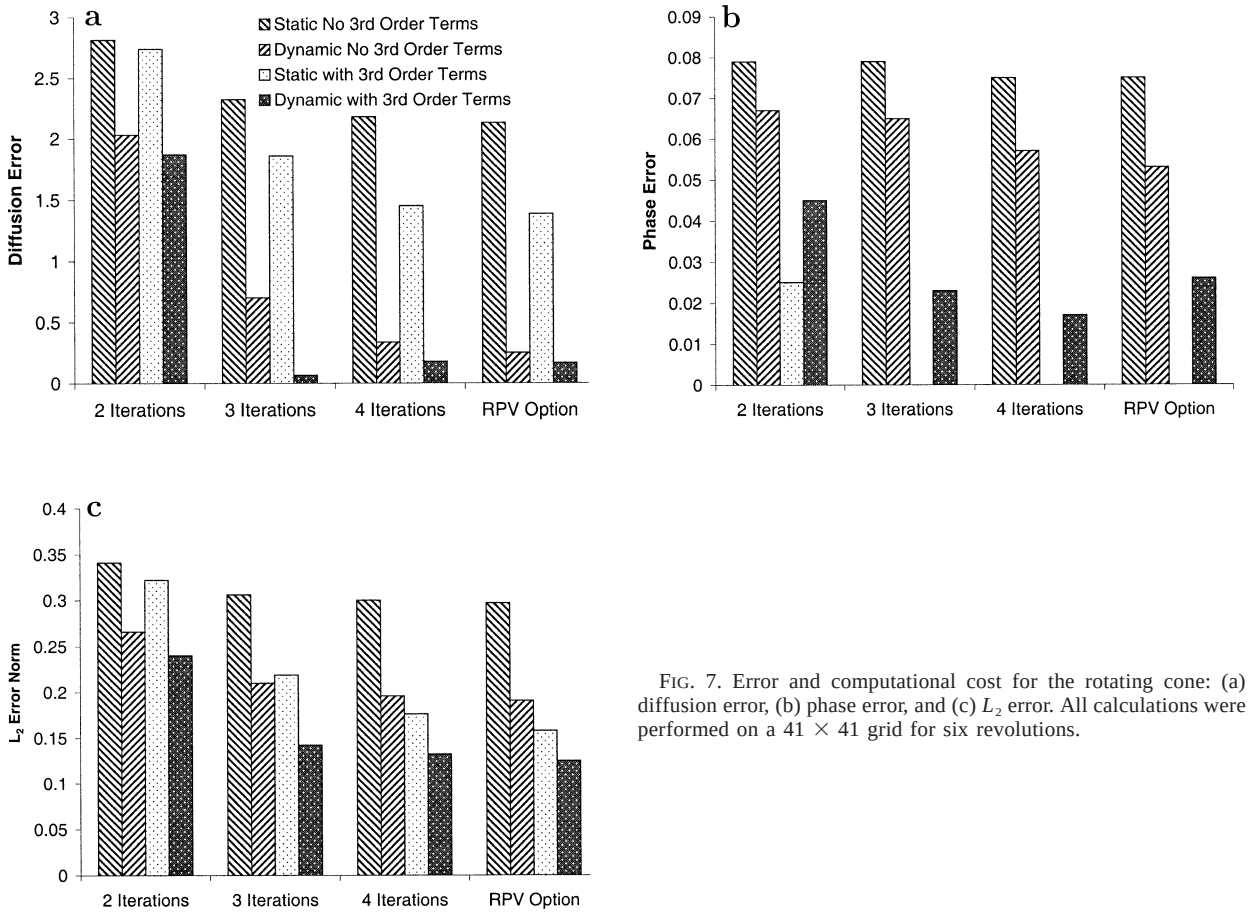

FIG. 7. Error and computational cost for the rotating cone: (a) diffusion error, (b) phase error, and (c) L_2 error. All calculations were performed on a 41 $\times$ 41 grid for six revolutions.

formed using the trapezoidal rule. In this test case the gradient of the the profile was a steep 26.7. Therefore, the slightest phase error resulted in dramatic increases in the L_2 error norm. The use of the third-order terms is thus highly advantageous in reducing the L_2 error norm. Surprisingly, the DGA technique without third-order error terms is quite competitive with the static grid computations with them.

