Journal of Computational Physics **165,** 437–472 (2000)
doi:10.1006/jcph.2000.6620, available online at http://www.idealibrary.com on IDEAL®

An Adaptive Grid Algorithm for Air-Quality Modeling

R. K. Srivastava,*[,1] D. S. McRae,* and M. T. Odman†[,2]

**Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, North Carolina 27695-7910; and †MCNC–Environmental Programs, P.O. Box 12889, Research Triangle Park, North Carolina 27709-2889*
E-mail: srivastava.ravi@epa.gov, mcrae@eos.ncsu.edu, odman@ncsc.org

Received May 12, 1999; revised August 30, 2000; published online November 16, 2000

A new dynamic adaptive grid algorithm has been developed for use in air-quality modeling. This algorithm uses a higher order numerical scheme—the piecewise parabolic method (PPM)—for computing advective solution fields, a weight function capable of promoting grid node clustering by moving grid nodes, and a conservative interpolation equation using PPM for redistributing the solution field after movement of grid nodes. Applications of the algorithm to model problems show that the algorithm provides solutions more accurate than those obtained with static grids. Performance achieved in model problem simulations indicates that the algorithm has the potential to provide accurate air-quality modeling solutions at costs that may be significantly less than those incurred in obtaining equivalent static grid solutions.

Key Words: grid adaptation; air-quality modeling; advection.

1. INTRODUCTION

The physical and chemical processes responsible for air pollution span a wide range of spatial scales. For example, there may be point sources, such as power plants, that are characterized by spatial scales relatively small compared to regional-scale pollutant plumes from such sources. Therefore, to accurately model the transport and chemistry of air pollutants, an air-quality model (AQM) must be able to adequately resolve the pertinent spatial scales. This can be achieved by varying the physical grid node spacing in an AQM to provide resolution where needed.

[1] Present affiliation: U.S. Environmental Protection Agency, MD-65, 86 Alexander Dr., Research Triangle Park, NC 27711.

[2] Present affiliation: Georgia Institute of Technology, School of Civil & Environmental Engineering, Atlanta, GA 30332-0512.

0021-9991/00 $35.00

One approach to achieving increased local solution resolution involves using embedded Cartesian grids, or static nested grids, such as those described by Odman and Russell [1] and Odman *et al.* [2]. This approach may be limited by (1) the uncertainty in nested grid(s) placement(s) since pertinent locations may not be known a priori, (2) the loss in solution accuracy resulting from grid boundary interface problems, and (3) the inability to adjust rapidly to dynamic changes in solution resolution requirements. Another approach to achieving local solution resolution involves using grids with moving nodes, i.e., dynamic adaptive grids. In principle, such grids would be continuous and would adjust dynamically to changing solution resolution requirements. Therefore, use of such grids would not be constrained by the limitations associated with use of nested grids.

Recently, ideas promoting use of dynamic adaptive grids in atmospheric modeling have gained popularity. Dietachmayer and Droegemeier [3] use a variational formulation of adaptive grid generation equations to compute solutions to test problems. Almgren *et al.* [4] have used a nested hierarchy of grids, with simultaneous refinement of grids in both space and time, to resolve the release of hot gas into the atmosphere. Skamarock and Klemp [5] have used a hierarchical grid approach to model a compressible formulation of the atmospheric flow equations. Tomlin *et al.* [6] have investigated the use of an adaptive unstructured grid method to obtain solutions of test problems of interest in air-quality modeling.

A grid adaptation algorithm for aerospace applications has been developed by Benson and McRae [7–9]. This algorithm, called the *Dynamic Solution Adaptive Grid Algorithm* (DSAGA), uses weight functions constructed from the absolute values of the gradients of solution variables, along with a *center-of-mass* scheme, to move grid nodes. These nodes are repositioned in a parametric space to avoid the potential for grid-line cross-over. After the repositioning of grid nodes, solution variables are corrected. This correction is completed using an equation that is obtained by splitting the time-dependent terms describing the movement of the grid nodes from the steady terms of the general conservation laws for moving control volumes. DSAGA has been used to compute unsteady, three-dimensional, turbulent, viscous flows [10–12]. Additionally, Srivastava *et al.* [13] have used DSAGA to compute test cases of interest in air-quality modeling. Recently, Laflin and McRae [14] and Laflin [15] have developed the *Solver-Independent, Efficient r-Refinement Algorithm* (SIERRA) that provides robust and computationally efficient implementation strategies for grid adaptation algorithms. SIERRA incorporates a solver-independent weight function and a solver-independent solution field redistribution procedure.

This paper describes a new dynamic adaptive grid algorithm suitable for use in AQMs. This algorithm was built on the foundation provided by the DSAGA algorithm of Benson and McRae [7–9]. However, the new algorithm extends the DSAGA framework by utilizing the *piecewise parabolic method* (PPM) developed by Collela and Woodward [16] for computing the advective fluxes and the fluxes resulting from grid movement. Moreover, the new algorithm incorporates the SIERRA weight function and solution field redistribution procedure. This new algorithm is called the *Dynamic Solution Adaptive Grid Algorithm–PPM* (DSAGA–PPM) [17].

Given a fixed number of grid nodes, DSAGA–PPM can determine automatically an appropriate spatial distribution of these nodes and can update this distribution in response to changes in the evolving numerical solution. Thus, this algorithm can adequately resolve any evolving solution features. In general, to provide a given level of solution accuracy, DSAGA–PPM would require fewer grid nodes than uniform or embedded grids since it

would cluster most of the grid nodes around solution features needing refinement and use few grid nodes elsewhere. DSAGA–PPM can distribute grid points in all grid directions simultaneously and can be applied to steady or unsteady flows and maintain temporal accuracy.

Following the description of DSAGA–PPM, this paper examines its ability to calculate accurate solutions. For each of the test problems, results were obtained using DSAGA–PPM and the corresponding *Static Grid Algorithm–PPM* (SGA–PPM), which is obtained by deactivating the grid adaptation procedures in DSAGA–PPM. These results are compared to reveal the advantages of using DSAGA–PPM. The SGA–PPM solutions are obtained on a uniformly spaced grid that is identical to the starting uniformly spaced grid used in the corresponding DSAGA–PPM application.

2. GOVERNING DIFFERENTIAL EQUATIONS

Before describing DSAGA–PPM, it is useful to review the theoretical foundations of air-pollution modeling. McRae *et al.* have provided a comprehensive review of this subject [18]. A summary of this review is presented in this section.

In an arbitrary, time-varying, spatial region $\Omega(t)$, located in the Euclidean space E^3 and bounded by $\partial\Omega(t)$, a spatial point is given by $\mathbf{X} = \{x, y, z\} \in \Omega(t)$. In $\Omega(t)$ the conservation of mass for each of N chemical species $c_l(\mathbf{X}, t), l = 1, \ldots, N$ is expressed as

$$\frac{\partial c_l}{\partial t} + \nabla \cdot (\mathbf{V} c_l) = \nabla \cdot (K \cdot \nabla c_l) + R_l(c_1, \ldots, c_N) + S_l(\mathbf{X}, t); \qquad l = 1, \ldots, N. \quad (1)$$

In (1) c_l is the mass concentration of pollutant l (mass of pollutant l/volume of air); $\mathbf{V}(\mathbf{X}, t) = (u, v, w)$ is the specified wind field; K is a second-order, diagonal, turbulent diffusivity tensor; R_l is the net generation of chemical species l by chemical reactions; and S_l is the rate of source addition for the chemical species l. The system of equations (1) is referred to as *the atmospheric diffusion equation* [18] and constitutes the governing system for an AQM.

The atmospheric diffusion equation is solved with specified initial and boundary conditions. For each of species c_l, an initial distribution $c_l(\mathbf{X}, 0)$ is specified and conditions at the boundary are imposed by inhomogeneous mixed Neumann and Dirichlet boundary conditions. A discussion of the initial and boundary conditions can be found in Srivastava [17] and Reynolds *et al.* [19].

In air-pollution modeling, (1) and the associated initial and boundary conditions are appropriately transformed to accommodate the resolution of meteorological (boundary layer, cloud) and geographical (topography) features. Discussions of the various transformations can be found in Kasahara [20], Toon *et al.* [21], and McRae *et al.* [18]. Since the purpose of this paper is to illustrate the applicability of DSAGA–PPM to air-pollution modeling via model test problems, such transformations are not examined herein.

3. DSAGA–PPM

DSAGA–PPM includes the finite-volume procedures for advancing the governing system (i.e., Eq. (1) and the associated initial and boundary conditions) in time using a non-uniform grid, for moving the grid nodes to region(s) requiring solution refinement, and

for conservatively redistributing the solution field over the resulting adapted grid. These procedures are briefly described below. Details of these procedures can be found in [17].

3.1. *Time Advancement of the Governing System*

In the AQMs based on finite-volume methods, the governing system, with transformed vertical coordinate, is usually advanced in time on static Cartesian grids with uniform spacing in the horizontal plane and non-uniform spacing in the vertical direction. However, in general, a grid adapted by moving nodes will not be uniform. Therefore in DSAGA–PPM, a coordinate transformation is applied to the governing system to relate the physical domain expressed in Cartesian coordinates (x, y, z) to a computational domain expressed in general curvilinear coordinates ξ^m, $m = 1, 2, 3$. The grid in the general curvilinear coordinates is assumed to be uniform and unit-spaced. This procedure allows the solution of the governing system to be obtained on arbitrary physical grids (i.e., grids that are curved in space and are aligned to shapes of existing solution features). The mapping between the two coordinate systems is given by

$$\xi^m = \xi^m(x, y, z); \qquad m = 1, 2, 3. \tag{2}$$

As discussed in Roache [22], in general, the most accurate numerical results are obtained if numerical differencing is based on the conservative form of the governing equations. Methods for manipulating partial differential equations that preserve conservative properties are described in Anderson *et al.* [23], Oberkampf [24], and Vinokur [25]. Using these methods, the conservative form of the transformed atmospheric diffusion equation is given by

$$\frac{\partial \hat{c}_l}{\partial t} + \frac{\partial \hat{E}_l^m}{\partial \xi^m} = \hat{R}_l + \hat{S}_l; \qquad l = 1, \ldots, N, \qquad m = 1, 2, 3, \tag{3}$$

where

$$\hat{c}_l = \frac{c_l}{J}, \qquad l = 1, \ldots, N \tag{4}$$

$$\hat{E}_l^1 = \frac{\left(\xi_x^1 E_l + \xi_y^1 F_l + \xi_z^1 G_l\right)}{J}, \qquad l = 1, \ldots, N \tag{5}$$

$$\hat{R}_l = R_l(c_1, \ldots, c_N)/J, \qquad l = 1, \ldots, N \tag{6}$$

and

$$\hat{S}_l = S_l(\mathbf{X}, t)/J, \qquad l = 1, \ldots, N \tag{7}$$

with $E_l = c_l u - K_{xx}\frac{\partial c_l}{\partial x}$, $F_l = c_l v - K_{yy}\frac{\partial c_l}{\partial y}$, and $G_l = c_l w - K_{zz}\frac{\partial c_l}{\partial z}$ for diagonal K. The expressions for the Jacobian J and the metrics of the transformation (2) are available in [23].

In general, Eq. (3) represents a system of stiff partial differential equations since the time scales associated with typical chemical transformations are far smaller than those associated with transport due to advection and turbulent diffusion. Therefore decoupling transport and chemistry and solving for these processes in sequential steps result in more

efficient computation of transport. As discussed in [18], Eq. (3) is operator- (or time-) split [26] to compute transport, chemistry, and source processes in sequential steps. The splitting of Eq. (3) is not unique and differs between AQMs. The splitting sequence used in this work is

$$c_l(t+\Delta t) = L_{c_l} L_{s_l} L_{\text{diff}} L_{\text{adv}} c_l(t), \qquad l = 1, \ldots, N \tag{8}$$

$$c_l(t+2\Delta t) = L_{\text{adv}} L_{\text{diff}} L_{s_l} L_{c_l} c_l(t+\Delta t), \qquad l = 1, \ldots, N \tag{9}$$

where

$$L_{\text{adv}} = \left[I + dt \cdot J \frac{\partial}{\partial \xi^m} \left(\left(\xi_x^m(-u) + \xi_y^m(-v) + \xi_z^m(-w) \right) / J \right) \right] \tag{10}$$

$$L_{\text{diff}} = \left[I + dt \cdot J \frac{\partial}{\partial \xi^m} \left(\left(\xi_x^m \left(K_{xx} \frac{\partial c_l}{\partial x} \right) + \xi_y^m \left(K_{yy} \frac{\partial c_l}{\partial y} \right) + \xi_z^m \left(K_{zz} \frac{\partial c_l}{\partial z} \right) \right) \Big/ J \right) \right] \tag{11}$$

$$L_{s_l} = [I + dt \cdot S_l(\mathbf{X}, t)] \tag{12}$$

$$L_{c_l} = [I + dt \cdot R_l(c_1, \ldots, c_N)]. \tag{13}$$

The splitting defined by the relations (8) and (9) is symmetric and, as explained in [18], is second-order accurate.

The remainder of this section develops the application of DSAGA–PPM to the atmospheric diffusion equation in two dimensions only. Application of this algorithm to the atmospheric diffusion equation in three dimensions would be a logical extension of the procedures presented in the following sections.

Calculation of two-dimensional advective transport. The finite-volume representation of the advection component of (3), with splitting (8) and (9), can be used to compute two-dimensional advection at $t + \Delta t$ as follows:

$$\left(\bar{c}_l^{n+1} V\right)_{i,j} - \left(\bar{c}_l^{n} V\right)_{i,j} + \Delta t_{\text{adv}} \left(\hat{E}_l^{\text{adv}} \big|_{i+1/2,j} - \hat{E}_l^{\text{adv}} \big|_{i-1/2,j} + \hat{F}_l^{\text{adv}} \big|_{i,j+1/2} - \hat{F}_l^{\text{adv}} \big|_{i,j-1/2} \right)^n = 0. \tag{14}$$

In the above equations, $\bar{c}_l$ is the average concentration of chemical species l in cell volume $V_{i,j}$ and $\hat{E}_l^{\text{adv}}$ and $\hat{F}_l^{\text{adv}}$ are the net mass effluxes due to advection at cell sides $i \pm 1/2$, j and i, $j \pm 1/2$, respectively. The time step for advection, Δt_{adv}, is bounded by stability considerations [17]. In a finite volume formulation, the Jacobian J is simply the ratio of the volume of the computational cell to the volume, V, of the corresponding physical cell. In the computational domain, the grid is chosen to be uniform and unit-spaced for convenience. Consequently, the volume of each computational cell is unity and the Jacobian for each cell becomes $\frac{1}{V}$. Note that in two dimensions, $V_{i,j}$, is the area of the cell i, j.

