

Chapter 7

NUMERICAL TRANSPORT ALGORITHMS FOR THE COMMUNITY MULTISCALE AIR QUALITY (CMAQ) CHEMICAL TRANSPORT MODEL IN GENERALIZED COORDINATES

Daewon W. Byun,^{*} Jeffrey Young,^{} and Jonathan Pleim^{**}**

Atmospheric Modeling Division
National Exposure Research Laboratory
U.S. Environmental Protection Agency
Research Triangle Park, NC 27711

M. Talat Odman⁺ and Kiran Alapaty

MCNC-Environmental Programs
P.O. Box 12889, 3021 Cornwallis Road
Research Triangle Park, NC 27709-2889

ABSTRACT

The transport processes in the atmosphere primarily consist of advection and diffusion, except for the mixing of pollutants by the parameterized subgrid-scale clouds. In this chapter, numerical algorithms for advection, vertical diffusion, and horizontal diffusion implemented in the Community Multiscale Air Quality (CMAQ) chemical transport models are discussed. To provide the CMAQ system with multiscale capability, we have formulated the transport processes, both advection and diffusion, in conservation (i.e., flux) forms for the generalized coordinate system. Therefore the numerical transport algorithms implemented in CMAQ will function under a wide variety of dynamical situations and concentration distribution characteristics. Users can not only choose transport algorithms from optional modules available in CMAQ, but also are encouraged to experiment with their own algorithms to test different numerical schemes for air quality simulations.

^{*} On assignment from the National Oceanic and Atmospheric Administration, U.S. Department of Commerce.
Corresponding author address: Daewon W. Byun, MD-80, Research Triangle Park, NC 27711.
E-mail: bdx@hpcc.epa.gov

^{**} On assignment from the National Oceanic and Atmospheric Administration, U.S. Department of Commerce.

⁺ Present Affiliation: Georgia Institute of Technology, Atlanta, GA.

7.0 NUMERICAL TRANSPORT ALGORITHMS FOR THE COMMUNITY MULTISCALE AIR QUALITY (CMAQ) CHEMICAL TRANSPORT MODEL IN GENERALIZED COORDINATES

In this chapter, we study numerical algorithms for the transport processes in a turbulent atmosphere. Many of the contents provided here are the results of a collaborative research project, the EPA's Cooperative Agreement CR822053-01 with MCNC-Environmental Programs (Exploratory Research on Air Quality Modeling Techniques: Research on Numerical Transport Algorithms for Air Quality Simulation Models), and other related in-house projects at EPA. Readers are referred to Alapaty et al. (1997), Byun (1999a, b), Byun and Lee (1999), and Odman (1998) for additional information.

In principal, the transport process consists of advection and diffusion that cause the movement and dispersion of pollutants in space and time. Transport of pollutants by the parameterized subgrid-scale cloud modules is not considered here. We have assumed that the transport of pollutants in the atmospheric turbulent flow field can be described by means of differential equations with appropriate initial and boundary conditions. In Eulerian air quality models, the transport process is modeled using numerical algorithms. These numerical algorithms for the advection and diffusion processes must satisfy several properties that are essential for making useful air quality simulations. As with all numerical methods, the numerical schemes for solving the transport equations must meet convergence conditions and correctly model the conservation, dissipation, and dispersion properties of the governing equations. A numerical scheme is said to be convergent if the solution approaches the true solution of the corresponding partial differential equation as the grid spacing and time-step size become infinitesimally small. Thus, a convergent numerical scheme can provide a numerical solution of any desired accuracy within finite precision bounds by reducing the grid spacing and the time-step size. For linear equations, consistency and stability are both necessary and sufficient conditions for convergence (Lax's equivalence theorem). In practice, machine precision and the computational resource availability limit the reduction of grid spacing and time-step size. Therefore, numerical errors associated with using limited grid spacing and time-step sizes must be of concern.

There have been many studies on the numerical advection algorithms used in air quality models. The reason it attracted so much attention is that the equation is hyperbolic in nature and spatial discretization of the solution generates a finite number of Fourier modes that travel at different speeds and leads to constructive and destructive interference. If the high wave-number Fourier modes are damped significantly, then numerical diffusion becomes prevalent. Solving the diffusion equation, on the other hand, is a lot safer because the stiffness matrix is diagonally dominant and the discretized solution is stable and sign preserving for a relatively wide range of conditions (Chock, 1999).

Transport processes are of central importance in turbulent flow studies, and in the literature there are numerous transport algorithms that have different numerical characteristics and varying degrees of accuracy and computational complexity. The skill needed here is to select appropriate numerical schemes that provide solutions with the desired accuracy at reasonable computational cost. This document does not intend to provide an extensive review of the transport algorithms used in air quality modeling. Instead, we describe several popular numerical schemes implemented in the Community Multiscale Air Quality Chemical Transport Model (CMAQ CTM or, hereafter, CCTM), expecting users to choose the algorithms appropriate to their own study objectives. We offer a few examples of good transport algorithms and present some key numerical characteristics users should look for. With this information, users can find the best algorithms through evaluation processes, and may even bring in their own algorithms to build a transport model for their applications. To provide the CMAQ system with the multiscale and multi-pollutant capabilities, we strive to incorporate schemes that can function under a wide variety of dynamic situations and distribution characteristics (e.g., distributions of different primary species and secondary species are quite distinct). Also, the schemes should be efficient in the use of computer time and storage. Selected numerical transport algorithms for horizontal and vertical advection and for vertical and horizontal diffusion are described below.

7.1 Numerical Advection Algorithms

Numerical advection algorithms for air quality models should satisfy several computational requirements.

- They should be free of mass conservation errors to accurately account for pollutant sources and sinks.
- They should have small numerical diffusion to minimize the spread of a signal in every direction and the smoothing of spatial gradients.
- They should also have small phase errors since disturbances that propagate at different speeds produce spurious oscillations.
- Given initial positive concentrations, the schemes should be positive-definite (i.e., they should not produce negative concentrations).
- They should be monotonic (i.e., they should not produce new extrema).

While it is essential that the schemes be positive-definite, this alone may not be sufficient because the monotonic property, for example, is just as desirable for air quality modeling.

Numerical algorithms have not been able to satisfy all the requirements listed above, and they are imperfect, with varying degrees of accuracy. Advection schemes with different properties introduce different errors, all of which are sources of uncertainty in air quality model predictions. Before recommending its use, it is critical to identify which of the computational properties a scheme possesses. Because an advection scheme with all the desired properties is not currently available, a user needs to select a scheme with the most desirable properties and greatest efficiency to meet the needs of the application.

7.1.1 Conservation Form Equation for Advection

The atmospheric advection process is expressed in conservation (flux) form as:

$$\left. \frac{\partial \phi_i^*}{\partial t} \right|_{adv} = -\nabla_{\xi} \bullet \left(\phi_i^* \widehat{\mathbf{V}}_{\xi} \right) - \frac{\partial(\phi_i^* \widehat{v}^3)}{\partial \widehat{x}^3} \quad (7-1)$$

where ϕ_i^* is the concentration of trace species i coupled with the coordinate Jacobian. Refer to Chapters 5 and 6 for the definition of symbols used in Equation 7-1. For convenience, the advection process is decomposed into horizontal and vertical advection processes, with the fractional time-step implementation:

$$\left. \frac{\partial \phi_i^*}{\partial t} \right|_{hadv} = -\nabla_{\xi} \bullet \left(\phi_i^* \widehat{\mathbf{V}}_{\xi} \right) = -\frac{\partial(\phi_i^* \widehat{v}^1)}{\partial \widehat{x}^1} - \frac{\partial(\phi_i^* \widehat{v}^2)}{\partial \widehat{x}^2} \quad (7-2)$$

$$\left. \frac{\partial \phi_i^*}{\partial t} \right|_{vadv} = -\frac{\partial(\phi_i^* \widehat{v}^3)}{\partial \widehat{x}^3} \quad (7-3)$$

where $\widehat{v}^1$, $\widehat{v}^2$ and $\widehat{v}^3$ are contravariant components of wind velocity. Splitting of the three-dimensional (3-D) advection into the horizontal and vertical components will lead some difficulties, such as the representativeness of the mass continuity and setting up of proper boundary conditions for non-orthogonal horizontal and vertical directions when simulating a region with complex topography.

Many models further split the horizontal advection equation in two directions and solve for two one-dimensional equations, one in each direction, using the solution of one as the initial condition of the other. We refer to this scheme as a one-dimensional (1-D) algorithm. Others solve the two-dimensional (2-D) form directly. Although using 1-D schemes is very common, it has been found that problems can arise due to this additional splitting (Flatoy, 1993, and Odman and Russell, 1993). Although 2-D schemes may be more desirable in this regard, fewer have been tested and they are often more difficult to implement and less computationally efficient than 1-D schemes. Also, there are general conditions in which the splitting scheme is actually more stable

and accurate than the non-splitting case for higher-order approximations because the splitting scheme intrinsically contains cross-spatial derivatives whereas the non-splitting scheme would not (Leith, 1965). Yanenko (1971) has shown that time-splitting is second-order accurate if the one-component advection operators commute. Alternating the sequence of operations would be quasi-second-order accurate in the case of non-commutativity (Chock, 1999). Here, only 1-D schemes will be discussed. When using appropriately interpolated contravariant wind components, the 1-D advection in the generalized coordinate system is equivalent to the 1-D equation in the Cartesian coordinate system. Therefore, it is sufficient here to discuss advection algorithms in Cartesian coordinates.