Figure 8 shows the diffusion error as a function of the computational cost for different combinations of grids and MPDATA options. Comparison of the static, uniform grid computations and the DGA computations predictably shows that DGA requires an order of magnitude more in computational resources but also delivers an order of magnitude lower diffusion error when the RPV and third-order terms are used. Two additional computations are shown by the diamond symbol in Fig. 8 where the number of the grid points of the uniform static grid was increased to 101 $\times$ 101 and 201 $\times$ 201. The decrease in computational error achieved by the DGA schemes was never realized even though over five times the computational resources were used. It is clear that the DGA technique is more efficient at reducing the diffusion error than a static, uniform grid with many more grid points.

In the DGA computations for this simple advection case most of the computational effort (78% of the CPU time) was expended solving the elliptic equations to determine the gridpoint distribution at each time step. The remaining time was used to generate the weight function from the prognostic variable field (6%), solve the advection problem using MPDATA and control the logic of the code (12%), and calculate the contravariant velocities at each step (4%). However, as in the one-dimensional case these percentages misrepresent the potential of DGA if extrapolated to a full weather prediction model where the velocity componets, a tracer component (water vapor), and pressure equation must be solved. By categorizing the types of equations that must be solved and using the above percentages as typical computational weights for a full two-dimensional model, an estimate of the effect of DGA on a full model can be made. Since the equations for the two velocity components and the tracer are hyperbolic, they would be given each a weight of 12, like the advection equation. Since the pressure equation would be elliptic, its solution would demand

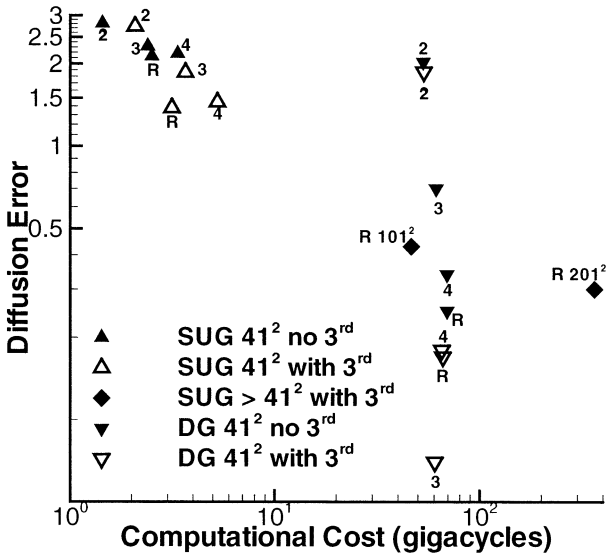

FIG. 8. Cost–benefit ratio of DGA as compared to static uniform grid computations. SUG and DG in the legend represent static uniform grid, and dynamic grid, respectively. The 41^2 indicates the number of grid points. The 2, 3, or 4 adjacent to the symbol indicates the number of MPDATA iterations used. An R indicates that the RPV option was used. The 101^2 and 201^2 adjacent to the diamond symbols indicates that number of grid points used in these uniform, static grid computations.

resources similar to the grid redistribution equation amounting to a weight of 78, like the elliptic grid redistribution equation. The sum of these in addition to the weights of 78, 6, and 4 for the grid redistribution; the calculation of the grid redistribution weights; and the calculation of the relative velocities yields a total of 202. This estimate indicates that DGA would consume 44% of the total computational resources as compared to 88% for the simple advection case.

Although further quantative extrapolation to a full three-dimensional model is difficult, the DGA overhead would clearly decrease further than in the full two-dimensional case. The computational effort to solve the pressure equation would increase by a factor at least as great as the number of vertical levels in the model due to the nature of elliptic solvers. When developing a three-dimensional model it could be sufficient and desirable to keep vertical columns of grid points aligned. A grid redistribution scheme to accomplish this could consist of the two-dimensional grid redistribution scheme in the horizontal directions and the one-dimensional grid redistribution scheme applied to each column of grid points. This computational expense would not approach that of the pressure solver. A conservative estimate of the additional overhead to perform DGA in a full three-dimensional model might be 30%. Assuming similar reductions in computational error can be achieved as inidicated in Fig. 8, then this added expense would certainly be warranted.