In air-quality simulations, mass conservation of species, monotonicity of the solution fields, and a high order of accuracy need to be maintained during numerical computations. The *piecewise parabolic method* (PPM)—a numerical scheme for computing advection—is monotonic, is third-order accurate for variable grid spacing, and is conservative [27].

Therefore, the PPM scheme is used to compute the fluxes $\hat{E}_l^{\text{adv}}$ and $\hat{F}_l^{\text{adv}}$ required in (14). The PPM scheme, developed for modeling fluid flows with strong shocks and discontinuities, is a higher order extension of Godunov's method and uses a parabola as the basic interpolation function in a finite-volume formulation [16]. A comparison of several popular advection schemes, including the PPM scheme, can be found in Odman and Russell [28].

In any two- or three-dimensional finite-volume scheme, numerical sources may result from errors in differencing of the metrics. As discussed in Hindman [29], such sources could affect solution quality significantly. To determine the presence of such sources, the *uniform flow* test has been used by Pulliam and Steger, Thomas and Lombard, and others [30–32]. In this test, each of the dependent variables is initialized with a constant value over the entire grid and then the solution is advanced through one time step with the boundary values fixed. If the numerics are consistently formulated, then the solution after the end of the time step must equal the initial conditions. As discussed in Gielda and McRae [33], an analytical analysis of discretized difference equations, conducted using uniform flow conditions, yields any numerical sources resulting from the difference algorithm. Subsequently, these sources are eliminated by adding terms equal to the negative of the numerical source terms to the difference equations. In DSAGA–PPM, this procedure was used to eliminate the advective numerical sources, as described below.

For a flow with uniform (or constant) dependent variables c_l, the PPM scheme reduces to the first-order upwind scheme since each cell now has constant (instead of parabolic) profiles of the dependent variables. Expanding (14) for constant c_l,

$$\begin{aligned}\overline{c}_l^{(n+1)}\big|_{i,j} = \overline{c}_l^{n}\big|_{i,j} &- \frac{\Delta t_{\text{adv}} \cdot \overline{c}_l^{n}\big|_{i,j}}{V_{i,j}} \times \left[\left(\frac{\xi_x^1}{J}u + \frac{\xi_y^1}{J}v\right)_{i+1/2,j}\right. \\ &\left. - \left(\frac{\xi_x^1}{J}u + \frac{\xi_y^1}{J}v\right)_{i-1/2,j}\right]^n - \frac{\Delta t_{\text{adv}} \cdot \overline{c}_l^{n}\big|_{i,j}}{V_{i,j}} \\ &\times \left[\left(\frac{\xi_x^2}{J}u + \frac{\xi_y^2}{J}v\right)_{i,j+1/2} - \left(\frac{\xi_x^2}{J}u + \frac{\xi_y^2}{J}v\right)_{i,j-1/2}\right]^n. \end{aligned} \tag{15}$$

Upon inspection of the above equations for constant dependent variables c_l, the numerical source term is identified to be

$$\begin{aligned}&- \frac{\Delta t_{\text{adv}} \cdot \overline{c}_l^{n}\big|_{i,j}}{V_{i,j}}\left[\left(\frac{\xi_x^1}{J}u + \frac{\xi_y^1}{J}v\right)_{i+1/2,j} - \left(\frac{\xi_x^1}{J}u + \frac{\xi_y^1}{J}v\right)_{i-1/2,j}\right]^n \\ &- \frac{\Delta t_{\text{adv}} \cdot \overline{c}_l^{n}\big|_{i,j}}{V_{i,j}}\left[\left(\frac{\xi_x^2}{J}u + \frac{\xi_y^2}{J}v\right)_{i,j+1/2} - \left(\frac{\xi_x^2}{J}u + \frac{\xi_y^2}{J}v\right)_{i,j-1/2}\right]^n. \end{aligned} \tag{16}$$

Subsequently, the negative of this term is added to (14) to eliminate numerical sources.

It should be noted that (16) is a discrete representation of velocity divergence in general curvilinear coordinates ξ^m, $m = 1, 2, 3$. While velocity divergence would analytically be zero for a divergence-free velocity field, (16) may still be non-zero due to approximations used in evaluation of metrics.

Calculation of two-dimensional diffusive transport. The finite-volume representation of the turbulent-diffusion component of (3), with splitting (8) and (9), can be used to compute

two-dimensional turbulent diffusion. Accordingly, the turbulent diffusion contribution to the solution at $t + \Delta t$ is given by

$$\left(\bar{c}_l^{n+1}V\right)_{i,j} - \left(\bar{c}_l^{n}V\right)_{i,j} + \Delta t_{\text{diff}}\left(\hat{E}_l^{\text{diff}}\big|_{i+1/2,j} - \hat{E}_l^{\text{diff}}\big|_{i-1/2,j} + \hat{F}_l^{\text{diff}}\big|_{i,j+1/2} - \hat{F}_l^{\text{diff}}\big|_{i,j-1/2}\right)^n = 0, \qquad l = 1, \ldots, N. \tag{17}$$

In Eq. (17), $\bar{c}_l$ is the average concentration of chemical species l in cell volume $V_{i,j}$ and $\hat{E}_l^{\text{diff}}$ and $\hat{F}_l^{\text{diff}}$ are the net mass effluxes due to turbulent diffusion at cell sides $_{i\pm1/2,j}$ and $_{i,j\pm1/2}$, respectively. Details on calculation of diffusive fluxes and the diffusive time step, Δt_{diff}, are available in Srivastava [17].

Note that advection and turbulent diffusion are calculated using their own time steps. These processes are synchronized in time to ensure that they advance through the same time period in any solution step. In DSAGA–PPM, this synchronization is accomplished by computing advection using Δt_{adv} and then repeatedly applying diffusion using Δt_{diff} (or a fraction of it) until the total time step for diffusion equals Δt_{adv}.

The boundary conditions for species transport are termed either *inflow* or *outflow*, depending on the direction of the flow at the grid boundary under consideration. In fluid flow situations, generally the species concentrations at inflow boundaries are known as a function of time. The outflow boundary conditions are generally not known and, therefore, need to be computed. In DSAGA–PPM, zero-concentration-gradient boundary conditions are used at outflow boundaries in calculation of advection and turbulent diffusion.

Treatment of emission sources. Most emissions are released from either point locations (point sources) or area regions (area sources). In order to determine the incremental contribution from source emissions into a computational cell, consider a particular cell of base area A_{b} and uniform height $h(t)$. Given the mass emission rate E_l^{P} (e.g., kg/s) from a point source, the corresponding rate of change of concentration *in the cell containing this source* is given by

$$S_l^{\text{P}} = \frac{E_l^{\text{P}}}{h(t)A_{\text{b}}}. \tag{18}$$

Note that emissions from area sources are introduced as flux boundary conditions.

Computation of chemistry. In Eq. (1), the terms $R_l, l = 1, \ldots, N$, describe the contributions to the rates of change of N chemical species concentrations, $c_1, c_2, \ldots, c_N$, due to chemical reactions. At any time, for any spatial point, the rate of change of each species concentration due to chemical reactions can be described by a set of coupled, nonlinear ordinary differential equations,

$$\frac{dc_l}{dt} = R_l(c_1, c_2, \ldots, c_N), \qquad l = 1, \ldots, N \tag{19}$$

and the associated initial conditions $c_l(0) = c_l^0, l = 1, \ldots, N$.

The functional form of the terms $R_l, l = 1, \ldots, N$, can be developed by considering a homogeneous system in which N single-phase species participate in m reaction steps of the form

$$\sum_{l=1}^{N} \alpha_{jl} c_l \rightarrow \sum_{l=1}^{N} \beta_{jl} c_l, \qquad j = 1, \ldots, m, \tag{20}$$

where the coefficients α_{jl}, β_{jl} denote the reactant and product stoichiometry, respectively, in the reaction step j. If the reaction rates, r_j, of the m individual reactions are described by the mass action law

$$r_j = k_j \prod_{l=1}^{N} c_l^{\alpha_{jl}}, \qquad j = 1, \ldots, m, \tag{21}$$

where k_j is a temperature-dependent rate constant, then the chemistry source terms are given by

$$\frac{dc_l}{dt} = R_l = \sum_{j=1}^{m} (\beta_{jl} - \alpha_{jl}) r_j, \qquad l = 1, \ldots, N. \tag{22}$$

Atmospheric chemistry mechanisms include reactions with characteristic time scales that differ by orders of magnitude. For such mechanisms, Eqs. (22) lead to "stiff" systems of differential equations.

In this work *the asymptotic integration method* of Young and Boris [34] is used to solve the system of equations (22). This method is self-starting, is very fast, and requires minimal storage [18]. In this method, a second-order predictor-corrector scheme that takes into account the stiffness of equations is employed to integrate the system (22). The method has been tested against the very accurate chemistry solver EPISODE [35] and has been found to provide comparable results at significantly reduced costs [18].

3.2. *Grid Adaptation and Solution Correction*

In the following sections, procedures for moving the grid nodes to region(s) requiring solution refinement, and for conservatively redistributing the solution field over the resulting adapted grid, are described.

Weight function formulation. DSAGA–PPM requires a weight function $w(x, y, t)$, which will be large in regions where grid clustering is necessary to achieve desired solution accuracy. In a flow with reacting species, these species may undergo complex dynamic transformations in time. Therefore, it may not be possible to predict *a priori* which of the species would dictate grid adaptation requirements in simulation of such a flow. Moreover, in atmospheric flows, the chemistry processes in one spatial region may be quite different from those in another region. As a result, some of the species may need resolution in one part of the modeled region while other species may need resolution in other part(s). Further, these resolution needs may change with time. One possible strategy to ensure adequate grid adaptation in such a flow may be to use a weight function that takes into account the resolution requirements of each of the species included in the chemical mechanism, at each time step in the simulation.

Using the SIERRA formulation, developed by Laflin and McRae [14] and Laflin [15], weight functions can be designed that are easy to compute and that promote both grid node clustering and grid alignment adaptation processes. Using this formulation, a weight function for use in simulations of reacting flows may be expressed as

$$w_o = V_o^{1+e_1} \left| \sum_l (\Delta^2 \phi_l)_o + w_{\min} \right|, \quad \phi_l = c_l \left(\frac{V}{V_o} \right)^{e_2}, \tag{23}$$

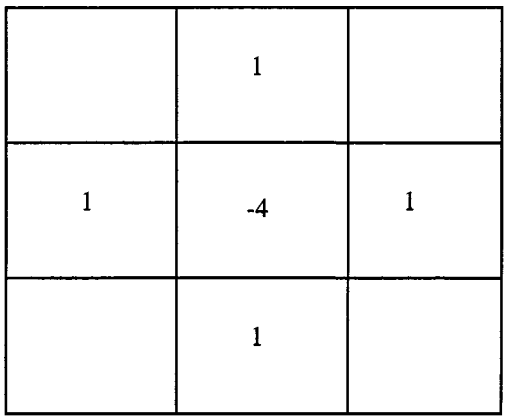

FIG. 1. Five-point stencil for computing discretized Laplacian.

where Δ^2 is the discrete approximation of the Laplacian operator and $(\Delta^2\phi_l)_o$ represents the error in the computed value of c_l at cell center o with respect to the value at o obtained using interpolation of c_l values in the neighboring cells [15]. For the lth species, $(\Delta^2\phi_l)_o$ is evaluated by

$$\left(\Delta^2\phi_l\right)_o = \sum_{k=1}^{N_k}[\alpha_k(\phi_l)_k] - \alpha_o(\phi_l)_o, \tag{24}$$

where

$$\alpha_o = \sum_{k=1}^{N_k}(\alpha_k). \tag{25}$$

In (24), N_k is the number of distinct discrete values $(\phi_l)_k \neq (\phi_l)_o$ used in the discrete approximation of the Laplacian and α_k are constant coefficients of the values $(\phi_l)_k$ that define the discrete approximation. As shown in (25), the coefficient of $(\phi_l)_o$ is α_o and α_o depends on the values α_k. In this work, a five-point stencil, shown in Fig. 1, is used to approximate $(\Delta^2\phi_l)_o$. The boxes in this figure represent the values $(\phi_l)_k$, and the number in a box is the value of the coefficient α associated with that box.

In (23), the parameter e_1 controls weighting of each cell volume V in relation to its size. If e_1 is negative then smaller cells will be weighted more than larger cells and vice versa. Thus, choosing e_1 to be negative may cause smooth flow features to be under-resolved. The parameter $e_2 \geq 0$ provides control over the rate of change of the cell volumes in the grid. If $e_2 > 0$ then evacuation of grid nodes from regions of uniform concentration will be inhibited and grid orthogonality will be promoted. The parameter $w_{\min}$ is the minimum allowable weight function value and is typically set such that $10 \times$ *machine zero* $\leq w_{\min} \leq 1$. Larger values of $w_{\min}$ are chosen if grid adaptation is only needed in regions with prominent solution features. The effects of e_1, e_2, and $w_{\min}$ on grid adaptation can be seen in Laflin [15]. In this work, the values of e_1 and e_2 were chosen to be -1.0 and 0.0, respectively.

As defined in Eq. (23), the weight w_o at cell center o would include information on the interpolation error at o in concentration of each of the species l and, therefore, would be

responsive to resolution needs of each of these species at o. However, in a flow with reacting species, the concentrations of these species may differ by several orders of magnitude. Therefore, each of the species-specific components $(\Delta^2\phi_l)_o$ in Eq. (23) needs to be scaled such that (1) the interpolation error $(\Delta^2\phi_l)_o$ is relatively independent of the magnitude of c_l, (2) the scaled $(\Delta^2\phi_l)_o$ adequately represents the resolution requirements at o for each of the species l, and (3) the scaling process is responsive to the dynamic changes in species concentrations. Taking these requirements into consideration, a weight function suitable for use in computations involving chemistry can be developed. This development is described below.