The 1-D advection equation written in the Cartesian coordinate system is:

$$\frac{\partial \phi}{\partial t} + \frac{\partial(u\phi)}{\partial x} = 0 \quad (7-4)$$

Equation 7-4 is the flux (or conservation) form and the quantity $F_x = u\phi$ is defined as the one-dimensional constituent flux. The flux form is a natural choice here because it is based on the continuity equation without any assumptions on the atmospheric dynamics. Maintaining the advection equation in flux form is key to providing transport schemes with multiscale and multi-pollutant capabilities. A flux-form discretization of Equation 7-4 with first-order accuracy in time results in:

$$\phi_j^{n+1} = \phi_j^n - \frac{\Delta t}{\Delta x_j} (F_{j+1/2}^n - F_{j-1/2}^n) \quad (7-5)$$

where $F_{j+1/2}^n$ and $F_{j-1/2}^n$ denote the advective fluxes through the faces of cell j , Δt is the time-step and $\Delta x_j = x_{j+1/2} - x_{j-1/2}$ is the horizontal grid spacing. To maintain numerical stability, and to accommodate other physical changes such as emissions input in a synchronized way, the time-step of 1-D advection should satisfy the Courant-Friedrich-Lewy (CFL) condition for hyperbolic equations:

$$\max_j (\beta_{j+1/2}) \leq 1 \quad (7-6)$$

where $\beta_{j+1/2} = |u_{j+1/2}| \frac{\Delta t}{\Delta x_j}$ is the Courant number for advection. This condition should be viewed

as a method defining a reference time scale for accommodating different physical processes in AQMs. If we do not consider the synchronization of mass injection to the cell through other physical processes, the CFL condition can be determined separately for individual advection schemes. For certain schemes the restriction can be significantly less than one. And for

advection alone, an implicit scheme may not have the Courant number restriction for stability (Chock, 1999).

7.1.2 Classification of Advection Schemes

Numerical advection schemes in the literature were developed using several different approaches (e.g., Chock and Dunker, 1983, and Chock, 1985, 1991). Following Rood (1987), we classify these schemes based on the methods used in their formulations. However, reviews in the literature may not capture the most recent developments in advection research. Depending on the methods used, the schemes may be classified as:

- Finite difference schemes;
- Finite volume schemes;
- Flux corrected schemes;
- Lagrangian Schemes;
- Finite element schemes; or
- Spectral schemes.

The distinction is somewhat arbitrary and only meant to convey the key intrinsic features of the scheme. Current trends in advection scheme development show a merging of the methods to take advantage of the most desirable properties of several schemes. For example, the Characteristic-Galerkin method (Childs and Morton, 1990) combines the best of the finite element and Lagrangian methods. Flux corrections are being used in the framework of finite element and spectral schemes (Löhner et al., 1987). Also, the classical finite difference schemes are being abandoned in favor of modern finite volume schemes. Refer to Odman (1998) for details of the classification.

7.1.3 Description of Advection Schemes in CCTM

In this section we describe the schemes that are available with the first release of the CCTM codes in the following order: the piecewise parabolic method (PPM), the Bott scheme (BOT), and the Yamartino-Blackman Cubic scheme (YAM).

Odman (1998) provides additional descriptions of the Smolarkiewicz scheme (SMO), the accurate space derivative scheme (ASD) (Chock, 1991, and Dabdub and Seinfeld, 1994), the flux-corrected transport, the semi-Lagrangian method, and the chapeau function scheme with Forester filter. These codes are not integrated into the CMAQ system yet, but along with other advection modules, will be added to the system in the near future.

To simplify the discussion, we will consider a uniform (i.e., constant $\Delta x = \Delta x_j$) and staggered grid (ϕ_j represents the grid cell average of the concentration, while $u_{j+1/2}$ is the advection velocity defined at grid cell interfaces). While discussing the finite-volume schemes (the piecewise parabolic method, the Bott scheme and the Yamartino scheme) below, we use the explicit flux formula presented in Equation 7-5. Further, a nondimensional coordinate η is defined as $\eta = (x - x_{j-1/2})/\Delta x$, so that, in grid cell j , $0 \leq \eta \leq 1$. Now, suppose that the concentration has a certain distribution $\phi_j(\eta)$ in each grid cell. Depending on the direction of the velocity, the flux $F_{j+1/2}$ can be expressed as:

$$F_{j+1/2} = \begin{cases} \frac{\Delta x}{\Delta t} \int_{1-\beta_{j+1/2}}^1 \phi_j(\eta) d\eta, & u_{j+1/2} \geq 0 \\ \frac{\Delta x}{\Delta t} \int_0^{\beta_{j+1/2}} \phi_{j+1}(\eta) d\eta, & u_{j+1/2} < 0 \end{cases} \quad (7-7)$$

where $\beta_{j+1/2}$ is the Courant number at the right boundary of grid cell j .

The conditions of high-order accuracy and freedom from spurious oscillations are difficult to be achieved simultaneously. The usual way to satisfy one of these conditions without significant violation of the other is to introduce a correction mechanism. Typically, this mechanism is provided by nonlinear flux-corrections, or by nonlinear filtering. In advection schemes, such adjustments are either applied implicitly through the solution or explicitly as a subsequent step to the linear solution. There is extensive literature on both solution algorithms (linear and nonlinear) and explicit nonlinear mechanisms.

7.1.3.1 Piecewise Parabolic Method (PPM)

In the piecewise parabolic method (Colella and Woodward, 1984) the concentration distribution is assumed to be parabolic in any given grid cell. In terms of the grid cell average concentration ϕ_j and the predicted values of the parabola at the left and right boundaries of the cell $\phi_{L,j}$ and $\phi_{R,j}$, this distribution can be written as:

$$\phi_j(\eta) = \phi_{L,j} + \eta \left[(\phi_{R,j} - \phi_{L,j}) + 6 \left(\phi_j - \frac{\phi_{L,j} + \phi_{R,j}}{2} \right) (1 - \eta) \right] \quad (7-8)$$

Since the initial cell average is known, the construction of the parabola involves the determination of the edge values. First, an approximation to ϕ at $x_{j+1/2}$ is computed subject to the constraint

that its value is within the range of the values at the neighboring cells. For the uniform Δx_j , a first guess for $\phi_{j+1/2}$ is estimated with:

$$c_{j+1/2} = \frac{7}{12}(c_j^n + c_{j+1}^n) - \frac{1}{12}(c_{j+2}^n + c_{j-1}^n) \quad (7-9)$$

In smooth parts of the solution away from extrema, $\phi_{L,j+1} = \phi_{R,j} = \phi_{j+1/2}$ so that the distribution is continuous at $x_{j+1/2}$. In other parts, the cell boundary values are further modified so that ϕ is monotonic on each grid cell. This step introduces discontinuities at cell edges and yields a piecewise continuous global distribution for concentration. There are two cases where the edge values are modified. First, if ϕ_j is a local extremum, then the distribution is assumed to be constant instead of parabolic. The second case is when ϕ_j is between $\phi_{L,j}$ and $\phi_{R,j}$, but sufficiently close to one of the values so that the parabola may take on values outside the range, and lead to overshoots or undershoots. In this case, to make the distribution monotonic, one of the edge values is reset so that the derivative of $\phi(\eta)$ is zero at the opposite edge.

The most distinctive feature of this monotonic scheme is that the nonlinear adjustments are purely geometric. The numerical diffusion introduced by this scheme may be slightly higher than in some other schemes discussed here, but its monotonic property is desirable for photochemical modeling purposes. The scheme has been used in meteorological modeling (e.g., Carpenter et al., 1990) and in air quality models (Odman et al., 1993) including the CCTM. The scheme can be modified so that, in the neighborhood of a discontinuity, it produces a narrower profile. This feature, known as steepening, avoids the smearing of sharp gradients. Though Carpenter et al. (1990) did not recommend steepening for meteorological modeling, this feature may be beneficial in air quality modeling practice, where steep gradients may occur in the vertical direction. However, it should be noted that steepening is of no value and the PPM reverts to a lower order method in the case of sharp spikes or extreme values in a single cell such as emissions from point sources.

7.1.3.2 Bott Scheme (BOT)

The numerical scheme introduced in Bott (1989) is a positive definite scheme with small numerical diffusion. The distribution of the concentration within the cell is represented by a polynomial of order l as:

$$\phi_j(\eta) = \sum_{k=0}^l a_{j,k} \eta^k \quad (7-10)$$

The polynomial can be made area-preserving by requiring:

$$\varphi_{j+i} = \int_i^{i+1} \sum_{k=0}^l a_{j,k} \eta^k d\eta, \quad i = 0, \pm 1, \pm 2, \dots, \pm \frac{l}{2} \quad (7-11)$$

over a stencil of $l+1$ grid cells by varying the value of i . The solution to this linear system yields the coefficients $a_{j,k}$. The coefficients obtained this way for a quadratic ($l=2$) and quartic ($l=4$) together with those of the donor cell (or upwind) scheme, and Tremback's scheme with second-order polynomials (Tremback et al., 1987), are listed in Table 7-1.