4. Conclusions

A dynamic grid adaptation (DGA) model has been developed that works effectively with the nonoscillatory, second-order-accurate multidimensional positive definite advection transport algorithm (MPDATA) scheme. Second-order accuracy was maintained in MPDATA by (i) deriving the correction terms to account for grid motion, and (ii) using the contravariant velocity for advecting the prognostic variable.

Based upon one-dimensional considerations, a new extended CFL stability condition has been developed for DGA methods that shows the partial Lagrangian characteristics of the technique. This extended CFL stability condition was successfully used, in the one-dimensional case, to control the movement of the grid points and keep the scheme stable. It was found that the gridpoint control was needed only in cases when many grid points ($>$250 in this case) were used. When the DGA technique is chosen, a smaller number of grid points is used in a more judicious manner than with a static, uniform-resolution grid model. A user should not expect to see the same degree of error reduction by dramatically increasing the number of grid points in a DGA model as they would expect to see in a static, uniform-resolution grid model.

The two-dimensional simulations of the rotating cone test problem showed very significant reduction in the diffusion error due to DGA of up to 90%. While DGA was more computationally expensive than using uniform grids with the same number of grid points, it was far less expensive than uniform grids with similar accuracy and many more grid points.

The major cost associated with DGA for the advection case is the solution of the elliptic grid redistribution scheme. If applied to the full set of momentum, energy, and elliptic pressure equations, which need to be solved iteratively, the cost of moving the grid points should be a much smaller fraction of the total.

Acknowledgments. This work was supported through funding by NSF Grant ATM-9619811 to Iowa State University, the Department of Energy through the Climate Change Prediction Program (CCPP) research initiative, and the Iowa State University Global Change Fellowship. The authors are grateful to the reviewers who gave valuable input and improved the quality of this paper.

APPENDIX

Two-Dimensional Grid Generation Equations

The two-dimensional grid redistribution scheme based on the scheme developed by Brackbill and Saltzman (1982) is developed by considering a weighted measure of the grid smoothness, cell volume, and grid orthogonality:

$$I = \int_D [(\nabla\xi \cdot \nabla\xi + \nabla\eta \cdot \nabla\eta) + \lambda_v w J_2 + \lambda_o (\nabla\xi \cdot \nabla\eta)^2 J_2^3]\, dA, \quad \text{(A1)}$$

where ∇ is the gradient with respect to the physical coordinates (x, y), w is the weight function, J_2 is the Jacobian of the transformation matrices, and V is the volume of the cell. The first term in brackets represents a measure of the smoothness of the grid. The terms beginning with the user-specified λ_v and λ_o represent a measure of the grid cell volume and grid orthogonality, respectively. According to Thompson et al. 1985 the J_2^3 is ''somewhat arbitrary, and caused orthogonality to be emphasized more strongly in the larger cells.''

In order to take advantage of the even grid spacing of the computational domain Eqs. (23) were used to switch the independent and dependent variables. Using calculus of variations these measures were minimized yielding a quasi-linear set of elliptic equations:

$$c_4\frac{\partial^2 x}{\partial\xi^2} + c_5\frac{\partial^2 x}{\partial\xi\partial\eta} + c_6\frac{\partial^2 x}{\partial\eta^2} + c_1\frac{\partial^2 y}{\partial\xi^2} + c_2\frac{\partial^2 y}{\partial\xi\partial\eta} + c_3\frac{\partial^2 y}{\partial\eta^2} = \lambda_v R_{v1}$$

$$c_1\frac{\partial^2 x}{\partial\xi^2} + c_2\frac{\partial^2 x}{\partial\xi\partial\eta} + c_3\frac{\partial^2 x}{\partial\eta^2} + c_7\frac{\partial^2 y}{\partial\xi^2} + c_8\frac{\partial^2 y}{\partial\xi\partial\eta} + c_9\frac{\partial^2 y}{\partial\eta^2} = \lambda_v R_{v2} \quad \text{(A2)}$$

where the cs are functions of the of λ_o and the first partial derivatives of x and y.

REFERENCES

Bacon, D. P., and Coauthors, 2000: A dynamically adapting weather and dispersion model: The Operational Multiscale Environmental Model with Grid Adaptivity (OMEGA). *Mon. Wea. Rev.,* **128,** 2044–2076.

Behrens, J., 1996: An adaptive semi-Lagrangian advection scheme and its parallelization. *Mon. Wea. Rev.,* **124,** 2386–2395.