Construction of a weight function with the above considerations starts with determination of the species-specific SIERRA interpolation error at cell center i, j, given by

$$(\Delta^2\phi_l)_{i,j} = |c_l|_{i-1,j} + c_l|_{i+1,j} + c_l|_{i,j-1} + c_l|_{i,j+1} - 4 \cdot c_l|_{i,j}|. \tag{26}$$

As mentioned above, the value of the parameter e_2 shown in Eq. (23) is chosen to be zero. Therefore, for example, $c_l|_{i+1,j}$ in Eq. (26) is not weighted by $(V_{i+1,j}/V_{i,j})^{e_2}$, as required in Eq. (23).

The concentrations of various species can differ by several orders of magnitude and, in general, would contain computational noise resulting from finite-precision machine calculations. The effects of this noise need to be removed from the SIERRA interpolation errors given by Eq. (26) before these errors are normalized and rescaled. In this work, the interpolation errors are adjusted to remove the computational noise. For each of the species, this adjustment involves normalizing the interpolation error at each cell center by the average value of the species concentration over the domain and setting the normalized value to zero in case this value is less than or equal to $1. \times 10^{-3}$. The adjustment is

$$c_l^{\text{avg}} = \left(\sum_{i,j} c_l|_{i,j}\right) \Big/ nn; \qquad nn = \text{number of grid cells} \tag{27}$$

$$\begin{aligned}(\Delta^2\phi_l)_{i,j}^{\text{adjusted}} &= (\Delta^2\phi_l)_{i,j} \big/ c_l^{\text{avg}} \,\forall\, (\Delta^2\phi_l)_{i,j} \big/ c_l^{\text{avg}} > 1.E-03 \\ &= 0 \,\forall\, (\Delta^2\phi_l)_{i,j} \big/ c_l^{\text{avg}} \le 1.E-03. \end{aligned} \tag{28}$$

The adjusted interpolation error given by Eq. (28) is normalized using the maximum value over the domain:

$$(\Delta^2\phi_l)_{i,j}^{\text{norm}} = (\Delta^2\phi_l)_{i,j}^{\text{adjusted}} \big/ w_{\text{norm}}^l; \qquad w_{\text{norm}}^l = MAX\big((\Delta^2\phi_l)_{i,j}^{\text{adjusted}}\big). \tag{29}$$

This normalization process scales the range of each of the $(\Delta^2\phi_l)_{i,j}^{\text{norm}}$ to be between 0 and 1 and, therefore, satisfies requirement (1) given above.

Note that the presence of a relatively large range in $(\Delta^2\phi_l)$ over the entire grid would reflect that the spatial distribution of the concentration of species l requires significant resolution. Since the process of normalization given above compresses the range of each $(\Delta^2\phi_l)$, it becomes necessary to restore each of these ranges by re-scaling.

First a linear combination, $wc_{i,j}$, of the species-specific interpolation errors is formed at each of the cell centers,

$$wc_{i,j} = \sum_l (\Delta^2\phi_l)_{i,j}^{\text{norm}}. \tag{30}$$

Then this linear combination is re-scaled using

$$wc_{i,j}^{\text{rescaled}} = (wc_{i,j} - wc_{\min})\frac{((\Delta^2\phi)_{\max} - w_{\min})}{(wc_{\max} - wc_{\min})} + w_{\min}, \tag{31}$$

where

$$wc_{\min} = MIN(wc_{i,j}), \qquad wc_{\max} = MAX(wc_{i,j}) \tag{32}$$

$$(\Delta^2\phi)_{\max} = MAX\left(w_{\text{norm}}^{l}\right) \forall l. \tag{33}$$

The re-scaling scheme utilizing Eqs. (31) through (33) adjusts the range of values (maximum to minimum) of the $wc_{i,j}^{\text{rescaled}}$ to be between $w_{\min}$ and $(\Delta^2\phi)_{\max}$. Since $(\Delta^2\phi)_{\max}$ represents the largest SIERRA interpolation error value for all species over the entire grid, this re-scaling scheme ensures that the resolution requirements of all species over the entire grid are represented in the weight function. Thus, requirement (2) given above is satisfied. Further, the normalization and re-scaling process depends on the time-dependent, species-specific SIERRA interpolation errors and, therefore, satisfies requirement (3) given above.

Note that in Eq. (31) the lower end of the range of $wc_{i,j}^{\text{rescaled}}$ is fixed at $w_{\min}$. Consequently, $w_{\min}$ can be used to control the degree of adaptation (or the amount of grid movement). Finally, note that the normalizing and re-scaling procedure described above requires only the selection of a value for $w_{\min}$ by the user. This value is chosen based on the experiments conducted to obtain an acceptable preadapted grid with nodes clustered around any solution field features, prior to the first time step in a simulation.

Using the $wc_{i,j}^{\text{rescaled}}$, the weight function at a cell center i, j is given by [see Eq. (23)]

$$w_{i,j} = V_{i,j}^{1+e_1}\left|wc_{i,j}^{\text{rescaled}}\right|. \tag{34}$$

The weight function resulting from Eq. (34) may result in highly sheared or skewed grids in which large cell volumes may exist next to small cell volumes. These large volumes can adversely affect the accuracy of solution calculations in the next time step. Hence, it is desirable to obtain smooth grids without highly sheared or skewed cells. Therefore, the weight function resulting from Eq. (34) is smoothed by applying the discretized Laplacian operator to it [36].

Repositioning of grid nodes. In the adaptive grid procedure described here, repositioning of the grid nodes is accomplished by a center-of-mass scheme, originally proposed by Eiseman [37]. In this scheme, a grid node is repositioned such that its position coincides with the center-of-mass of a local cluster of cells, with mass distribution over the grid being defined by the weight function. Using this scheme, the new position coordinates of grid node o are given by

$$\mathbf{P}_o^{\text{new}} = \left(\sum_{i=1}^{4} w_i \mathbf{P}_i\right) \Big/ \sum_{i=1}^{4} w_i. \tag{35}$$

In (35), $\mathbf{P}_i, i = 1, \ldots, 4$, are the position coordinates in physical space of the centers of the cells that are local to the grid node o and $w_i, i = 1, \ldots, 4$, are the weights associated with these cells. Note that in a two-dimensional grid, node o lies over the center of mass of any four contiguous cells.

Solution redistribution. In DSAGA–PPM, a solver-independent solution field redistribution procedure developed in Laflin [15] is employed. In this procedure, the solution field is fixed with respect to an inertial frame while a control volume, $\hat{\Omega}$, is allowed to move arbitrarily through the spatial domain. To describe the motion of $\hat{\Omega}$ through the inertially fixed solution field, a fictitious grid time, t_g, is introduced. The moving control volume is a function of grid time $\hat{\Omega} = \hat{\Omega}(t_g)$, but is not a function of the physical time, $\hat{\Omega} \neq \hat{\Omega}(t)$. Conversely, the solution field is a function of the physical time, $c_l = c_l(t)$, but is not a function of grid time, $c_l \neq c_l(t_g)$. Using these concepts, Reynolds transport theorem reduces to a *conservative interpolation equation* that is suitable for numerical application [15]. This equation is

$$(\bar{c}_l V)^{n_g+1} = (\bar{c}_l V)^{n_g} + \sum_{p=1}^{4} \left(\hat{V}_p|_{n_g}^{n_g+1}\right)\left(\bar{c}_{l_p}|_{n_g}^{n_g+1}\right), \qquad l = 1, \ldots, N, \tag{36}$$

where $\bar{c}_l = \bar{c}_l(t_g)$ is the average value of c_l over the cell volume $V = V(t_g)$, $\hat{V}_p|_{n_g}^{n_g+1}$ is the volume swept by the cell side p during movement between grid time levels n_g and $n_g + 1$, and $\bar{c}_{l_p}|_{n_g}^{n_g+1}$ is the average value of c_l in $\hat{V}_p|_{n_g}^{n_g+1}$.

Equation (36) is used to compute the interpolated cell-averaged values for the dependent variables, $\bar{c}_l^{n_g+1}$, $l = 1, \ldots, N$. To use (36), the values of the volumes V^{n_g}, V^{n_g+1}, and $\hat{V}_p|_{n_g}^{n_g+1}$; values of cell-averaged dependent variables before grid movement, $\bar{c}_l^{n_g}$, $l = 1, \ldots, N$; and values of $\bar{c}_{l_p}|_{n_g}^{n_g+1}$, $p = 1, \ldots, 4$, $l = 1, \ldots, N$ are needed. The volumes V^{n_g}, V^{n_g+1}, and $\hat{V}_p|_{n_g}^{n_g+1}$ are computed using the grid node coordinates before and after grid movement. Values $\bar{c}_l^{n_g}$, $l = 1, \ldots, N$ are available either as initial conditions or as a result of the time advancement of the solution discussed in Section 3.1. However, the values $\bar{c}_{l_p}|_{n_g}^{n_g+1}$, $p = 1, \ldots, 4$, $l = 1, \ldots, N$ need to be determined. In two-dimensional calculations, the cell and sweep volumes can be calculated exactly and, therefore, the accuracy of determination of $\bar{c}_l^{n_g+1}$ is entirely dependent on the accuracy of determination of $\bar{c}_{l_p}|_{n_g}^{n_g+1}$, $p = 1, \ldots, 4$, $l = 1, \ldots, N$. An error analysis of $\bar{c}_l^{n_g+1}$, $l = 1, \ldots, N$, conducted by Laflin [15] shows that to ensure accurate determinations of $\bar{c}_{l_p}|_{n_g}^{n_g+1}$, $p = 1, \ldots, 4$, $l = 1, \ldots, N$, (1) grid-node movements need to be restricted and (2) a higher order scheme needs to be used for computing $\bar{c}_{l_p}|_{n_g}^{n_g+1}$, $p = 1, \ldots, 4$, $l = 1, \ldots, N$. Accordingly in DSAGA–PPM, an interim-step procedure [15] is used to increase the accuracy of solution interpolation while allowing for an arbitrary amount of grid movement, and a higher order scheme, PPM, is used to compute $\bar{c}_{l_p}|_{n_g}^{n_g+1}$, $p = 1, \ldots, 4$, $l = 1, \ldots, N$.

In the interim-step procedure, the grid time step, $\Delta t_g = (t_g^{n_g+1} - t_g^{n_g})$, is divided into M smaller interim steps, δt_g. If the change in position coordinates of a grid node o over Δt_g is $\Delta \mathbf{X}_o = \mathbf{X}_o^{n_g+1} - \mathbf{X}_o^{n_g}$ then the change in position coordinates of this grid node over δt_g is given by $\delta \mathbf{X}_o = (\Delta \mathbf{X}_o)/M$. In DSAGA–PPM, the value of M is chosen such that during any interim step, the movement of any cell side is restricted to one-half of the corresponding cell length. The interpolated cell-averaged values for the dependent variables, $\bar{c}_l^{n_g+1}$, $l = 1, \ldots, N$, are obtained by M iterative applications of

$$\bar{c}_l^{n_g+\beta_m} = \frac{1}{V^{n_g+\beta_m}}\left((\bar{c}_l V)^{n_g+\beta_{m-1}} + \sum_{p=1}^{4} \left(\hat{V}_p|_{n_g+\beta_{m-1}}^{n_g+\beta_m}\right)\left(\bar{c}_{l_p}|_{n_g+\beta_{m-1}}^{n_g+\beta_m}\right)\right), \qquad l = 1, \ldots, N. \tag{37}$$

In (37), m is the interim-step counter such that $m = 1, \ldots, M$; $\beta_m = m/M$ with $\beta_o = 0$.

Grid convergence. The center-of-mass scheme, depicted by (35), is a discrete approximation of Poisson's equation [15]. Therefore, an iterative process is used to obtain adequately converged grid node positions. In each time step, the following steps are taken in order: (1) weights are computed using the SIERRA approach described above; (2) grid nodes are moved using (35); (3) the solution is redistributed as described above; and (4) a check is made on grid convergence. If the grid is converged within a specified tolerance, then the solution is advanced through a new time step, otherwise weights are recomputed and the grid movement–solution redistribution procedure is repeated. In DSAGA–PPM, a limit, δ, is set on the maximum movement of grid nodes relative to largest cell side in the starting Cartesian grid such that

$$\frac{\mathrm{MAX}|\Delta \mathbf{X}_{i,j}|}{\mathrm{MAX}(\Delta x, \Delta y)_{\text{starting Cartesion grid}}} \leq \delta \,\forall\, i, j \tag{38}$$

where $\Delta \mathbf{X}_{i,j}$ is the change in position coordinates of the node i, j. If (38) was satisfied, then the grid was considered to be converged.

Preadaptation. In an air-quality simulation, the domain being modeled will, in general, contain regions with relatively large gradients in species concentrations. Such gradients may result from complex interactions between emissions from sources, meteorological conditions, and atmospheric chemistry. A simulation using DSAGA–PPM would start with a uniform distribution of grid nodes and would modify this distribution based on the spatial resolution requirements of the various species. Consequently, using the starting uniform grid with initial gradients in species concentration can result in an inaccurate calculation of the solution field in the first time step. In order to remedy this potentiality, a preadaptation step has been included in the DSAGA–PPM algorithm. In this step, the starting uniform grid is preadapted to regions with initial concentration gradients before the solution field is calculated in the first time step. In general, this preadaptation is accomplished by computing weights based on the initial solution field, moving the grid nodes, checking grid convergence, and stopping the preadaptation process once the grid is converged, as described in the previous sections. There could, however, be situations in which the initial solution field may not contain any concentration gradients but sources in the domain start emitting at the beginning of the first time step. In such situations, emissions from sources are assumed to occur before the first time step. The preadaptation process is then completed based on the gradients resulting from these emissions, and the solution field is reinitialized with background concentrations.

4. ADVECTION TESTS

The accuracy of results produced by an AQM depends to a large measure on the accuracy of computation of the advection process, for two reasons. First, horizontal transport of pollutant species is dominated by advection. Second, advection affects the accuracy of the species concentration data available at the beginning of each chemistry integration step and these data can have a significant influence on the results of atmospheric chemistry calculations. Thus, it is very important that the advection component of an AQM be computed accurately.