Using Equation 7-7, integrating the polynomial of Equation 7-10 between appropriate limits, we arrive at a first estimate of the fluxes. Finally, to make the scheme positive-definite, the total outflux from cell j is limited by requiring that it should be positive and less than what the available mass in the cell would allow:

$$0 \leq F_j^{\text{out}} \leq \frac{\Delta x}{\Delta t} c_j \quad (7-12)$$

The outflux F_j^{out} is a combination of the boundary fluxes and its expression depends on the sign of the velocities. In the CCTM implementation, we used fourth-order polynomials as recommended by Bott (1989) except for the boundary cells. The scheme is receiving increasing attention in current air quality models because of its high accuracy and low computational cost.

Recently, a monotonic version of the scheme was also developed (Bott, 1992) and the time-splitting errors associated with the use of one-dimensional operators in multidimensional applications were reduced (Bott, 1993). Monotonicity is obtained by directly replacing the positive-definite flux limiter of the original approach by new monotone flux limiters as:

$$\begin{aligned} \min(\varphi_{j-1}^n, \varphi_j^n) &\leq \varphi_j^{n+1} \leq \max(\varphi_{j-1}^n, \varphi_j^n), \quad \text{if } u \geq 0 \\ \min(\varphi_{j+1}^n, \varphi_j^n) &\leq \varphi_j^{n+1} \leq \max(\varphi_{j+1}^n, \varphi_j^n), \quad \text{if } u < 0 \end{aligned} \quad (7-13)$$

Although the new flux limited Bott scheme yields monotonic results, there is an inherent mass conservation problem. This problem is directly related to the flux limiting that takes place. Near the leading edge of a sharp wave the use of second or higher order polynomials causes an underestimation of a certain advective flux, $F_{k+1/2}$. When this flux is not corrected it is less than $F_{k+1/2}$, and an undershoot occurs in cell k , as experienced with the original algorithm (Bott, 1989). The motivation for the monotone flux limitation is to avoid such undershoots. However, there are cases when the monotone flux limiter leaves the underestimated flux intact. Instead of increasing the underestimated flux, the limiter reduces the advective flux downwind, $F_{k+1/2}$, in order to avoid an undershoot in cell k . This eventually reduces the net flux out of the domain resulting in an accumulation of mass in the domain.

Table 7-1. Coefficients of the Polynomials Used in Each Scheme

Donor Cell	Tremback-2	Bott-2	Bott-4
φ_j	φ_j	$-\frac{1}{24}(\varphi_{j+1} - 26\varphi_j - \varphi_{j-1})$	$\frac{1}{1920}(9\varphi_{j+2} - 116\varphi_{j+1} + 2134\varphi_j - 116\varphi_{j-1} + 9\varphi_{j-2})$
a_0			
—	$\frac{1}{2}(\varphi_{j+1} - \varphi_{j-1})$	$\frac{1}{2}(\varphi_{j+1} - \varphi_{j-1})$	$\frac{1}{48}(-5\varphi_{j+2} + 34\varphi_{j+1} - 34\varphi_{j-1} + 5\varphi_{j-2})$
a_1			
—	$\frac{1}{2}(\varphi_{j+1} - 2\varphi_j + \varphi_{j-1})$	$\frac{1}{2}(\varphi_{j+1} - 2\varphi_j + \varphi_{j-1})$	$\frac{1}{16}(-\varphi_{j+2} + 12\varphi_{j+1} - 22\varphi_j + 12\varphi_{j-1} - \varphi_{j-2})$
a_2			
—	—	—	$\frac{1}{12}(\varphi_{j+2} - 2\varphi_{j+1} + 2\varphi_{j-1} - 9\varphi_{j-2})$
a_3			
—	—	—	$\frac{1}{24}(\varphi_{j+2} - 4\varphi_{j+1} + 6\varphi_j - 4\varphi_{j-1} + \varphi_{j-2})$
a_4			

7.1.3.3 Yamartino-Blackman Cubic Scheme (YAM)

Yamartino (1993) presents another finite volume scheme where the interpolating polynomial is a cubic spline:

$$\varphi_j(\eta) = a_0 + a_1\eta + a_2\eta^2 + a_3\eta^3 \quad (7-14)$$

where $a_0 = \varphi_j$

$$a_1 = d_j \Delta x$$

$$a_2 = -\frac{1}{4}(\varphi_{j+1} - 2\varphi_j + \varphi_{j-1}) + \frac{3\Delta x}{8}(d_{j+1} - d_{j-1}) \quad (7-15)$$

$$a_3 = (\varphi_{j+1} - \varphi_{j-1}) - \frac{\Delta x}{6}(d_{j+1} + 10d_j + d_{j-1})$$

The spline derivatives, d_j , are obtained from the tridiagonal system:

$$\alpha d_{j-1} + (1 - 2\alpha) d_j + \alpha d_{j+1} = \frac{\varphi_{j+1} - \varphi_{j-1}}{2\Delta x} \quad (7-16)$$

with $\alpha=0.22826$. Note that a value of $\alpha=0$ would correspond to explicit expressions of d_j .

The positivity of $\varphi_j(\eta)$ is ensured by various mechanisms. First, when φ_j is a local minimum, a donor-cell scheme is used instead of the cubic spline. Second, the spline is spectrally limited by the relation:

$$\left| \frac{a_k}{a_0} \right| \leq \frac{\pi^k}{k!}, \quad k = 1, 2, 3. \quad (7-17)$$

Third, a mass conservative flux renormalization is applied, where the fluxes are normalized with the ratio for the upwind cell of the cell concentration (i.e., concentration at $\eta=0$) divided by the average concentration. Finally, a mildly diffusive filter is applied in an attempt to block the depletion of donor cells. Yamartino's scheme is not monotonic and can generate new maxima.

7.1.4 Treatment of Boundary Conditions

Mathematically, the advection equation is a first-order hyperbolic partial differential equation, so it accepts only inflow boundary conditions. Physically, the solution is not affected by the downwind concentrations, so no outflow boundary conditions should be imposed. In practice, however, some outflow boundary conditions have to be imposed. Often, the characteristics of a numerical advection scheme are affected by the boundary condition used.

In the CCTM, a positive-definite zero-flux outflow boundary condition with appropriate flow divergence restriction is used. The zero-flux divergence condition at the boundary (flux gradient out of the boundary cell is set equal to the flux gradient into the cell, i.e.,

$u_1(\varphi_1 - \varphi_0)/\Delta x = u_2(\varphi_2 - \varphi_1)/\Delta x$) gives:

$$\varphi_0 = \max \left\{ 0, \varphi_1 - \frac{u_2}{u_1}(\varphi_2 - \varphi_1) \right\} \quad (7-18)$$

where φ_0 , φ_1 , and φ_2 represent concentrations outside the computational domain, at the boundary cell, and first cell of the inner domain, respectively. u_1 is wind at the outer boundary flux point and u_2 is wind at the inner boundary flux point. Refer to Figure 7-1 for the cell subscript definitions at the outflow boundary. To prevent spurious boundary flux situations that are often associated with the zero-flux divergence boundary condition, the following constraints are applied. When u_1 is sufficiently small (for example smaller than 10^{-3} m/s), or when the wind is divergent at the boundary cell (i.e., $u_1 \cdot u_2 < 0$), a zero concentration gradient at the outflow boundary (i.e., $\varphi_0 = \varphi_1$) is imposed. This boundary scheme can be used with any of the numerical advection algorithms implemented in CCTM.

On the other hand, Odman (1998) introduced a “no condition at outflow boundary” [sic], for Bott’s scheme where the scheme is modified to remove the need for the concentration φ_0 at the downwind cell. A first-order polynomial (derived from φ_1 and φ_2) is used in Cell 1 to compute the flux out of the domain. A second-order polynomial (derived from φ_1 , φ_2 and φ_3) is used in Cell 2 and a fourth-order polynomial is used in Cell 3. Mathematically, this condition is more correct than the others. However, the order of the polynomial is reduced to one at the boundary while the other conditions use a second-order polynomial to compute the flux out of the domain. Because of the lack of generality of this approach, we have not implemented Odman’s boundary scheme in the CCTM. The improved positive-definite zero-flux outflow boundary condition scheme essentially reproduces his results without having to rely on the modified advection algorithms near boundary.

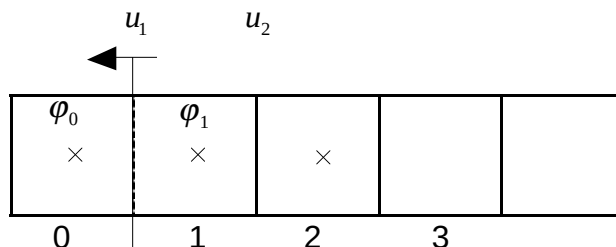


Figure 7-1. Outflow boundary condition. Cell 0 is outside the domain. Vertical solid line denotes the domain boundary and the vertical dotted lines denote the cell interfaces. The advection scheme computes concentrations for Cells 1, 2, 3, and so on.