Berger, M., and J. Oliger, 1984: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.,* **53,** 484–512.

Boris, J. P., and D. L. Book, 1973: Flux-corrected transport, I. SHASTA, a fluid transort algorithm that works. *J. Comput. Phys.,* **11,** 38–69.

Brackbill, J. U., and J. S. Saltzman, 1982: Adaptive zoning and singular problems in two dimensions. *J. Comput. Phys.,* **46,** 342–368.

DeMaria, M., S. D. Aberson, and K. A. Ooyama, 1992: A nested spectral model for hurricane track forecasting. *Mon. Wea. Rev.,* **120,** 1628–1643.

Dietachmayer, G. S., and K. K. Droegemeier, 1992: Application of continuous dynamic grid adaptation techniques to meteorological modeling. Part I: Basic formulation and accuracy. *Mon. Wea. Rev.,* **120,** 1675–1706.

Fiedler, B. H., and R. J. Trapp, 1993: A fast dynamic grid adaptation scheme for meteorological flows. *Mon. Wea. Rev.,* **121,** 2879–2888.

Finley, C. A., W. R. Cotton, and R. A. Pielke, 1998a: Numerical simulation of two tornadoes produced by a high-precipitation supercell. Preprints, *19th Conf. on Severe Local Storms,* Minneapolis, MN, Amer. Meteor. Soc., 206–209.

——, ——, and ——, 1998b: Secondary vortex development in a tornado vortex produced by a simulated supercell thunderstorm. Preprints, *19th Conf. on Severe Local Storms,* Minneapolis, MN, Amer. Meteor. Soc., 359–362.

Hawken, D., J. Gottlieb, and J. Hansen, 1991: Review of some adaptive node-movement techniques in finite-element and finite-difference solutions of partial differential equations. *J. Comput. Phys.,* **95,** 254–302.

Iselin, J. P., 1999: A dynamic adaptive grid MPDATA scheme: Application to the computational solution of atmospheric tracer transport problems. Ph.D. dissertation, Iowa State University, 174 pp.

Kim, J. K., and J. R. Thompson, 1990: Three-dimensional adaptive grid generation on a composite-block grid. *AIAA J.,* **38,** 470–477.

Margolin, L. G., and P. K. Smolarkiewicz, 1989: Antidiffusive velocities for multipass donor cell advection. Tech. Rep. UCID-21866, Lawerence Livermore National Laboratory, Livermore, CA, 44 pp.

Prusa, J. M., P. K. Smolarkiewicz, and R. R. Garcia, 1996: Propogation and breaking at high altitudes of gravity waves excited by tropospheric forcing. *J. Atmos. Sci.,* **53,** 2186–2216.

——, ——, and A. A. Wyszogrodzki, 2001: Simulations of gravity wave induced trubulence using 512 PE CRAY T3E. *Int. J. Appl. Math. Comput. Sci.,* **11,** 101–115.

Skamarock, W., and J. B. Klemp, 1993: Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flow. *Mon. Wea. Rev.,* **121,** 788–804.

——, M. L. Weisman, and J. B. Klemp, 1994: Three-dimensional evolution of simulated long-lived squall lines. *J. Atmos. Sci.,* **51,** 2563–2583.

Smolarkiewicz, P. K., and L. G. Margolin, 1998: MPDATA: A finite-difference solver for geophysical flows. *J. Comput. Phys.,* **140,** 459–480.

——, and J. M. Prusa, 2002: VLES modeling of geophysical fluids with nonoscillatory forward-in-time schemes. *Int. J. Numer. Methods Fluids,* in press.

Thompson, J. F., Z. U. A. Warsi, and C. W. Mastin, 1985: *Numerical Grid Generation: Foundations and Applications.* Elsevier Science.

Tomlin, A., M. Berzens, J. Ware, J. Smith, and M. Pilling, 1997: On the use of adaptive gridding methods for modelling chemical transport from multi-scale sources. *Atmos. Environ.,* **31,** 2945–2959.

Yanenko, N., V. D. Lisseikin, and V. M. Kovenia, 1979: The method of the solution of gaz dynamical problems in moving meshes. *Lect. Notes Phys.,* **91,** 48–61.

Zalesak, T. S., 1979: Fully mulidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.,* **31,** 335–362.