Tests with model advection problems with known analytical solutions were used in this work to characterize the improvements in accuracy achieved by using DSAGA–PPM. For

each of the model problems, DSAGA–PPM and corresponding SGA–PPM results were obtained on a CRAY T90 vector-processing machine and then compared.

In this work, following Smolarkiewicz and Rasch [38], the errors in the numerical solutions were characterized by the following normalized measures:

$$\text{EMIN} = \left(c^{\min} - c_{\mathrm{e}}^{\min}\right)\Big/ c_{\mathrm{e}}^{\max}, \quad \text{the normalized error in the minimum value of the solution field;} \tag{39}$$

$$\text{EMAX} = \left(c^{\max} - c_{\mathrm{e}}^{\max}\right)\Big/ c_{\mathrm{e}}^{\max}, \quad \text{the normalized error in the maximum value of the solution field;} \tag{40}$$

$$\text{EMAS} = \left(\sum cV - \sum c_{\mathrm{e}} V\right)\Big/ \sum c_{\mathrm{e}} V, \quad \text{the normalized error in the total mass; and} \tag{41}$$

$$\text{ERMS} = \sqrt{\sum [(c - c_{\mathrm{e}})^2 V]\Big/ \sum V}, \quad \text{the root-mean-square error of the solution field.} \tag{42}$$

In the expressions above, min and max refer to the global minimum and maximum values, respectively, in the solution field; the subscript e stands for the analytical solution; and $\sum$ is the discrete integral over the modeled region. As defined above, EMIN indicates the maximum undershoot in the solution field; EMAX measures the damping or overshoot of the initial solution peak; EMAS indicates the extent of mass conservation; and ERMS indicates overall error in the solution field. In the definitions of EMAS and ERMS, the integrals are weighted by cell volume to account for the significant cell volume variation resulting from grid adaptation.

4.1. *A Rotating Cone*

In this test, the solid-body rotation of a conical distribution is examined in a two-dimensional region that is 42 km long in both the x and y directions. The region is initially discretized with 43×43 grid nodes, spaced uniformly. The initial conditions consist of a conical concentration distribution (cone) with a base radius of 4 km and a peak concentration of 100 units, centered at coordinates (26.5 km, 21.5 km) in the region. The background concentration in the region is 5 units. This cone is advected in the counterclockwise direction around the center of the region by a wind with a constant angular velocity of 0.1 rad/h. The analytical solution of the above problem is a solid-body rotation of the cone [17]. Note that the cone presents a relatively complex solution feature with a steep gradient at the side and discontinuities at the apex and at the juncture with the background. It was desirable to rigorously test the ability of DSAGA–PPM to represent steep gradients and produce monotonic solution fields. Therefore, the peak concentration was increased to 100 units, in contrast to those used in other works [18, 39, 40], and the background concentration was selected to be greater than zero.

In each of the simulations the Courant Friedrich Levy number (CFL) was set at 0.4. In the DSAGA–PPM simulation, the values of the grid adaptation–related parameters $w_{\min}$, δ, and number of smoothing iterations were set at $8. \times 10^{-3}$, $3. \times 10^{-2}$, and 15, respectively. Additionally, in the interim step procedure the movement of any cell side was restricted to one-half of the corresponding cell length. The values of the adaptation parameters were determined through numerical experiments and were selected to obtain a good preadapted grid.

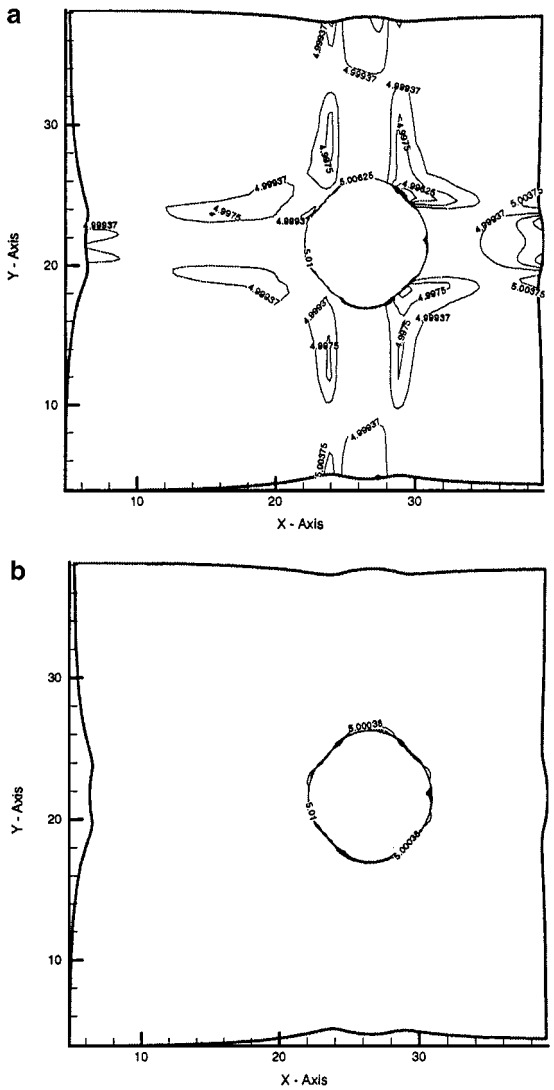

FIG. 2. Solution field (a) with oscillations and (b) without oscillations.

Initially, a version of the adaptive grid code without the correction for numerical sources, described in Section 3.1, was used. Shown in Fig. 2a is the solution after two time steps, obtained with this code. This solution clearly reflects oscillations in the concentration field. The code was then modified to incorporate the correction for numerical sources described in Section 3.1. Shown in Fig. 2b is the solution after two time steps, obtained with the modified code. This solution is free of oscillations.

Finally, one revolution of the cone is followed using SGA–PPM and DSAGA–PPM. The results shown in Figs. 3a and 3b reflect that 87% of the peak is retained using DSAGA–PPM while only 61% of the peak concentration is retained using SGA–PPM. This illustrates that the adaptive grid locally reduces numerical diffusion and thereby provides better peak concentration maintenance. As shown in Fig. 3c, the nodes of the adaptive grid at the end of the simulation are clustered in the cone and have tended to align with the discontinuity at the background juncture.

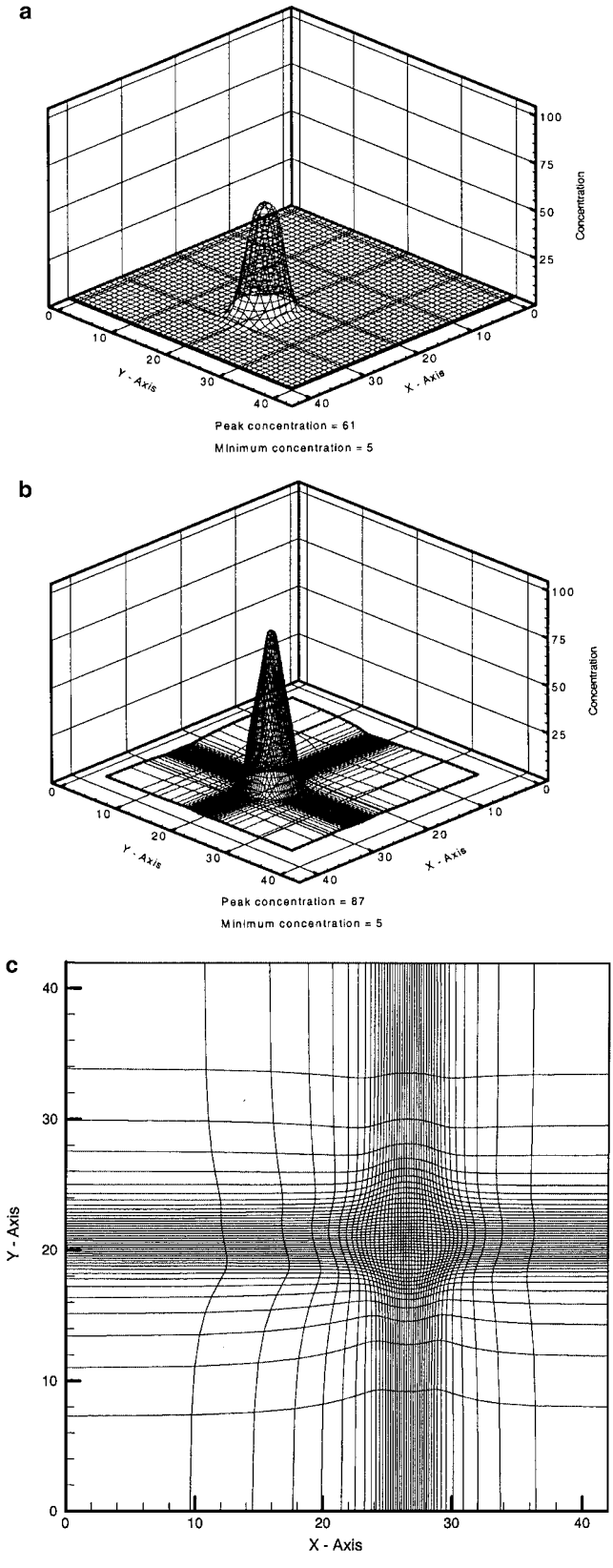

FIG. 3. Rotating cone results obtained with (a) SGA–PPM and (b) DSAGA–PPM. (c) Adaptive grid after one revolution of the cone.

TABLE I
A Summary of Error Characteristics for the Simulations of a Rotating Cone

Solution	EMIN	EMAX	EMAS	ERMS
SGA–PPM (43 × 43 nodes)	−1.1E-14	−3.9E-01	−5.7E-07	1.5E-01
DSAGA–PPM (43 × 43 nodes)	1.6E-09	−1.3E-01	−1.3E-04	2.9E-02
SGA–PPM (203 × 203 nodes)	0	−1.0E-01	4.4E-13	4.8E-02

The error measures introduced above were used to further compare the static and adaptive grid solutions. The values of these measures are shown in Table I. The small value of EMIN for the DSAGA–PPM solution indicates that this algorithm did not introduce appreciable undershoots as a consequence of improving resolution. Further, the value of EMAS for the DSAGA–PPM solution reflects that the algorithm has maintained global mass with reasonable accuracy (about 0.01% loss). The values of EMAX for the SGA–PPM and DSAGA–PPM solutions reflect that the adaptive grid maintains solution features better than the static grid (compare peak values of 87 and 61, seen in Figs. 3b and 3a). Finally, the magnitudes of ERMS reflect that compared to the static grid solution, the adaptive grid solution has much less overall (root-mean-square or rms) error.

4.2. *Four Rotating Cones*

In the second two-dimensional test, the ability of DSAGA–PPM to resolve multiple concentration distributions is examined. In this test, four conical distributions (cones), located in a 42 × 42-km two-dimensional region, are advected in the counterclockwise direction by a wind with a constant angular velocity of 0.1 rad/h. The region is initially discretized with 43 × 43 grid nodes, spaced uniformly. The initial conditions comprise four pollutant cones, each with a base radius of 4 km and a peak concentration of 100 units, centered at coordinates (30.5, 21.5), (20.5, 30.5), (11.5, 20.5), and (21.5, 11.5) in the two-dimensional region. The background concentration in the region is 5 units.

In each of the simulations the CFL was set at 0.4. In the DSAGA–PPM simulation, the values of the grid adaptation–related parameters $w_{\min}$, δ, and number of smoothing iterations were set at $1. \times 10^{-3}$, 4×10^{-2}, and 15, respectively. Additionally, in the interim step procedure the movement of any cell side was restricted to one-half of the corresponding cell length. These values of the adaptation parameters were determined through numerical experiments and were selected to obtain a good preadapted grid.

The results after one full revolution of the cones, obtained without and with grid adaptation, are shown in Figs. 4a and 4b, and the corresponding error characteristics are displayed in Table II. A comparison of the error characteristics for the static and the adaptive grid

TABLE II
A Summary of Error Characteristics for the Simulations of Multiple Rotating Cones

Solution	EMIN	EMAX	EMAS	ERMS
SGA–PPM (43 × 43 nodes)	−7.4E-13	−4.6E-01	−3.3E-04	5.0E-01
DSAGA–PPM (43 × 43 nodes)	3.9E-06	−2.3E-01	6.3E-04	2.7E-01
DSAGA–PPM (85 × 85 nodes)	4.4E-8	−1.3E-01	−8.7E-05	9.6E-02
SGA–PPM (115 × 115 nodes)	−5.9E-9	−2.0E-01	−2.0E-07	2.4E-01

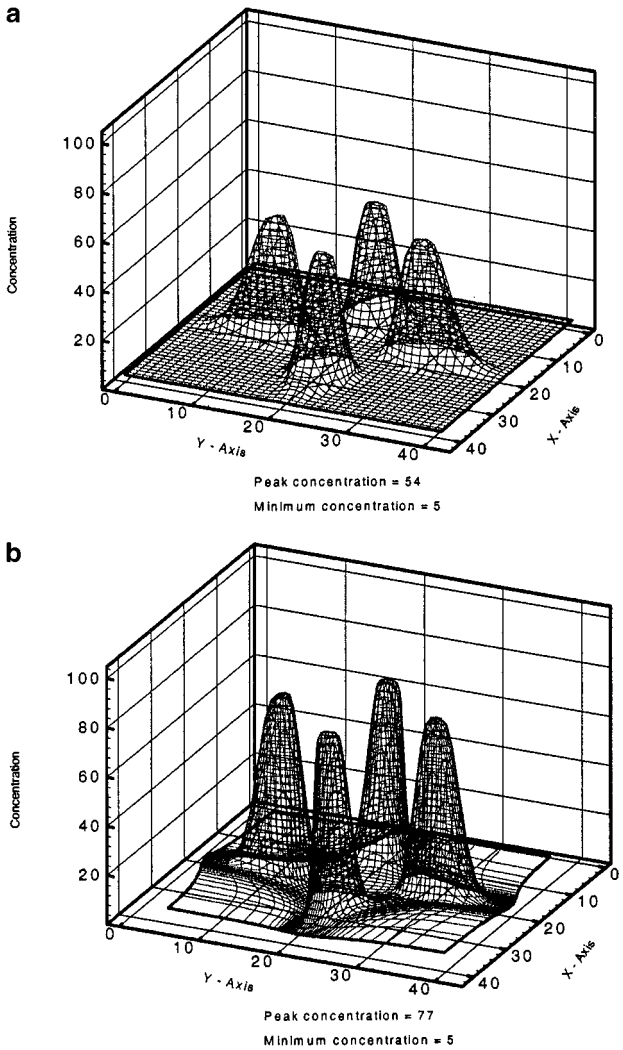

FIG. 4. Results for four cones obtained with (a) SGA–PPM and (b) DSAGA–PPM.

solutions yields observations similar to those seen in the previous test case. The DSAGA–PPM (1) does not introduce appreciable spurious oscillations (indicated by the low value of EMIN); (2) maintains solution features better than SGA–PPM (e.g., peak of 77 versus 54 seen in Figs. 4b and 4a); (3) maintains global mass with reasonable accuracy (0.06% gain in global mass indicated by EMAS); and (4) provides a solution with less overall (rms) error than the static grid solution. The adaptive grid at the end of the computation, shown in Fig. 5, is clustered around the solution field features. These observations reflect that DSAGA–PPM is able to resolve multiple zones of interest and provide an accurate solution field.