7.1.5 Test of Algorithms with Idealized Linear Horizontal Flow Fields

Typically, the performances of advection schemes are measured and compared with each other using test cases with idealized flow fields. These ideal flow tests have analytic solutions and may be very useful for determining certain properties of the schemes. Odman (1998) provides some evaluation results identifying schemes with desirable properties. Here we summarize the results of the one-dimensional tests and rotating cone test. The evaluation and comparison of the schemes are based on the performance measures listed in Table 7-2. Again, readers are referred to Odman (1998) for the details of the analysis.

Table 7-2. Summary of Performance Measures Used to Test the Effects of Numerical Advection. ϕ_i^* and ϕ_i^{*e} methods and exact concentrations.⁺

Performance Measure	Formula	Description
peak ratio ^{&}	$\frac{\max(\phi_i^*)}{\max(\phi_i^{*e})}$	Measure of peak preservation (best when 1.0)
Background to peak ratio	$\frac{\min(\phi_i^*)}{\max(\phi_i^{*e})}$	Measure of ripples introduced by a non-monotonic scheme
mass ratio	$\frac{\sum(\phi_i^*)}{\sum(\phi_i^{*e})}$	Measure of mass conservation characteristic (best when 1.0)
Distribution ratio	$\frac{\sum(\phi_i^*)^2}{\sum(\phi_i^{*e})^2}$	Measure of shape retention (best when 1.0)
Average absolute error	$\frac{1}{N} \sum \phi_i^* - \phi_i^{*e} $	Measure of absolute difference (best when 0.0)
root-mean square error	$\sqrt{\frac{1}{N} \sum \left(\frac{\phi_i^* - \phi_i^{*e}}{\phi_i^{*e}} \right)^2}$	Measure of distribution error (best when 0.0)

⁺ To account for the total mass correctly, the concentration should be coupled with the Jacobian of the grid system

[&] Peak ratio alone is not a meaningful criteria unless the positions of the computed and actual peaks are also given.

7.1.5.1 Advection of One-Dimensional Pulses

Various tests have been conducted in the literature with pulses of different shapes advected with uniform velocity. We conducted tests using a Gaussian signal of exactly $8\Delta x$ width with 100 ppm peak advected from cell 25 to cell 75 in a 100-cell uniform grid domain. Background values were set at 5 ppm. Table 7-3 summarizes the value of various performance measures after advected the signal for a distance of $50\Delta x$ at a Courant number of 0.25 (i.e., after 200 time-steps). The Accurate Space Derivative (ASD) and Yamartino's schemes (YAM) preserved the peak height very well. However, the distribution ratio has a lower value for YAM, indicating distortions of the pulse's shape. For the same reason, the average and RMS errors are larger than those of the ASD. On all accounts, these two schemes perform much better than the other schemes in this test. Bott's scheme (BOT) ranks third overall, but large ripples are observed at leading and trailing edges of the pulse as indicated by the values below the background (as much as 4% of the peak height). When the monotonic limiter is used (BOT-M), the ripples are eliminated but the peak retention performance deteriorates. Also, a 2% increase in mass is observed. PPM is somewhat more diffusive than BOT-M. However, PPM performed better than BOT-M in minimizing the average absolute error and root mean square error and because it is intrinsically monotonic, did better in regard to shape retention as measured by the distribution ratio. Smolarkiewicz's scheme (SMO) displays poorer performance than the other schemes.

SMO produces ripples upwind from the pulse and leads to average and RMS errors larger than other schemes.

Table 7-3. Gaussian signal test (Abridged from Odman, 1998)

Scheme	ASD	BOT	BOT-M	PPM	SMO	YAM
Peak Ratio	0.99	0.87	0.74	0.69	0.61	0.98
Background	0.05	0.01	0.05	0.05	0.02	0.05
Mass Ratio	1.00	1.00	1.02	1.00	1.00	1.00
Distribution	0.99	0.93	0.83	0.79	0.66	0.92
Average Error	0.08	0.87	1.38	1.16	2.21	0.51
RMS Error	0.01	0.18	0.27	0.17	0.50	0.12

7.1.5.2 Rotating Cone Test

In this test, a cone-shaped puff is introduced into a rotational flow field and followed for a certain number of revolutions. The exact solution is a rigid-body rotation of the puff without any change to its original shape. Various errors can be revealed in this test. For example, numerical diffusion (or dissipation) manifests itself in the drop of the peak height during rotation. Also, by observing the location of the peak, one can determine the leading or lagging phase-speed errors.

A 32x32 grid is used for this test (i.e., $-16\Delta x \leq x \leq +16\Delta x$; $-16\Delta y \leq y \leq +16\Delta y$ $\Delta x = \Delta y$). A cone-shaped puff with peak concentration equal to 100 ppm and a base radius of $4\Delta x$ is initialized such that its peak is located at $[+8\Delta x, 0]$. Note that the peak is not initially at a grid-cell center but a cell corner (i.e., there are four cells around the peak with the same average concentration). The background concentration is set to 5 ppm. To obtain a counterclockwise rotation around an axis passing through the center of the domain, the wind field is defined as $u = -\omega y$ and $v = \omega x$. The angular velocity, ω , is adjusted so that the Courant number of approximately 0.28 at the location of the peak of the puff.

Table 7-4. Rotating Cone Test (Abridged from Odman, 1998)

Scheme	ASD	BOT	BOT-M	BOT2D	PPM	SMO	YAM
Peak Ratio	0.99	0.87	0.65	0.65	0.61	0.49	0.99
Background	0.06	0.03	0.06	0.06	0.06	0.02	0.06
Mass Ratio	1.00	1.00	1.02	1.02	1.00	1.00	1.00
Distribution	0.96	0.93	0.83	0.83	0.78	0.64	0.91
Average Error	0.18	0.46	0.76	0.76	0.54	1.60	0.33
RMS Error	0.05	0.16	0.30	0.30	0.18	0.51	0.13

ASD and YAM maintain the peak height and the overall shape of the cone better than the others. BOT performs third best in this test, but it yields non-monotonicity, which results in values below the background (as seen in a ring-shaped valley at the base of the cone). BOT-M and

PPM predict similar peak heights (65% and 61%, respectively), but the shape distortions look very different in each case. PPM has the worst peak clipping effect but the resulting shape has the smallest base span among the three schemes. SMO is clearly the most diffusive scheme; it also introduces a ripple upwind from the cone. Table 7-4 summarizes the performance measures at the end of two rotations. Since there is no shear in the flow field, BOT-M and the two-dimensional version of Bott's scheme (BOT2D) (Bott, 1993) produce identical results. Again, the mass conservation problem is revealed with BOT-M. BOT preserves 87% of the peak height (third best after ASD and YAM), but it leads to ripples with an amplitude of 3% of the original peak height. Performances of BOT-M and PPM are comparable in predicting the peak. But PPM produces the lower distribution ratio, smaller absolute average and RMS errors. Notice that the comparison results obtained from this test are very similar to those of the Gaussian signal test. Additional test results such as skew advection of a point-source plume and advection with shear flow are available in Odman (1998). Also, effects of density distribution on the numerical advection are studied with a set of linear flows in Byun and Lee (1999). The solvers integrated into the CCTM are BOT, PPM and YAM. Although ASD has very high accuracy except for the $2 \Delta x$ wavelengths, ASD is neither strictly mass conservative nor monotonic. It is also the most CPU-intensive scheme (taking about 4-5 times longer than BOT). In addition to BOT's overall performance, results reported in Odman (1998) for a broad series of tests showed that BOT had the best computational performance of all the schemes tested. However, because of the concerns over the non-monotonicity about BOT and the mass conservation problem and diffusive nature of BOT-M, we chose PPM for a number of demonstration executions (Byun et al., 1998). Similar testing with BOT and YAM is underway. We intend to integrate other methods into the CCTM at a later time.

7.1.6 Vertical Advection

Algorithms in the CCTM for vertical advection are essentially the same as those for the one-dimensional horizontal algorithms. However, the vertical advection is performed in terms of the generalized vertical coordinate in CMAQ. The contravariant vertical velocity component is used as the transport wind for the irregular vertical grid spacing (usually expanding with altitude) represented in the generalized coordinate. For the irregular grid, the computational time-step should satisfy the CFL condition:

$$\Delta t \leq \min_k \left\{ \frac{\Delta \xi_{k-1}}{|\hat{v}_{k-1/2}^3|}, \frac{\Delta \xi_k}{|\hat{v}_{k-1/2}^3|}, \frac{\Delta \xi_k}{|\hat{v}_{k+1/2}^3|}, \frac{\Delta \xi_{k+1}}{|\hat{v}_{k+1/2}^3|} \right\} \quad (7-19)$$

As in the case of Cartesian representation, we assume there is no mass exchange by advection (i.e., $\overline{\hat{v}^3} = 0$) at the top and bottom boundaries of the model. Therefore, there is no need to apply special algorithms for the boundary process. Because vertical grid spacing is usually irregular in

most air quality models, a scheme that can accommodate irregular spacing must be used for vertical advection. This means that numerical algorithms used to represent the vertical advection may be different from that used for horizontal advection. For example, when the ASD algorithm, which requires equal spacing in the computational domain, is used for horizontal advection, a numerical algorithm that allows irregular spacing (e.g., BOT and PPM) would need to be used for vertical advection.