The accuracy of any DSAGA–PPM solution would appear to depend in part on the number of nodes available for resolving the solution field features. To examine this hypothesis, a simulation of one revolution of the cones was completed using an adaptive grid with 85×85 grid nodes. This refined grid provided approximately the same number of grid nodes per solution feature (cone) as the single-cone test case conducted with 43×43 grid nodes. The

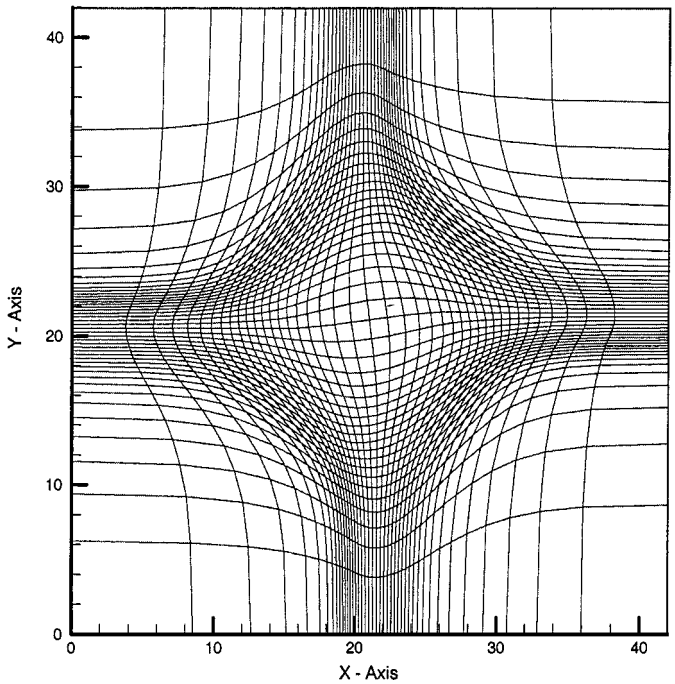

FIG. 5. Adaptive grid after one revolution of the cones.

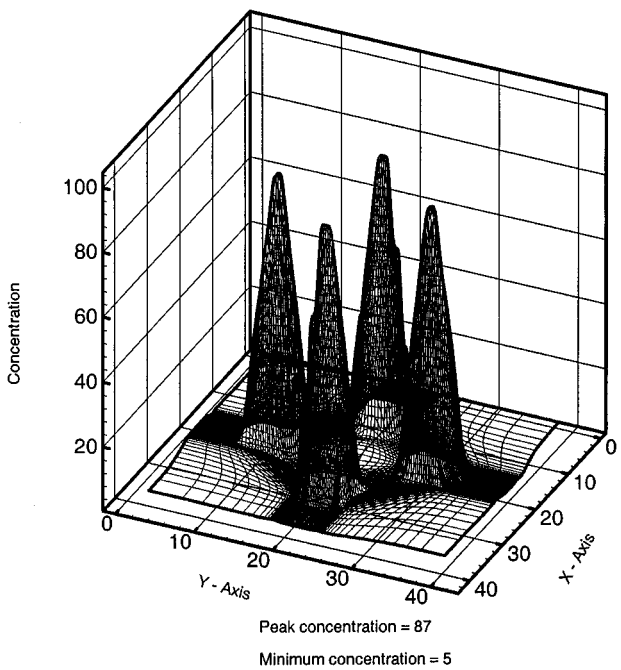

FIG. 6. DSAGA–PPM solution obtained using a refined grid with 85×85 nodes.

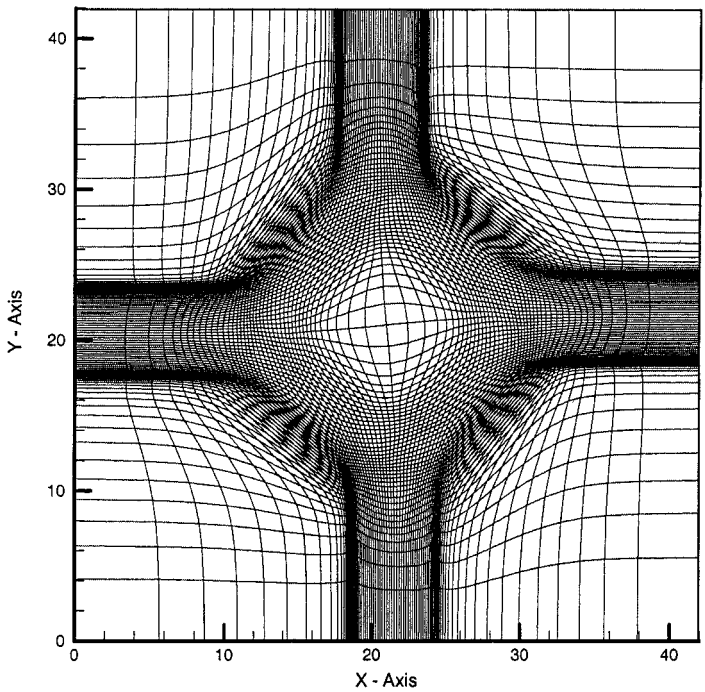

FIG. 7. Adaptive refined (85 × 85 nodes) grid after one revolution of the cones.

grid adaptation parameters and CFL used in this simulation were identical to those used in the simulation with 43 × 43 nodes.

The DSAGA–PPM result after one full revolution of the cones is shown in Fig. 6, and the corresponding error characteristics are displayed in Table II. A comparison of the error characteristics for the two DSAGA–PPM solutions and Figs. 4b and 6 reflects solution quality improvement as the number of grid nodes available for adaptation increases. Furthermore, by maintaining approximately the same number of grid nodes per solution feature, after one revolution the values of the cone peaks in the DSAGA–PPM solution for the four cones are identical to the value of the peak for the single cone (see Figs. 6 and 4b). Finally, the adaptive refined grid at the end of the simulation, shown in Fig. 7, exhibits better resolution of the cones than the adaptive grid with 43 × 43 nodes (see Fig. 5). A comparison of Figs. 5 and 7 also reveals that the grid spacings in these figures have the same number of nodes in the borders of the domain and in the center. This reflects that DSAGA–PPM has responded intelligently by distributing the extra nodes in the refined grid in the regions requiring resolution.

5. REACTING POLLUTANT PUFF

An air-quality model includes coupled transport and nonlinear chemistry processes. These processes can interact and significantly alter the distributions of species in time. For example, the shape of a puff of pollutants can change constantly due to changes in the gradients and extrema of the pollutant distributions in the puff. Since chemistry associated with air pollution is, in many cases, nonlinear, the accuracy of the AQM results depends

TABLE III
Initial Concentrations of Species in the Pollutant Puff

Species	Background (molecules/cm^3)	Peak (molecules/cm^3)
CO	1.00E+12	
H_2O	2.50E+15	
HC	2.50E+09	1.00E+11
HCHO	1.25E+10	5.00E+11
HO_2	1.00E+06	
NO	2.50E+09	1.00E+11
NO_2	2.50E+09	1.00E+11
O(1*d*)	1.00E-03	
O_3	5.00E+11	
OH	1.00E+05	
RO_2	1.00E+06	

on the local resolution of the concentration changes. Averaging of these changes over large cells may greatly underestimate local concentrations, resulting in lower production of expected products. Consequently, the ability of DSAGA–PPM to adapt to such changes can only be seen and evaluated in tests with model problems incorporating chemistry. In this section DSAGA–PPM is tested on a model problem in which a pollutant puff undergoes advection and chemistry. This model problem is similar to that used by Odman and Russell [1] and Chock and Winkler [41] in their evaluations of static nested grid applications.

This model test problem consists of advecting a pollutant puff that initially contains collocated conical distributions of nitric oxide (NO), nitrogen dioxide (NO_2), formaldehyde (HCHO), and lumped hydrocarbons (HC) with the peak concentrations shown in Table III. Each of these initial distributions has a base radius of 4 km and is centered at coordinates (26.5, 21.5) km in a domain that is 42 km long in both the x and y directions. The initial background concentration for each of the species is also shown in Table III. These concentrations are identical to those used by [1]. The puff is advected in the counterclockwise direction around the center of the two-dimensional domain by a wind with a constant angular velocity. During advection, the species in the chemical mechanism undergo chemical transformations that result in formation of ozone (O_3).

Following Hov *et al.* [42] and Odman and Russell [1], a simple chemical mechanism describing production and destruction of tropospheric O_3 is used in this work. This mechanism is shown in Table IV. The solar zenith angle, θ, that appears in the photolysis reactions of Table IV is held constant at 71.5°. This value corresponds to the average zenith angle experienced during an equinox day at the equator [1].

The chemistry-induced changes in the peak and background values of NO, NO_2, HCHO, and O_3 species are shown in Fig. 8 for a time period of 150 s. From this figure it is evident that rapid transformations of the species occur in the initial 150 s. Accordingly, the ability of DSAGA–PPM to respond to resolution requirements generated by rapid nonlinear chemical transformations is examined by completing one revolution of the puff in 150 s.

In the absence of diffusion, the analytically exact solution of advection and chemistry processes can be assembled as follows. If only advection is occurring, then in any time period δt the concentration c_1 of a species is moved from a location $\mathbf{X}_1$ to another location

TABLE IV
A Simplified Chemical Mechanism for Photochemical Production of Ozone

Reaction	Rate
$HC + OH \rightarrow 4RO_2 + 2HCHO$	$k_1 = 6.0 \times 10^{-12}$
$HCHO + h\nu \rightarrow 2HO_2 + CO$	$J_2 = 7.8 \times 10^{-5} e^{-0.87/\cos\theta}$
$RO_2 + NO \rightarrow NO_2 + HCHO + HO_2$	$k_3 = 8.0 \times 10^{-12}$
$NO + HO_2 \rightarrow NO_2 + OH$	$k_4 = 8.3 \times 10^{-12}$
$NO_2 + h\nu \rightarrow NO + O_3$	$J_5 = 1.0 \times 10^{-2} e^{-0.39/\cos\theta}$
$NO + O_3 \rightarrow NO_2 + O_2$	$k_6 = 1.6 \times 10^{-14}$
$O_3 + h\nu \rightarrow O_2 + O(1d)$	$J_7 = 1.9 \times 10^{-4} e^{-1.9/\cos\theta}$
$O(1d) + H_2O \rightarrow 2OH$	$k_8 = 2.3 \times 10^{-11}$
$NO_2 + OH \rightarrow HNO_3$	$k_9 = 1.0 \times 10^{-11}$
$CO + OH \rightarrow CO_2 + HO_2$	$k_{10} = 2.9 \times 10^{-13}$

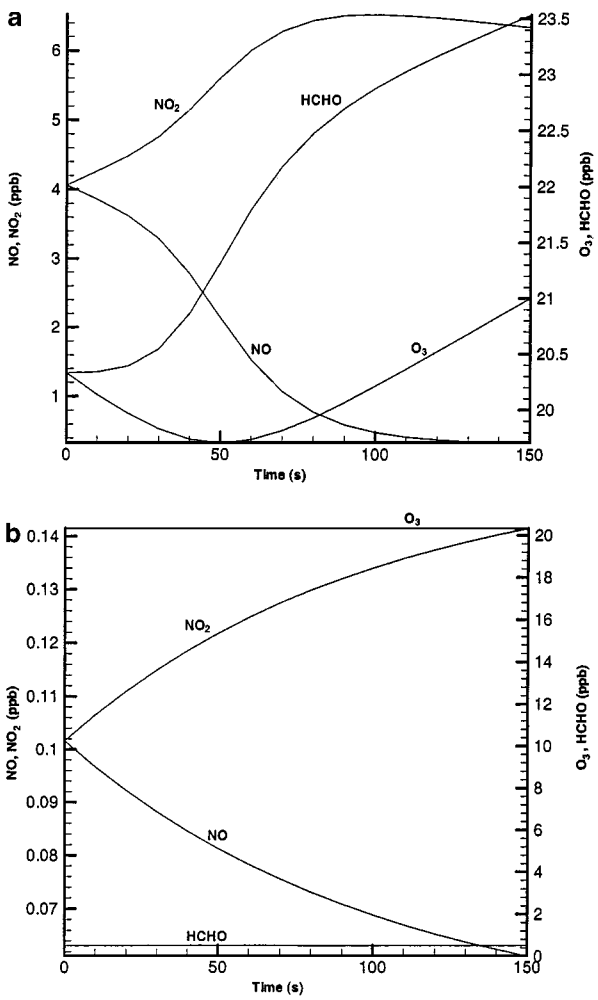

FIG. 8. Changes in concentrations of selected species over 150 s (a) peak concentrations, (b) background concentrations.