7.1.7 Adjustment of Mass Conservation Error

Recently, Byun (1999a, b) has highlighted the importance of dynamic consistency in meteorological and air quality modeling for multiscale atmospheric applications. Mass consistency quantifies how well the density and wind fields satisfy the continuity equation for air. One of the fundamental requirements for the numerical transport algorithms used in air quality models is the conservation of trace species in the domain. Ideally, the input meteorological data for air quality simulations should be mass consistent. However, using numerical models with highly parameterized physical and cloud algorithms, inappropriate set of governing equations, misapplication of four-dimensional data assimilation (FDDA) schemes, or using incomplete objective analysis methods to characterize the atmosphere for a CTM could result in meteorological conditions that are not mass consistent. In this situation, even precisely mass conserving numerical algorithms may fail to conserve trace species mass in the domain. Preferably, the mass inconsistency must be minimized before air quality simulation using a suitable diagnostic relation or a variational wind field adjustment scheme as discussed in Byun (1999b) and in Chapter 5. For certain vertical coordinates with appropriate dynamic assumptions, the diagnostic methods can be used within the CTM to provide mass consistent wind data. This is accomplished usually by adjusting the vertical wind component for the advection process. However, the variational methods are applied during the meteorological data preparation stage, instead of inside CTMs, mostly due to the computational efficiency reasons.

Whether the meteorological data are mass consistent or not, Byun (1999b) has shown that the tracer mixing ratio must be conserved as a precondition for the tracer mass conservation. He has reported several adjustment schemes used in current air quality models and proposed a two-step time splitting numerical algorithm that satisfies the mixing ratio conservation equation. In the event the meteorological data are not mass consistent, the mixing-ratio conservation scheme is demonstrated to be useful for photochemical air quality models where chemical production and loss terms are computed using molar mixing ratio. For this purpose, Equation 7-1 should be modified as follows to conserve trace species:

$$\frac{\partial \phi_i^*}{\partial t} = -\frac{\partial(\phi_i^* \bar{u})}{\partial \hat{x}^1} - \frac{\partial(\phi_i^* \bar{v})}{\partial \hat{x}^2} - \frac{\partial(\phi_i^* \bar{v}^3)}{\partial \hat{x}^3} + \phi_i^* \frac{Q_p}{\rho} \quad (7-20)$$

where Q_ρ is the density error term for the meteorological data. This is a necessary condition for the tracer mass conservation. The total tracer mass of the domain is conserved only when the additional condition that total air mass of the domain is conserved, i.e.:

$$\iiint_{\partial\Omega} \frac{J_s}{m^2} Q_\rho dV = 0 \quad (7-21)$$

where $\partial\Omega$ represents the boundary of a computational domain. However, even this condition does not guarantee the cell base conservation of tracer mass except for the case of uniform mixing ratio, because $\iiint_{\partial\Omega} \phi_i^* \frac{Q_\rho}{\rho} dV = \iiint_{\partial\Omega} q_i \frac{J_s}{m^2} Q_\rho dV \neq q_i \iiint_{\partial\Omega} \frac{J_s}{m^2} Q_\rho dV$ in general. Because of this, the correction methods should only be used for improving mass conservation characteristics of the numerical advection algorithms after the mass inconsistency is minimized. Nevertheless, if the objective is to maintain the property of cell-based mixing ratio conservation, the mass conservation error caused by the inconsistency in meteorology data ($\phi_i^* \frac{Q_\rho}{\rho}$) can be corrected with:

$$(\phi_i^*)^{cor} = (\phi_i^*)^T \exp\left[\int \frac{Q_\rho}{\rho} dt\right] \quad (7-22)$$

where superscripts ' T ' and ' cor ' represent values after transport (advection) and after correction, respectively. Table 7-5 summarizes several adjustment methods reported in Byun (1999b). An adequate correction scheme for photochemical Eulerian AQMs should conserve the tracer mixing ratio at least even if wind and density fields are not mass consistent. The correction scheme proposed and labeled *A5* in Table 7-5 is expected to maintain conservation of mixing ratio up to machine precision. Correction schemes *A1* and *A2* ignore the effects of coordinate/grid structures during the advection process. For Cartesian coordinates where the Jacobian is uniform and constant with time, the scheme *A2* is equivalent to *A5*. For a steady-state flow, schemes *A3*, *A4*, and *A5* become identical. Note that all the correction algorithms in Table 7-5 become the same for a steady state flow with uniform density distribution in Cartesian coordinates.

Table 7-5. Trace Gas Mass Correction Schemes

Symbol	Correction Method	Mass correction algorithm
A0	No correction	
A1	Advection of unity	$\varphi_i^{cor} = \frac{\varphi_i^T}{\varphi_r^T} \varphi_r, \varphi_r = 1.0$
A2	Advection of air density	$\varphi_i^{cor} = \frac{\varphi_i^T}{\rho^T} \rho^{int}$
A3	Advection of $\rho J_\xi / m^2$ with anelastic approximation	$(\varphi_i J_\xi / m^2)^{cor} = (\varphi_i J_\xi / m^2)^T \exp \left[\frac{(\rho J_\xi / m^2)_n - (\rho J_\xi / m^2)^T}{(\rho J_\xi / m^2)_n} \right]$
A4	Advection of $\rho J_\xi / m^2$ with finite differencing of tendency term	$(\varphi_i J_\xi / m^2)^{cor} = (\varphi_i J_\xi / m^2)^T \exp \left[\frac{(\rho J_\xi / m^2)^{int} - (\rho J_\xi / m^2)^T}{(\rho J_\xi / m^2)^{int}} \right]$
A5	Advection of $\rho J_\xi / m^2$ with two-step time splitting	$(\varphi_i J_\xi / m^2)^{cor} = \frac{(\varphi_i J_\xi / m^2)^T}{(\rho J_\xi / m^2)^T} (\rho J_\xi / m^2)^{int}$

(Note: Superscripts *cor*, *int*, and *T* represent corrected, interpolated, and advected quantities, respectively.)

Byun and Lee (1999) compared the performance of the correction algorithms under several idealized two-dimensional linear flows and different density fields. In addition, the linearity of the advection processes was studied to show how well the different precursors from same sources can be advected without losing their integrity. The characteristics of the correction procedures can be summarized as follows:

- The correction scheme fixes mass conservation errors due to the time splitting and numerical algorithms as well as the mass inconsistent meteorological data input.
- The correction scheme does not, however, improve inherent properties of the numerical advection scheme such as monotonicity, or numerical diffusion.
- In a budget study, such as in process analysis, the adjustment process must be considered as an integral part of the three-dimensional advection. Horizontal advection, vertical advection, and the mass correction step all together simulate the atmospheric advection process.

7.2 Vertical Mixing Algorithms

In this section, we describe turbulence closure schemes for the CCTM. We describe numerical algorithms for vertical mixing and dry deposition, and discuss the details of the mathematical representations, integration time-steps and solvers. Unless stated otherwise, the vertical structures of meteorological and chemical transport models are assumed to be the same. This is

desirable because the interpolation of meteorological quantities in vertical direction would alter the original turbulent flux exchange characteristics.

7.2.1 Closure Problem

Because of the stochastic nature of atmospheric motion, the primitive equations in the set describing the atmosphere are averaged to form a set of deterministic equations before they can be solved numerically. The decomposition of velocity components and concentrations into mean and turbulent terms and the application of ensemble averaging produces Reynolds flux terms in the species mass continuity equation. Introduction of the Reynolds flux terms generates a new problem set in which the number of unknowns is larger than the number of equations. This closure problem is caused by the attempt to represent nonlinear processes such as momentum advection using a linear decomposition such as the Reynolds decomposition. We describe some approaches on the closure of the Reynolds flux terms below.

7.2.1.1 Local Closure

Local closure assumes that turbulence is analogous to molecular diffusion, i.e., that an unknown turbulence flux at any point in space is parameterized by values of known quantities at the same point (Stull, 1988). First order closure retains the prognostic equations for only the mean variables such as wind, temperature, humidity, and trace gas concentrations while the second-order moments (Reynolds fluxes) are approximated. An example of a local closure scheme is the approximation of Reynolds flux terms using a gradient transport theory, or a mixing length theory resulting in an eddy diffusion method. One of the problems with the gradient transport theory is finding a rational basis for parameterizing the eddy diffusivity. Also, the theory fails when eddies larger than the grid size are present, like they are in a convective boundary layer.

The so-called one-and-a-half order closure retains the prognostic equations for the mean variables and adds equations for the variances of those variables. The set of one-and-a-half order equations is obtained by simplifying the full second-order turbulence equations. Instead of the velocity component variance equations, the turbulent kinetic energy (TKE) equation is often used. By including the variance equations, we have increased the number of unknowns that need to be parameterized compared to the first-order closure approach. However, the benefit is that the eddy diffusivity can be parameterized not only with the mean quantities but also with the TKE and the temperature variance which characterize turbulence intensity. However, if an air quality model is based on one-and-a-half order closure in a true sense, the prognostic equations of the variances for the tracer concentrations should be included explicitly. In practical Eulerian air quality models that deal with photochemical problems, additional prognostic variance equations for the tracer species are very expensive computationally. Also, the additional closure problem

must be dealt with by parameterizing the Reynolds average terms involved with the variances for tracer species. Therefore, the one-and-a-half order closure for an air quality model often actually means that the diffusion equations for the tracer species are formulated with first order closure, while eddy diffusivities (if gradient theory is applied for the closure) or turbulent fluxes (if non-local flux-based closure is used) are estimated with the TKE information from a meteorological model with one-and-a-half order schemes for wind, temperature, and humidity.