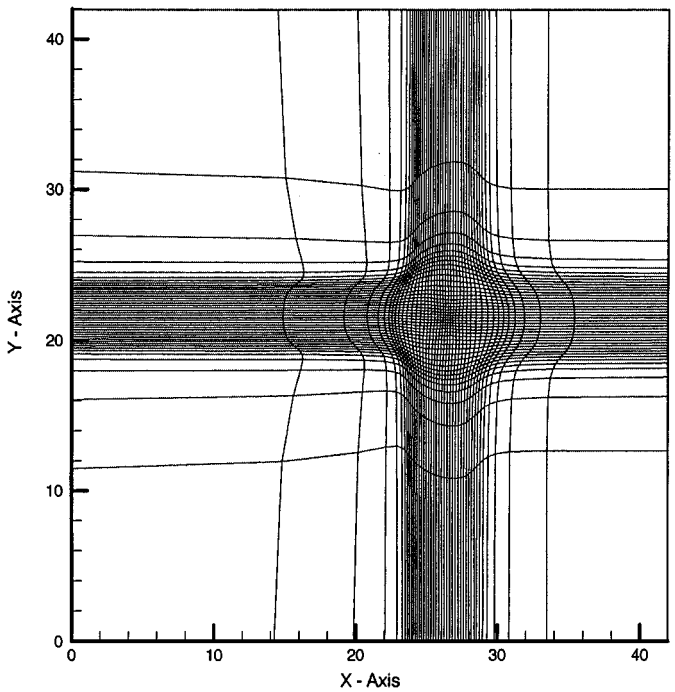

FIG. 9. Preadapted grid reflecting clustering of nodes in and around the pollutant puff.

$\mathbf{X}_2$. On the other hand, if only chemistry is occurring, then the initial concentration c_1 of the species at $\mathbf{X}_1$ is simply converted to c_2. Therefore, if both advection and chemistry are present, then the concentration at $\mathbf{X}_2$ should be c_2.

For each of the simulations, a reference solution was created using the above procedure. Note that a reference solution on a finite-volume grid is simply the discretized representation of the corresponding analytically exact solution on that grid. Consequently, the appropriate reference solutions were used to assess the accuracy of the simulation results.

For the numerical simulations, the domain was initially discretized with 43×43 uniformly spaced grid nodes and the CFL was set at 0.4. In the DSAGA–PPM simulation, the values of the grid adaptation–related parameters $w_{\min}$ and δ and the number of smoothing iterations were set at $1. \times 10^{-5}$, $3. \times 10^{-2}$, and 10, respectively. Additionally, in the interim step procedure described above, the movement of any cell side was restricted to one-half of the corresponding cell length. These values of the adaptation parameters were determined through numerical experiments and were selected to obtain a preadapted grid with cells closely clustered in and around the puff. This grid is shown in Fig. 9.

Presented in Figs. 10 through 17 are the SGA–PPM and DSAGA–PPM solutions as well as the corresponding reference solutions, for NO, NO_2, O_3, and HCHO, after one revolution of the puff. Figure 18 shows the resulting adaptive grid after 150 s with grid nodes clustered in and around the puff.

It is interesting to qualitatively compare the results of the simulation with the chemistry results presented in Figs. 8a and 8b. Per these figures, between 0 and 150 s, the peak and background NO concentrations drop continuously, and at 150 s the peak concentration is higher than the background concentration. The results in Figs. 10 and 11 are consistent with these observations. In each of these figures, the distribution of NO

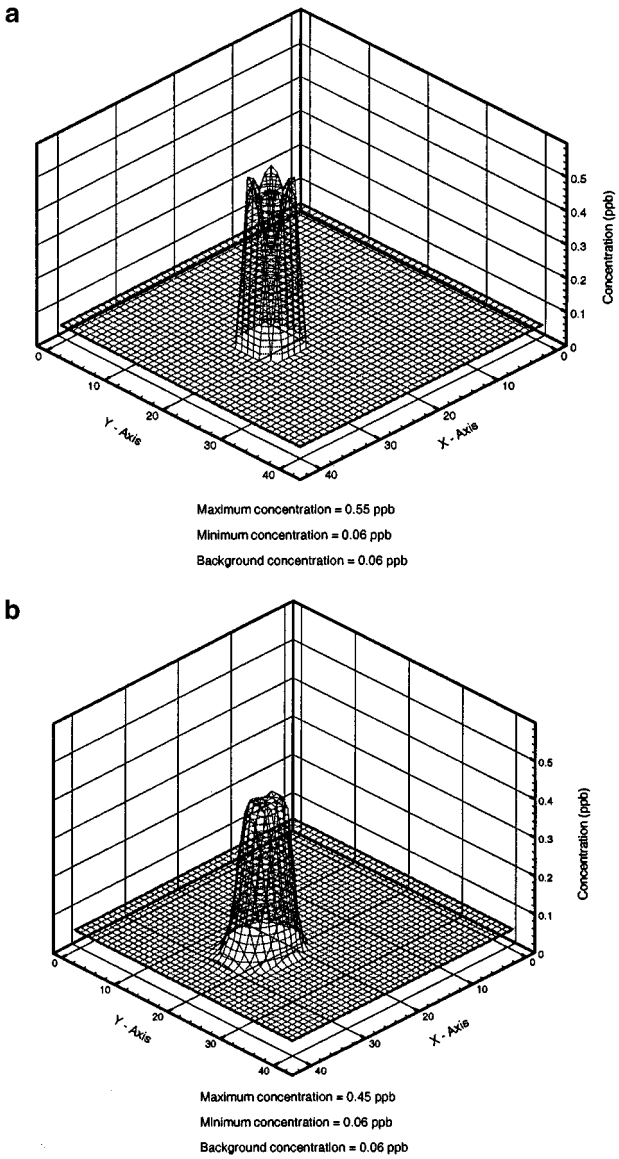

FIG. 10. NO distribution after 150 s in the pollutant puff (a) exact solution on uniform grid, (b) SGA–PPM solution.

concentration has a valley (formed by the folding-in of the initial peak), the floor of the valley is higher than the background, and the background is much lower than the starting value of about 4 ppb. However, while the NO peak in the DSAGA–PPM solution is similar to that in its reference solution, the same is not the case for the SGA–PPM solution.

The valley in the O_3 profile in Fig. 8a reflects that the peak concentration of O_3 drops below the background concentration for about 50 s and then starts growing. The O_3 profile in Fig. 8b indicates that the background concentration of O_3 remains virtually constant. These figures also reflect that at 150 s the peak concentration of O_3 is higher than the background concentration. The results in Figs. 14a, 15a, and 15b are consistent with these observations;

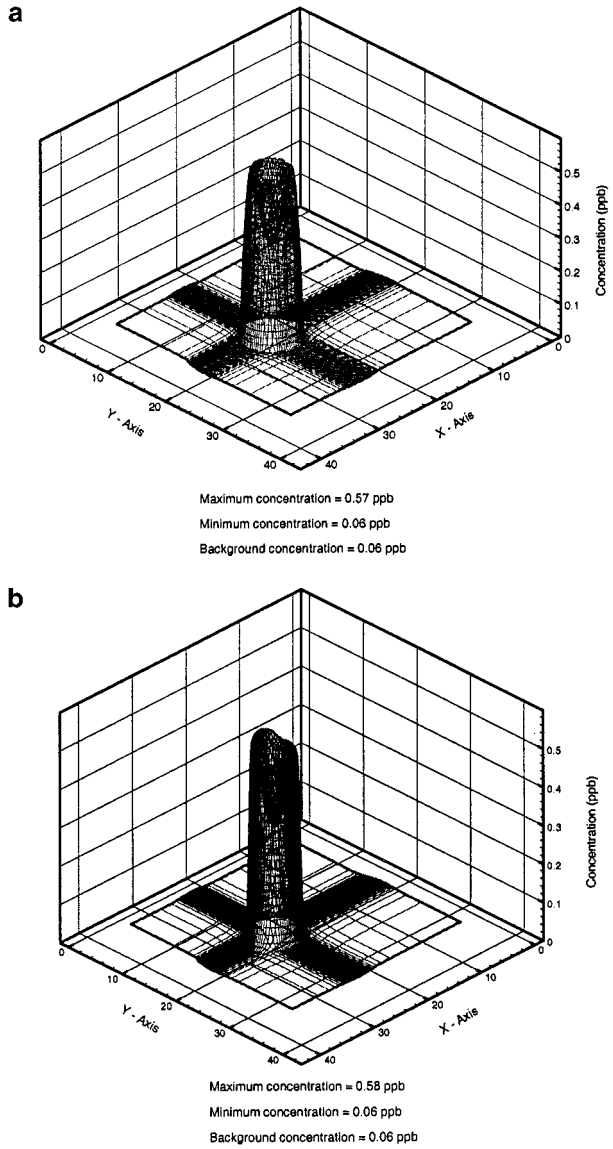

FIG. 11. NO distribution after 150 s in the pollutant puff (a) exact solution on adaptive grid, (b) DSAGA–PPM solution.

each of these figures depicts a peak and a valley in concentration of O_3. However, the SGA–PPM result in Fig. 14b reflects that the peak O_3 concentration is below the background concentration. The concentrations of NO_2 and HCHO, shown in Figs. 8a and 8b, do not undergo any peak inversions. Correspondingly, the results in Figs. 12, 13, 16, and 17 do not exhibit any peak inversions.

The above qualitative observations reveal that some of the fine-scale solution structures for NO and O_3 are captured in the DSAGA–PPM simulation but not in the SGA–PPM simulation.

As discussed above, the species profiles resulting from chemical interactions can contain peaks and valleys. Thus, to obtain an accurate assessment of such profiles, it is important

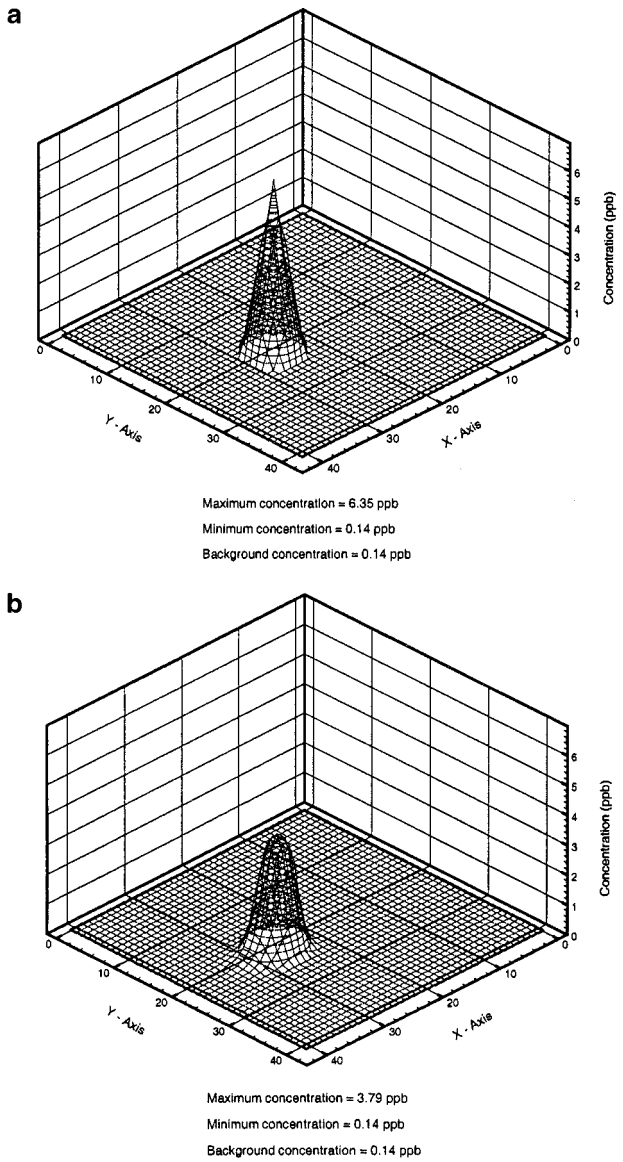

FIG. 12. NO_2 distribution after 150 s in the pollutant puff (a) exact solution on uniform grid, (b) SGA–PPM solution.

to be able to characterize these peaks and valleys. Accordingly, some of the error measures introduced earlier were modified to provide such characterizations and facilitate comparisons between the static and the adaptive grid solutions. These modified measures are

$$\text{EVALLEY} = \frac{c^{\text{valley}} - c_e^{\text{valley}}}{c_e^{\text{valley}}} \tag{43}$$

$$\text{EPEAK} = \frac{c^{\text{peak}} - c_e^{\text{peak}}}{c_e^{\text{peak}}}. \tag{44}$$

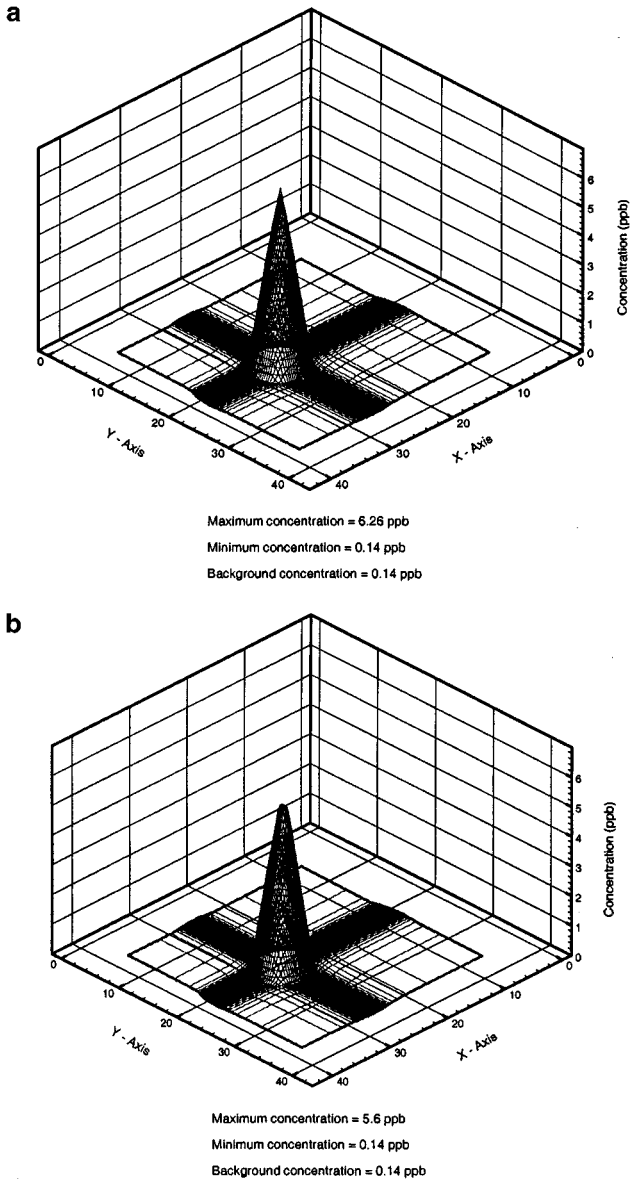

FIG. 13. NO_2 distribution after 150 s in the pollutant puff (a) exact solution on adaptive grid, (b) DSAGA–PPM solution.