The set of second-order turbulence equations includes all the second moment terms. To derive these terms, parameterizations on a full set of third-order moments are required. Similar to the first-order case, the second-order closure approximates terms involving third moments. Several basic closure assumptions such as down-gradient diffusion, return to isotropy, and turbulent dissipation in the inertial subrange are used in the parameterization of the third moment terms. These parameterizations must be valid, especially, for the scales of the energy-containing eddies that are sensitive to atmospheric stability. Measurements of high-order moments in the real atmosphere are difficult because of large scatter in the direct flux measurements and because a long averaging time or a very large sample size of data is required because the events with a much lower probability of occurrence must be gathered to estimate higher-order moments using eddy-correlation methods. For air quality applications, especially for a complex chemical reaction system, the technique requires too many ad-hoc assumptions that cannot be confirmed by observations or other theoretical reasoning. In addition the second-order closure incurs a prohibitively high computational cost. As before, we can solve the first-order tracer diffusion equations with the variances and covariances for wind components, temperature, and humidity from the second-order meteorological models. For air quality applications, the true second-order closure formulation solves for the cross-species covariances explicitly. Some researchers have attempted second-order closure for simple chemical mechanisms with a limited number of photochemically reactive species. The introduction of additional parameterizations for the third-order moment terms among the tracer species themselves and wind components (about which we lack sufficient knowledge) and the added cost of solving for a large number of covariance terms make this scheme impractical and prohibitively costly for operational Eulerian photochemical models.

7.2.1.2 Non-local Closure

Non-local closure recognizes that larger-size eddies can transport fluid across finite distances before the smaller eddies have a chance to cause mixing. This advection-like concept is supported by observations of thermals rising with undiluted cores, finite size swirls of leaves or snow, and the organized circulation patterns sometimes visible from cloud photographs.

Two main approaches of non-local closure methods are the transilient turbulence theory and the spectral diffusion theory. Both allow a range of eddy sizes to contribute to the turbulent mixing process. The spectral diffusion theory attempts to simulate mixing process by transforming signals into a spectral space. For example, a spectral diffusion model of Otte and Wyngaard (1996) represents the mean variables within the planetary boundary layer (PBL) by a truncated series of Legendre polynomials. The first Legendre mode represents the layer average, and additional modes add structure to the vertical profiles. Only a few modes are necessary to resolve vertical profiles comparable to high resolution diffusion models. However, the need to fit a different number of spectral modes for each trace species makes the scheme less attractive for air quality application and thus it have not been considered here. The transilient turbulence theory (e.g., Stull, 1988) is a general representation of the turbulent flux exchange process. The Latin word transilient, meaning to jump over, is used since turbulent eddies that exist in the PBL can transport mass and momentum directly across several grid layers. A variety of mixing processes can be modeled with the transilient scheme depending on the form of the transilient matrix. Examples include complete mixing, top-down/bottom-up mixing, asymmetric convection mixing, small-eddy mixing, cloud top entrainment, a detraining updraft core, patchy turbulence, no turbulence, or eddies triggered by the surface layer. Non-local closure is most suitable for describing vertical turbulence mixing process, which should represent turbulent diffusion and atmospheric transport by eddies of different sizes simultaneously.

In the following, we describe only the first-order turbulent mixing schemes. The true one-and-a-half and the second-order closure schemes are not discussed here. We organize the description of vertical mixing algorithms based on the details of the turbulence parameterization; the eddy-diffusion form or the Reynolds-flux form. Both the eddy-diffusion and the Reynolds-flux forms are capable of accommodating information from higher order turbulence closure for momentum and other meteorological parameters such as potential temperature and humidity. The vertical diffusion modules in CMAQ will include two different ways of parameterizing the eddy diffusivity (using PBL similarity theory and using TKE) and three flux form non-local algorithms (Blackadar, ACM, and Transilient Turbulence).

7.2.2 Computing Vertical Mixing with the Eddy Diffusion Formulation: K-Theory

The eddy diffusion algorithm in the CCTM computes the following:

$$\left. \frac{\partial \phi_i^*}{\partial t} \right|_{\text{vdiff}} = \frac{\partial}{\partial x^3} \left[\sqrt{\hat{\gamma}} \bar{\rho} \left(\hat{K}^{33} \frac{\partial \bar{q}_i}{\partial x^3} \right) \right] \quad (7-23)$$

where $\hat{K}^{33}$ is the contravariant vertical component of eddy diffusivity in the generalized coordinates. The contravariant eddy diffusivity is related to the diffusivities in Cartesian coordinates as:

$$\begin{aligned}\hat{K}^{33} &= \left(\frac{\partial \hat{x}^3}{\partial x}\right)^2 K_{xx} + \left(\frac{\partial \hat{x}^3}{\partial y}\right)^2 K_{yy} + \left(\frac{\partial \hat{x}^3}{\partial z}\right)^2 K_{zz} \\ &= J_\xi^{-2} \left[\left(m \frac{\partial h}{\partial \hat{x}^1} \right)^2 + \left(m \frac{\partial h}{\partial \hat{x}^2} \right)^2 \right] K_H + K_{zz}\end{aligned}\quad (7-24)$$

where $J_\xi^{-1} = \left| \frac{\partial \hat{x}^3}{\partial z} \right|$. Here we focus on the parameterization of the eddy diffusivity, K_{zz} , in a generic Cartesian vertical coordinate z (geometric height h). Parameterizations of the horizontal eddy diffusivity are described later. However, in the current CMAQ implementation, the term involving horizontal diffusivity K_H in Equation 7-24 for the estimation of $\hat{K}^{33}$ is neglected. Evaluation of the effects of this simplification is left for future work.

7.2.2.1 Parameterization of Vertical Eddy Diffusivity K_{zz} with PBL Similarity Theory

There are several eddy diffusivity parameterizations using different similarity theories. Since these are somewhat similar, we consider the formulations suggested by Businger et al. (1971) and Hass et al. (1991) to represent the turbulent process in the surface layer and mixed layer. Previous studies (Chang et al., 1987, and Hass et al., 1991) indicated that this type of formulation can represent turbulent mixing in air quality models adequately. With K -theory, we assume that trace species have non-dimensional profile characteristics similar to potential temperature, Θ , i.e., $K_{zz} = K_h$. We briefly describe the surface and boundary layer similarity theory used for the parameterization of eddy diffusivity for different stability regimes of the PBL below. The stability regime is defined with a nondimensional number z/L , where z is the height above the ground and L is the Monin-Obukhov length.

For the surface layer, the non-dimensional profile functions of the vertical gradient of Θ are expressed as:

$$\phi_h = \text{Pr}_o \left(1 + \beta_h \frac{z}{L} \right) \quad \text{for moderately stable conditions } (1 \geq z/L \geq 0) \quad (7-25a)$$

$$\phi_h = (1 - \gamma_h \frac{z}{L})^{-1/2} \quad \text{for unstable conditions } (z/L < 0) \quad (7-25b)$$

where Pr_o is the Prandtl number for neutral stability and β_h and γ_h are coefficients of the profile functions determined through field experiments. In addition, following Holtslag et al. (1990) we

add a function for the very stable condition ($z/L \geq 1$) to extend the applicability of the surface layer similarity:

$$\phi_h = \text{Pr}_o \left(\beta_h + \frac{z}{L} \right) \quad (7-25c)$$

Parameterizations for eddy diffusivity for the surface layer can be shown as:

$$K_h = \frac{ku_*z}{\phi_h(z/L)} \quad (7-26a)$$

where u_* is the surface friction velocity.

For the PBL (above the surface layer), eddy diffusivity is parameterized with:

$$K_h = \frac{ku_*z(1 - z/h)^{3/2}}{\phi_h(z/L)} \quad \text{for } \frac{z}{L} > 0 \text{ (stable)} \quad (7-26b)$$

$$K_h = kw_*z(1 - z/h) \quad \text{for } \frac{z}{L} < 0 \text{ (unstable)} \quad (7-26c)$$

In the above expressions, h is the depth of the boundary layer, k the Von Karman constant, and w_* the convective velocity. Refer to Chapter 12 for the method used to estimate the PBL height in the CMAQ modeling system.