In addition to these measures, the error in mass, EMAS, and the root-mean-square error, ERMS, introduced above, are also used to characterize the quality of SGA–PPM and DSAGA–PPM solutions. Note that in the expressions (43) and (44), the subscript *e* stands for the reference solution described above. Thus for each of the DSAGA–PPM and SGA–PPM simulations, EVALLEY, EPEAK, EMAS, and ERMS are computed based on the corresponding reference solutions.

For the chemical species, initialized as conical concentration distributions, values of the above error measures are shown in Tables V and VI. In Table V, the much lower values of

TABLE V
EVALLEY and EPEAK for the 150-s Simulation of the Puff

	SGA–PPM		DSAGA–PPM	
Species	EVALLEY (%)	EPEAK (%)	EVALLEY (%)	EPEAK (%)
HC	2.5E-07	−38.0	4.3E-05	−10.0
HCHO	9.1E-07	−43.0	4.4E-05	−12.0
NO	7.5	−18.2	8.0E-05	1.8
NO_2	−1.2E-04	−40.3	−5.6E-05	−10.5
O_3	0.60	−3.4	−5.0E-02	−1.2

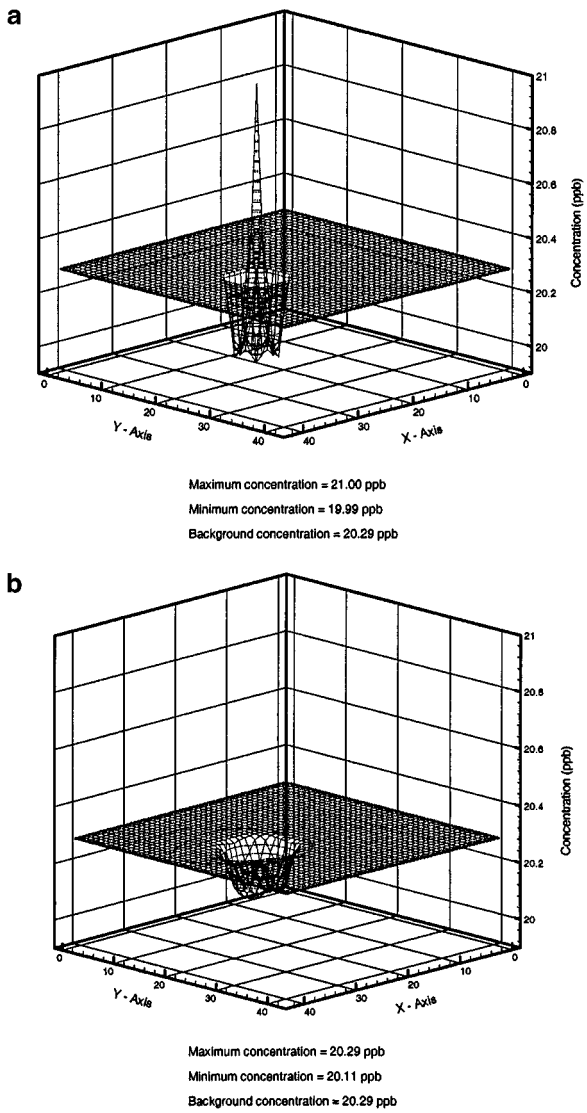

FIG. 14. O_3 distribution after 150 s in the pollutant puff (a) exact solution on uniform grid, (b) SGA–PPM solution.

TABLE VI
EMAS and ERMS for the 150-s Simulation of the Puff

	SGA–PPM		DSAGA–PPM	
Species	EMAS (%)	ERMS (%)	EMAS (%)	ERMS (%)
HC	0.23	30.0	0.03	3.9
HCHO	−0.26	31.0	0.03	3.9
NO	1.80	28.0	1.8E-03	3.6
NO_2	−0.18	32.0	0.04	3.9
O_3	−5.1E-03	0.15	−7.1E-05	0.03

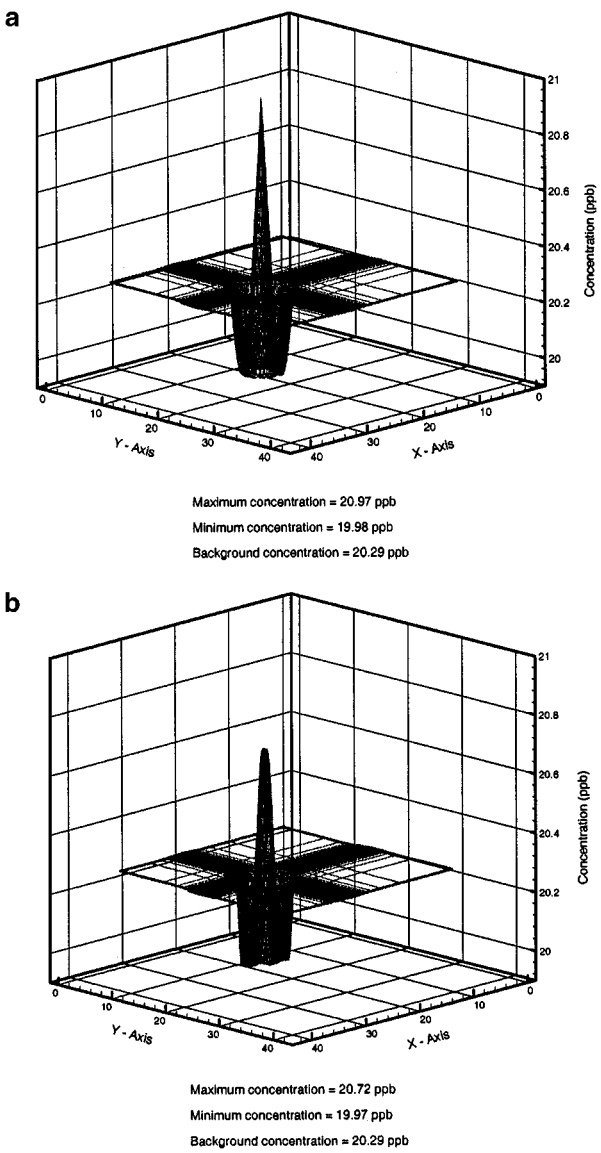

FIG. 15. O_3 distribution after 150 s in the pollutant puff (a) exact solution on adaptive grid (b) DSAGA–PPM solution.

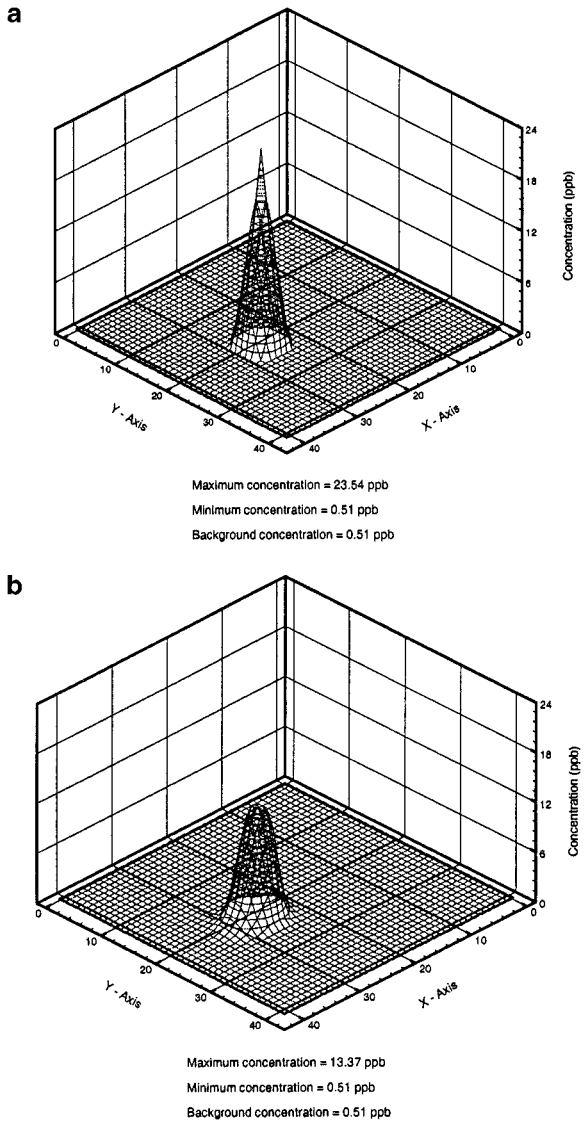

FIG. 16. HCHO distribution after 150 s in the pollutant puff (a) exact solution on uniform grid, (b) SGA–PPM solution.

EVALLEY (%) for the DSAGA–PPM solution for NO and O_3 indicate that the adaptive grid algorithm predicts these valleys much better than the corresponding SGA solution (also see Figs. 10, 11, 14, and 15).

The lower values of EPEAK (%) for the DSAGA–PPM solutions, compared to those for the SGA–PPM solutions, for HC, HCHO, NO, NO_2, and O_3 reflect that the adaptive grid predicts solution peaks better than the static grid. Note that a negative EPEAK (%) value indicates that the peak is lower than that in the corresponding reference solution. Note also that a positive value of EVALLEY and a negative value of EPEAK for O_3 in the SGA–PPM solution reflect that the valley in this solution is higher and the peak is lower than these features in the corresponding reference solution. These findings are consistent with the fact

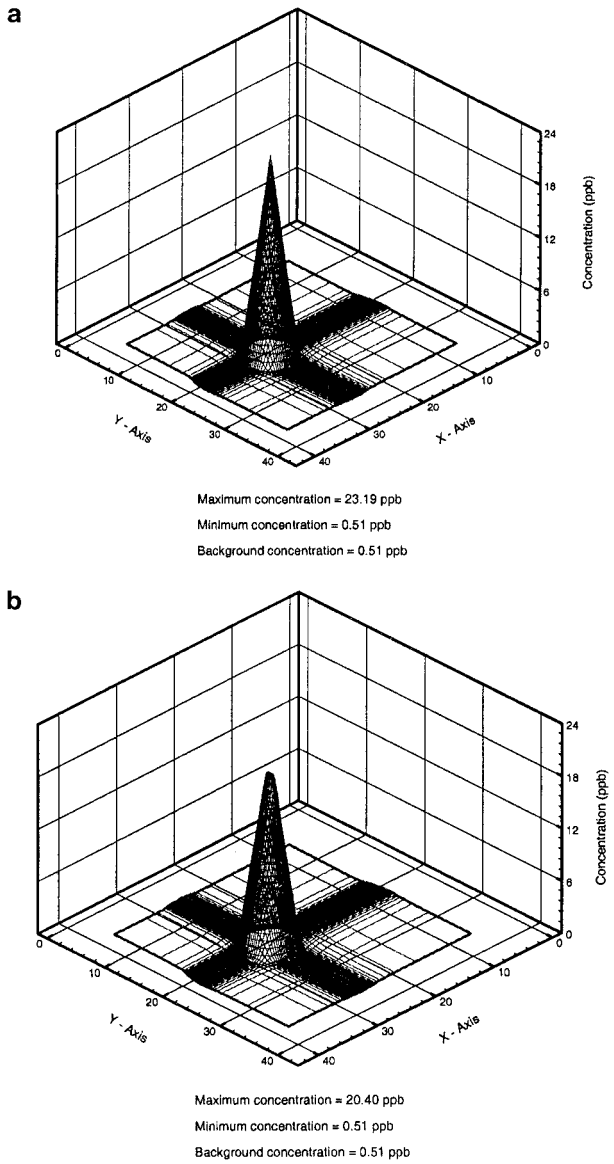

FIG. 17. HCHO distribution after 150 s in the pollutant puff (a) exact solution on adaptive grid, (b) DSAGA–PPM solution.

that the O_3 profile is not as well developed in the SGA-PPM solution as it should be (see Figs. 14 and 15).

As seen in Table VI, the low values of EMAS (%) for the DSAGA–PPM and SGA–PPM solutions reflect that the global mass in each of these solutions compares well with the global mass in the corresponding reference solution. Finally, the much lower magnitudes of ERMS for the DSAGA–PPM solutions reflect that, compared to the static grid solution, the adaptive grid solutions have undergone less numerical diffusion and have less overall (rms) error.

The results of this test indicate that DSAGA–PPM can follow rapid chemical transformations more accurately than the corresponding static grid algorithm.

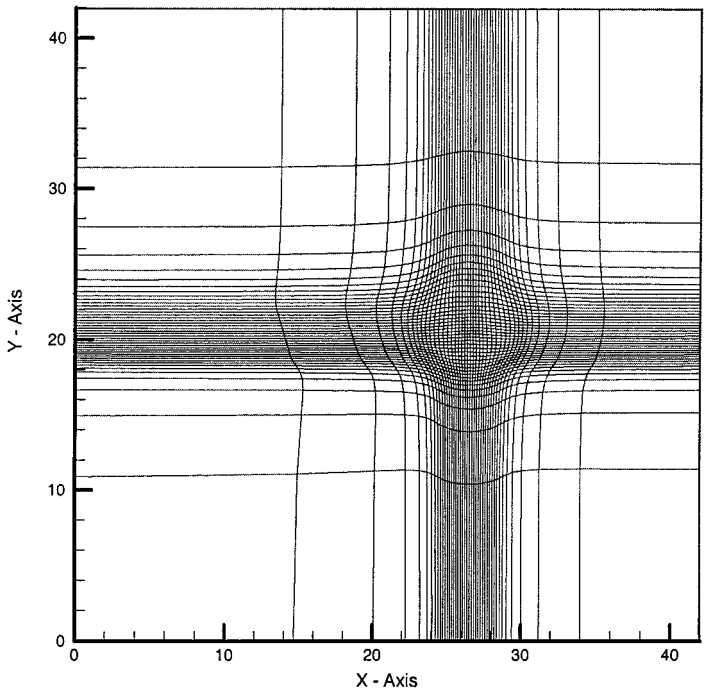

FIG. 18. Adaptive grid after 150 s reflecting clustering of nodes in and around the pollutant puff.

6. COMPUTATIONAL PERFORMANCE

As discussed in the previous sections, a solution with a specific level of accuracy can be obtained using either DSAGA–PPM with relatively few grid nodes or a refined grid SGA–PPM with many more nodes. However, to assess the practicality of using DSAGA–PPM in air-quality simulations, it is useful to obtain an indication of its computational performance.

It is well known that the chemistry calculations consume the majority of the CPU time used in an air-quality simulation [43]. Therefore, the reacting pollutant puff model problem described in this work is appropriate for an assessment of DSAGA–PPM's computational performance. Shown in Table VII are the CPU times and minimum cell sizes associated with three simulations of this model problem. Two of these simulations used SGA–PPM and DSAGA–PPM on a grid with 43×43 grid nodes; the third simulation used SGA–PPM on a refined grid with 127×127 nodes. All these simulations were conducted on a CRAY T90 machine using a CFL of 0.4, and the code used performs about 320 million

TABLE VII
CPU Times Associated with Simulations of Reacting Pollutant Puff

Simulation	Grid nodes	Minimum cell size (area units)	CPU time (s)
SGA–PPM	43×43	1	33.0456
DSAGA–PPM	43×43	2.2E-02–3.2E-02	299.0532
Refined grid SGA–PPM	127×127	1.1E-01	18,726.9411

floating-point operations per second for an average vector length of about 40 units. Note that in the DSAGA–PPM simulation, the smallest cell size changed with time and ranged between 0.022 and 0.032 area units.

As shown in Table VII, the cell size in each of the static grid simulations (SGA–PPM or refined grid SGA–PPM) is substantially larger than the smallest cell size in the DSAGA–PPM simulation. This indicates that the extent of resolution of spatial distributions of species is much greater in the DSAGA–PPM simulation. Therefore, the grid in the refined grid SGA–PPM simulation would have to be refined further to obtain a solution with accuracy comparable to that achieved in the DSAGA–PPM simulation. These statements are corroborated by the results obtained in these two simulations. As an example, the values of the error metrics EVALLEY, EPEAK, EMAS, and ERMS for O_3 obtained using refined-grid SGA–PPM are -0.18×10^{-2}, -0.17×10^{-1}, -0.53×10^{-5}, and 0.46×10^{-3}, respectively. The corresponding values obtained using DSAGA–PPM are -0.75×10^{-3}, -0.12×10^{-1}, -0.71×10^{-6}, and 0.30×10^{-3}. Clearly the metric values are lower for the DSAGA–PPM result, thereby indicating that the DSAGA–PPM solution is more accurate than the refined grid SGA–PPM solution. These results reflect that refined grid SGA–PPM takes about 63 times more CPU time than DSAGA–PPM to provide a less accurate solution.

While it is difficult to generalize the above indications, the results do reflect that DSAGA–PPM has the potential to provide accurate air-quality simulations at significant cost savings. In the near future, a more detailed assessment of DSAGA–PPM's performance will be deduced from realistic air-quality simulations.

7. CONCLUSIONS

In this paper the theoretical foundations of a dynamic adaptive-grid algorithm (DSAGA–PPM) are described. DSAGA–PPM incorporates a weight function formulation that is designed to resolve chemistry-induced changes in species concentrations. Since the grid movement and solution interpolation steps in this algorithm are independent of time advancement of the solution field, the algorithm can be used efficiently with the operator-split governing equations used in air-quality modeling.

Testing of DSAGA–PPM using a two-dimensional model problem with a rotating conical distribution shows that the algorithm can resolve dynamic solution features that have sharp gradients and discontinuities. An error analysis conducted on the results of this model problem obtained with DSAGA–PPM reveals that representation of extrema and overall error are greatly improved without any significant loss in mass conservation.

The capability of the algorithm to simultaneously resolve multiple features in a solution field was examined in another model problem with four rotating conical distributions. As for the single rotating conical distribution problem, the DSAGA–PPM simulation results in improved solution compared to that achieved with the corresponding static grid. When the number of mesh nodes in the four-cones problem is increased to approximately four times that used for the one-cone problem, the DSAGA–PPM simulation provides solution resolution and accuracy comparable to those achieved in the adaptive grid solution of the one-cone problem.

DSAGA–PPM was also applied to a model problem with a rotating pollutant puff undergoing atmospheric chemistry. This model problem was solved for very rapid chemical changes occurring during transport. Also, reference solutions were computed to facilitate comparisons between results obtained using DSAGA–PPM and the corresponding static

grid algorithm (SGA–PPM). The DSAGA–PPM results, as evaluated by error measures, are much closer to the reference solutions than the SGA–PPM results. This demonstrates that DSAGA–PPM responds to solution resolution requirements generated by rapid nonlinear chemical transformations and transport of distributed atmospheric pollutants.

Finally, a significant computational efficiency advantage may be possible if DSAGA–PPM is used in an AQM with chemistry. The results for the reacting pollutant puff model problem indicate that an SGA–PPM simulation using a refined grid with 127×127 nodes takes about 63 times more CPU time than a DSAGA-PPM simulation on a grid with 43×43–nodes, but provides a less accurate solution. Therefore, it is concluded that DSAGA–PPM has the potential to greatly improve AQM accuracy or efficiency or both.

The capability of DSAGA–PPM to provide accurate solutions of coupled transport and nonlinear chemistry processes has been investigated in additional model problems. Currently, these results are being processed for future publication. In addition, efforts are underway to integrate DSAGA–PPM with a currently used AQM. Subsequently, a more detailed assessment of DSAGA–PPM's performance will be deduced from realistic air-quality simulations.

ACKNOWLEDGMENT

This research was supported by the U.S. Environmental Protection Agency under Cooperative Agreement CR822053-01 Task 4.2.3.

REFERENCES

1. M. T. Odman and A. G. Russell, A multiscale finite element pollutant transport scheme for urban and regional modeling, *Atmos. Environ.* **25A**, 2385 (1991).
2. M. T. Odman, R. Mathur, K. Alapaty, R. K. Srivastava, D. S. McRae, and R. J. Yamartino, Nested and adaptive grids for multiscale air quality modeling, in *Next Generation Environmental Models and Computational Methods*, edited by Delic and Wheeler (Soc. for Industr. & Appl. Math., Philadelphia, 1957), p. 59.
3. G. S. Dietachmayer and K. K. Droegemeier, Application of continuous dynamic grid adaption techniques to meteorological modeling. I. Basic formulation and accuracy, *Mon. Weather Rev.* **120**, 1675 (1992).
4. A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, A high resolution adaptive projection method for regional atmospheric modeling, in *Next Generation Environmental Models and Computational Methods*, edited by Delic and Wheeler (Soc. for Industr. & Appl. Math., Philadelphia, 1957), p. 69.
5. W. C. Skamarock and J. B. Klemp, Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flow, *Mon. Weather Rev.* **121**, 788 (1993).
6. A. Tomlin, M. Berzins, J. Ware, J. Smith, and M. J. Pilling, On the use of adaptive gridding methods for modelling chemical transport from multi-scale sources, *Atmos. Environ.* **31**, 2945 (1997).
7. R. A. Benson and D. S. McRae, *A Three-Dimensional Dynamic Solution-Adaptive Mesh Algorithm*, AIAA Paper 90-1566 (AIAA, June 1990).
8. R. A. Benson and D. S. McRae, A solution adaptive mesh algorithm for dynamic/static refinement of two and three dimensional grids, in *Proceedings of the Third International Conference on Numerical Grid Generation in Computational Field Simulations, Barcelona, 1991*, p. 185.
9. R. A. Benson and D. S. McRae, Time accurate simulations of unsteady flows with a dynamic solution-adaptive algorithm, in *Proceedings of the Fourth International Conference on Numerical Grid Generation in Computational Field Simulations*, Swansea, *1994*, p. 573.
10. R. A. Benson, D. S. McRae, and J. R. Edwards, *Numerical Simulations Using a Dynamic Solution-Adaptive Grid Algorithm with Applications to Unsteady Internal flows*, AIAA Paper 92-2719 (AIAA, June 1992).
11. R. A. Benson and D. S. McRae, *Numerical Simulations of the Unstart Phenomena in a Supersonic Inlet/Diffuser*, AIAA Paper 93-2239 (AIAA, June 1993).

12. R. A. Benson, *Development of a Time-Accurate Solution Algorithm Coupled to a Dynamic Solution- Adaptive Grid Algorithm with Application to Generic Inlet/Diffuser Configurations*, Ph.D. thesis (North Carolina State University, Raleigh, 1994.)
13. R. K. Srivastava, D. S. McRae, and M. T. Odman, Application of a solution adaptive grid technique to air quality modeling, in *Proceedings of the Fifth International Conference on Numerical Grid Generation in Computational Field Simulations, Vol. II, Starkville, Mississippi, 1996*, edited by Soni, Thompson, Hauser, and Eiseman, p. 1241.
14. K. R. Laflin and D. S. McRae, Three-dimensional viscous flow computations using near-optimal grid redistribution, in *Proceedings of First AFOSR Conference on Dynamic Motion CFD, Rutgers University, New Brunswick, NJ, 1996*.
15. K. R. Laflin, *Solver Independent r-Refinement Adaptation for Dynamic Numerical Simulations*, Ph.D. thesis (North Carolina State University, Raleigh, 1997).
16. P. Collela and P. R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, *J. Comput. Phys.* **54**, 174 (1984).
17. R. K. Srivastava, *An Adaptive Grid Algorithm for Air Quality Modeling*, Ph.D. thesis (North Carolina State University, Raleigh, 1998).
18. G. J. McRae, W. R. Goodin, and J. H. Seinfeld, Numerical solution of the atmospheric diffusion equation for chemically reacting flows, *J. Comput. Phys.* **45**, 1 (1982).
19. S. D. Reynolds, P. M. Roth, and J. H. Seinfeld, Mathematical modeling of photochemical air pollution—I: Formulation of the model, *Atmos. Environ.* **7**, 1033 (1973).
20. A. Kasahara, Various vertical coordinate systems used for numerical weather prediction, *Mon. Weather Rev.* **102**, 509 (1974).
21. O. B. Toon, R. P. Turco, D. Westphal, R. Malone, and M. S. Liu, A multidimensional model for aerosols: Description of computational analogs, *J. Atmos. Sci.* **45**, 2123 (1988).
22. P. J. Roache, *Computational Fluid Dynamics* (second ed., Hermosa Publications, Albuquerque, 1976).
23. D. A. Anderson, J. C. Tannehill, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer* (first ed., Hemisphere, New York, 1984).
24. W. L. Oberkampf, Domain mappings for the numerical solution of partial differential equations, *Int. J. Numer. Meth. Eng.* **10**, 211 (1976).
25. M. Vinokur, Conservative equations of gasdynamics in cartesian coordinate systems, *J. Comput. Phys.* **14**, 105 (1974).
26. N. N. Yanenko, *The Method of Fractional Steps* (Springer-Verlag, New York, 1971).
27. R. L. Carpenter, Jr., K. K. Droegemeier, P. R. Woodward, and C. E. Hane, Application of the piecewise parabolic method (PPM) to meteorological modeling, *Mon. Weather Rev.* **118**, 586 (1990).
28. M. T. Odman and A. G. Russell, Computational fluid dynamic techniques in air quality modeling in *Applied Computational Fluid Dynamics*, edited by V. K. Garg (Dekker, New York, 1998), p. 335.
29. R. G. Hindman, Generalized coordinate forms of governing fluid equations and associated geometrically induced errors, *AIAA J.* **20**, 1359 (1982).
30. T. H. Pulliam and J. L. Steger, Implicit finite difference simulations of three-dimensional compressible flow, *AIAA J.* **18**, 159 (1980).
31. P. D. Thomas and C. K. Lombard, *The Geometric Conservation Law—A Link Between Finite-Difference and Finite-Volume Methods for Flow Computation on Moving Grids*, AIAA Paper 78-1208 (AIAA, July 1978).
32. P. D. Thomas, *Numerical Method for Predicting Flow Characteristics and Performance of Nonaxisymmetric Nozzles—Theory*, NASA CR-3147 (NASA, September 1979).
33. T. P. Gielda and D. S. McRae, *Explicit Parabolized Solutions for Secant-Ogive-Cylinder-Boat-Tail and Finned Projectiles*, AIAA 87-2292 (AIAA 5th Applied Aerodynamics Conference, Monterey, CA, August 1987).
34. T. R. Young and J. P. Boris, A numerical technique for solving stiff ordinary differential equations associated with the chemical kinetics of reactive flow problems, *J. Phys. Chem.* **81**, 2424 (1977).
35. A. C. Hindmarsh and G. D. Byrne, *EPISODE: An Experimental Program for the Integration of Systems of Ordinary Differential Equation Systems*, Lawrence Livermore Laboratory Report UCID-30112 (1975).

36. G. F. Carey, *Computational Grids: Generation, Adaptation, and Solution Strategies* (Taylor & Francis, Washington, D.C., 1997.)
37. P. R. Eiseman, Adaptive grid generation, *Comput. Meth. Appl. Mech. Eng.* **64**, 321 (1987).
38. P. K. Smolarkiewicz and P. J. Rasch, Monotone advection on the sphere: An Eulerian versus semi-Lagrangian approach, *J. Atmos. Sci.* **48**, 793 (1991).
39. A. Bott, A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes, *Mon. Weather Rev.* **117**, 1006 (1989).
40. P. K. Smolarkiewicz, A simple positive definite advection scheme with small implicit diffusion, *Mon. Weather Rev.* **111**, 479 (1983).
41. D. P. Chock and S. L. Winkler, A particle grid air quality modeling approach, 2. Coupling with chemistry, *J. Geophys. Res.* **99**, 1033 (1994).
42. O. Hov, Z. Zlatev, R. Berkowicz, A. Eliassen, and L. P. Prahm, Comparison of numerical techniques for use in air pollution models with non-linear chemical reactions, *Atmos. Environ.* **23**, 967 (1989).
43. L. K. Peters, C. M. Berkowitz, G. R. Carmichael, R. C. Easter, G. Fairweather, S. J. Ghan, J. M. Hales, L. R. Leung, W. R. Pennell, F. A. Potra, R. D. Saylor, and T. T. Tsang, The current state and future direction of eulerian models in simulating the tropospheric chemistry and transport of trace species: A review, *Atmos. Environ.* **29**, 189 (1995).