These parameterizations for K_h are sensitive to the boundary-layer height (h) and surface-layer height. Therefore, when the vertical resolution is too coarse in the boundary layer, using a "representative" eddy diffusivity together with the mean-concentration gradient at the interface seems to be more appropriate for the estimation of the diffusive flux. In fact, the diffusive flux across the interface can be estimated more accurately with the mean diffusivity and mean concentration gradient than with local diffusivity and mean concentration gradient; the former has an error of $O[(\Delta\xi)^2]$ (at best) while the latter has an error of $O[\Delta\xi]$. To estimate the "representative" eddy diffusivity at the layer interface, integrated eddy diffusivity formulas are used as in RADM and CMAQ. They are summarized in the following equations (Byun and Dennis, 1995)

- Surface Layer

(a) Stable conditions:

$$\bar{K}_h = \frac{1}{z_2 - z_1} \int_{z_1}^{z_2} \frac{ku_* z}{\text{Pr}_o (1 + \beta_h z / L)} dz = \frac{ku_*}{\text{Pr}_o (z_2 - z_1)} \left[\frac{z_2 - z_1}{\beta_1} - \frac{1}{\beta_1} \left(\ln \frac{\beta_1 z_2 + 1}{\beta_1 z_1 + 1} \right) \right] \quad (7-27a)$$

where $\beta_1 = \frac{\beta_h}{\text{Pr}_o L}$.

(b) Unstable conditions:

$$\begin{aligned} \bar{K}_h &= \frac{1}{z_2 - z_1} \int_{z_1}^{z_2} \frac{ku_* z}{\text{Pr}_o (1 - \gamma_h z / L)} dz \\ &= \frac{ku_*}{\text{Pr}_o (z_2 - z_1)} \left[\frac{2(3\alpha z_2 - 2)(1 + \alpha z_2)^{\frac{3}{2}}}{15\alpha^2} - \frac{2(3\alpha z_1 - 2)(1 + \alpha z_1)^{\frac{3}{2}}}{15\alpha^2} \right] \end{aligned} \quad (7-27b)$$

where $\alpha = -\gamma_h / L$.

- Planetary Boundary Layer

(a) Stable conditions:

$$\begin{aligned} \bar{K}_h &= \frac{ku_*}{z_2 - z_1} \int_{z_1}^{z_2} \frac{z(1 - z/h)^{3/2}}{\text{Pr}_o (1 + \beta_h z / L)} dz \\ &= \left[\frac{r_2^5}{5} + \frac{a^2 - 1}{3} r_2^3 + (a^4 - a^2) r_2 \right] - \left[\frac{r_1^5}{5} + \frac{a^2 - 1}{3} r_1^3 + (a^4 - a^2) r_1 \right] \\ &= (a^6 - a^4) \left[\frac{1}{2a} \ln \frac{(a + r_2)|a - r_1|}{|a + r_2|(a + r_1)} \right] \end{aligned} \quad (7-27c)$$

where $r_1 = (1 - z_1 / h)^{1/2}$, $r_2 = (1 - z_2 / h)^{1/2}$, $a^2 = (1 + \delta) / \delta$, and $\delta = \frac{\beta_h h}{\text{Pr}_o L}$.

(b) Unstable conditions:

$$\bar{K}_h = \frac{kw_*}{z_2 - z_1} \int_{z_1}^{z_2} \left(z - \frac{z^2}{h} \right) dz = kw_* \left[\frac{z_2 + z_1}{2} - \frac{z_2^2 + z_2 z_1 + z_1^2}{3h} \right] \quad (7-28)$$

In the free atmosphere above the mixed layer, turbulent mixing is parameterized using the formulation as implemented in RADM (Chang et al., 1987), in which vertical eddy diffusivities are represented as functions of the bulk Richardson number and wind shear in the vertical. This formulation can be written as:

$$K_h = K_o + S^2 \frac{Ri_c - Ri_B}{Ri_c} \quad (7-29)$$

where the Richardson number is defined as $Ri_B = \frac{g}{\Theta_o S^2} \frac{\Delta \Theta_k}{\Delta z_k}$, K_o is the background value set at $1 \text{ m}^2 \text{ s}^{-1}$, and S is the vertical wind shear. Effects of the parameterization, Equation 7-29, on air quality simulation will be evaluated in near future. Furthermore, not much is known of the eddy diffusivity formulas for strong stable condition (i.e., $z/L > 1$) and therefore require further research in this area.

7.2.2.2 Estimation of Vertical Eddy Diffusivity Using Turbulent Kinetic Energy

Various forms of higher order closure schemes are becoming common in mesoscale meteorology models. Typically, such models are referred to as TKE models. The simplest form of a TKE model, which Mellor and Yamada (1974) referred to as level 2.5, has only one second order prognostic equation, for TKE itself. The next level up in complication is the true one-and-a-half order closure model, level 3 in Mellor and Yamada (1974) nomenclature, which in addition to TKE includes prognostic equations for the turbulent variances of other relevant quantities such as temperature and humidity. The TKE scheme in the latest version of MM5 can be run as either level 2.5 or level 3 where prognostic equations for temperature variance, moisture variance, and temperature-moisture covariance are added to the TKE equation (Burk and Thompson, 1989). Another similar form of a higher order closure model is known as TKE- ϵ , which has prognostic equations for TKE and the turbulent dissipation rate (ϵ). An example of this type of model is described by Alapaty et al (1996).

A common feature of TKE models, for level 2.5 or 3 or TKE- ϵ , is that the turbulent fluxes of momentum, heat, and moisture are represented as local gradient diffusion similar to Equation 7-23. It is this characteristic that distinguishes 1.5 order closure from true 2nd order closure where the flux covariances are explicitly represented by prognostic equations. Therefore, adaptation of a CTM to a meteorology model that includes TKE closure is essentially simple if the TKE fields are available to the CTM. Only a small part of the TKE model need be reproduced in the CTM, namely the parameterization of eddy diffusion coefficients based on quantities already produced in the meteorology model. For example, in the TKE- ϵ model of Alapaty et al. (1996), the governing TKE equations in Cartesian coordinates are:

$$\frac{\partial E}{\partial t} = -u \frac{\partial E}{\partial x} - v \frac{\partial E}{\partial y} - w \frac{\partial E}{\partial z} - \overline{u''w''} \frac{\partial u}{\partial z} - \overline{v''w''} \frac{\partial v}{\partial z}$$

$$+ \frac{g}{\Theta} \overline{\theta'' w''} - \frac{\partial}{\partial z} \left(\overline{w'' E''} + \frac{1}{\rho} \overline{p'' w''} \right) - \varepsilon \quad (7-30a)$$

$$\begin{aligned} \frac{\partial \varepsilon}{\partial t} = & -u \frac{\partial \varepsilon}{\partial x} - v \frac{\partial \varepsilon}{\partial y} - w \frac{\partial \varepsilon}{\partial z} + \frac{c_1 \varepsilon}{E} \left(-\overline{u'' w''} \frac{\partial u}{\partial z} - \overline{v'' w''} \frac{\partial v}{\partial z} + \frac{g}{\Theta} \overline{\theta'' w''} \right) \\ & - \frac{c_2 \varepsilon^2}{E} - c_3 \frac{\partial}{\partial z} (\overline{w'' \varepsilon''}), \end{aligned} \quad (7-30b)$$

where u, v are horizontal wind components;

E is the turbulent kinetic energy per unit mass;

z is height;

Θ_v is the virtual potential temperature;

ε is the rate of TKE dissipation;

$\overline{u'' w''}$, $\overline{v'' w''}$, and $\frac{g}{\Theta} \overline{\theta'' w''}$ are turbulence fluxes of momentum components and heat flux;

ρ is the density; and

p'' is the fluctuating pressure.

The constants c_1 , c_2 , and c_3 can be estimated following Detering and Etling (1985). The first three terms on the right-hand-side of the first equation represent advection of turbulent kinetic energy, and the other terms, shear production, buoyancy production, turbulence transport and the dissipation, respectively. Similarly, the first three terms on the right-hand-side of the second equation represent advection of ε . The rest of the terms represent the net rate change of ε due to shear and buoyancy production, the rate change of ε related to the time-scale of turbulence, and the vertical transport, respectively. The coefficients of eddy viscosity for momentum and heat can be written as:

$$K_m = \frac{c_2 E^2}{\varepsilon} \quad (7-31a)$$

$$K_h = K_m \frac{\phi_m(z/L)}{\phi_h(z/L)} \quad (7-31b)$$

where K_m and K_h are the eddy exchange coefficients for momentum and heat, respectively. E and ε need to be provided by meteorological models in their native coordinates, thus the parameterization does not depend on the grid spacing explicitly. Obviously, we are assuming here that the vertical layer structure in the CTM is compatible with that used in the preprocessor meteorological model. Once K_h is computed, chemical fluxes can be modeled assuming similarity

with heat flux using the same eddy diffusion numerical algorithm described below (section 7.2.3.1). The derivation of eddy coefficients in the Burk and Thompson (1989) model is algebraically more complex (see Mellor and Yamada, 1974, 1982) but based only on mean quantities plus TKE. Therefore, given the TKE fields, which are output automatically from MM5 when the Burk and Thompson (1989) option is invoked, K_h can be re-diagnosed within the CTM.

7.2.2.3 Numerical Algorithm for Vertical Eddy Diffusion Modules

As described in Chapter 5, the vertical eddy diffusion module must solve the following equation in terms of mixing ratio $\bar{q}$:

$$\left. \frac{\partial \bar{q}_i}{\partial t} \right|_{\text{vdiff}} = \frac{\partial}{\partial \hat{x}^3} \left(\hat{K}^{33} \frac{\partial \bar{q}_i}{\partial \hat{x}^3} \right) + \hat{K}^{33} \frac{\partial \ln(\sqrt{\hat{\gamma}} \bar{\rho})}{\partial \hat{x}^3} \frac{\partial \bar{q}_i}{\partial \hat{x}^3}, \quad (7-32)$$

where the vertical mixing is represented with the pure diffusion term and the turbulent flux exchange term, respectively.

Numerical algorithm for the diffusion kernel

In its generic form, the diffusion kernel solves for:

$$\frac{\partial q}{\partial t} = \frac{\partial}{\partial \xi} \left(K \frac{\partial q}{\partial \xi} \right) \quad (7-33)$$

where ξ is the generic vertical coordinate which increases with geometric height. To account for the loss process due to deposition in the lowest model layer ($j=1$), dry deposition flux is considered as the flux boundary condition at the surface, i.e.:

$$\left. \frac{\partial q_1}{\partial t} \right|_{\text{dep}} = - \frac{v_d}{h_{\text{dep}}} q_1 \quad (7-34)$$

where the geometric thickness of the lowest model layer is used for h_{dep} .

The diffusion equation can be discretized for level j ($1 < j < N$) in a N -layer model and time-steps t^n and t^{n+1} as:

$$q_j^{n+1} = q_j^n + \frac{\Delta t}{\Delta \xi_j} \left[\frac{K_{j+1/2}}{\Delta \xi_{j+1/2}} (\vartheta(q_{j+1}^{n+1} - q_j^{n+1}) + (1-\vartheta)(q_{j+1}^n - q_j^n)) - \frac{K_{j-1/2}}{\Delta \xi_{j-1/2}} (\vartheta(q_j^{n+1} - q_{j-1}^{n+1}) + (1-\vartheta)(q_j^n - q_{j-1}^n)) \right], \quad (7-35a)$$

where ϑ is a time-step weighting factor and

$$\xi_{j+1/2} = (\xi_j + \xi_{j+1})/2; \Delta \xi_{j+1/2} = \xi_{j+1/2} - \xi_{j-1/2}; \Delta \xi_{j+1} = \xi_{j+1} - \xi_j$$

For the lowest model layer, we need to account for the loss due to deposition process:

$$q_1^{n+1} = q_1^n + \frac{\Delta t}{\Delta \xi_1} \left[\frac{K_{1+1/2}}{\Delta \xi_{1+1/2}} (\vartheta(q_2^{n+1} - q_1^{n+1}) + (1-\vartheta)(q_2^n - q_1^n)) \right] - \Delta t \frac{v_d}{h_{dep}} [\vartheta q_1^{n+1} + (1-\vartheta)q_1^n] \quad (7-35b)$$

and for the top layer we have zero flux through the top boundary:

$$q_N^{n+1} = q_N^n - \frac{\Delta t}{\Delta \xi_N} \frac{K_{N-1/2}}{\Delta \xi_{N-1/2}} (\vartheta(q_N^{n+1} - q_{N-1}^{n+1}) + (1-\vartheta)(q_N^n - q_{N-1}^n)) \quad (7-35c)$$

Depending on ϑ , the finite difference scheme is explicit ($\vartheta=0$), semi-implicit ($\vartheta=1/2$), or fully implicit ($\vartheta=1$). In the current version of CMAQ, the semi-implicit (Crank-Nicholson) algorithm is implemented. Equations 7-35a-c can be rearranged to yield a matrix equation:

$$\mathbf{A}\mathbf{q} = \mathbf{b} \quad (7-36)$$

where $\mathbf{A}$ is tridiagonal whose coefficients (sub-diagonal component a_j , diagonal component d_j , and super-diagonal component c_j) when $j=1$ are given as:

$$d_1 = 1 + \frac{\vartheta \Delta t}{\Delta \xi_1} \frac{K_{1+1/2}}{\Delta \xi_{1+1/2}} + \vartheta \Delta t \frac{v_d}{h_{dep}}$$

$$c_1 = -\frac{\vartheta \Delta t}{\Delta \xi_1} \frac{K_{1+1/2}}{\Delta \xi_{1+1/2}} \quad (7-37a)$$

and for $j=N$:

$$a_N = -\frac{\vartheta \Delta t}{\Delta \xi_N} \frac{K_{N-\frac{1}{2}}}{\Delta \xi_{N-\frac{1}{2}}}$$

$$d_N = 1 + \frac{\vartheta \Delta t}{\Delta \xi_N} \frac{K_{N-\frac{1}{2}}}{\Delta \xi_{N-\frac{1}{2}}} \quad (7-37b)$$

and for $2 \leq j \leq N-1$:

$$a_j = -\frac{\vartheta \Delta t}{\Delta \xi_j} \frac{K_{j-\frac{1}{2}}}{\Delta \xi_{j-\frac{1}{2}}}$$

$$d_j = 1 + \frac{\vartheta \Delta t}{\Delta \xi_j} \left(\frac{K_{j+\frac{1}{2}}}{\Delta \xi_{j+\frac{1}{2}}} + \frac{K_{j-\frac{1}{2}}}{\Delta \xi_{j-\frac{1}{2}}} \right)$$

$$c_j = -\frac{\vartheta \Delta t}{\Delta \xi_j} \frac{K_{j+\frac{1}{2}}}{\Delta \xi_{j+\frac{1}{2}}} \quad (7-37c)$$

Coefficients of vector $\mathbf{b}$ are, for $j=1$ and N :

$$b_1 = \left(1 - \frac{(1-\vartheta)\Delta t}{\Delta \xi_1} \left[\frac{K_{1\frac{1}{2}}}{\Delta \xi_{1\frac{1}{2}}} \right] - (1-\vartheta)\Delta t \frac{v_d}{h_{dep}} \right) q_1^n + \left(\frac{(1-\vartheta)\Delta t}{\Delta \xi_1} \left[\frac{K_{1\frac{1}{2}}}{\Delta \xi_{1\frac{1}{2}}} \right] \right) q_2^n$$

$$b_N = \left(1 - \frac{(1-\vartheta)\Delta t}{\Delta \xi_N} \left[\frac{K_{N-\frac{1}{2}}}{\Delta \xi_{N-\frac{1}{2}}} \right] \right) q_N^n + \left(\frac{(1-\vartheta)\Delta t}{\Delta \xi_N} \left[\frac{K_{N-\frac{1}{2}}}{\Delta \xi_{N-\frac{1}{2}}} \right] \right) q_{N-1}^n \quad (7-37d)$$

and for $2 \leq j \leq N-1$:

$$\begin{aligned}
b_j = & \left(1 - \frac{(1-\vartheta)\Delta t}{\Delta \xi_j} \left[\frac{K_{j+\frac{1}{2}}}{\Delta \xi_{j+\frac{1}{2}}} + \frac{K_{j-\frac{1}{2}}}{\Delta \xi_{j-\frac{1}{2}}} \right] \right) q_j^n \\
& + \left(\frac{(1-\vartheta)\Delta t}{\Delta \xi_j} \left[\frac{K_{j+\frac{1}{2}}}{\Delta \xi_{j+\frac{1}{2}}} \right] \right) q_{j+1}^n + \left(\frac{(1-\vartheta)\Delta t}{\Delta \xi_j} \left[\frac{K_{j-\frac{1}{2}}}{\Delta \xi_{j-\frac{1}{2}}} \right] \right) q_{j-1}^n
\end{aligned} \tag{7-37e}$$

The numerical algorithm solving the tridiagonal system is based on the Thomas algorithm. Refer to Appendix 7A.1 for details of the algorithm and stability characteristics.

Numerical algorithm for the coordinate divergence kernel

The coordinate divergence kernel solves for

$$\frac{\partial q}{\partial t} = K \frac{\partial \ln(\sqrt{\hat{\gamma}} \bar{\rho})}{\partial \xi} \frac{\partial q}{\partial \xi} = V_{mix} \frac{\partial q}{\partial \xi} \tag{7-38}$$

where $V_{mix} = K \frac{\partial \ln(\sqrt{\hat{\gamma}} \bar{\rho})}{\partial \xi}$. The differential equation is in advective form with the effective mass transfer velocity V_{mix} . It can be solved with a vertical advection code. Because most operational meteorological models rely on logarithmically spaced vertical layering based on *sigma-p* type coordinates, V_{mix} is expected to be small. Currently, this component is not implemented in CMAQ. However, a quantitative study is needed to assess the importance of this term.

Integration time-steps

According to Oran and Boris (1987), any numerical algorithm for the diffusion equation (with equal grid spacing) should produce the following quantitative properties:

- the total integral of $q(\xi, t)$ should be conserved;
- the amplitude $|q(\xi, t)|$ should decay monotonically;
- there should be no phase errors introduced by the algorithm (for equal grid spacing); and
- positivity should be preserved.

Although the numerical solver algorithm for the semi-implicit scheme is stable in the sense that the amplitude of the signal either decays or stays the same, the positivity condition may not be satisfied if we choose a large time-step for the integration, especially for signals with small wavelengths. For example, with equal grid spacing, we can use the Von Neuman stability analysis technique to demonstrate that some part of the short-wavelength spectrum shows a

negative amplitude when the time-step is too long. The semi-implicit scheme is positive definite given the rather stringent CFL condition, for a uniform vertical grid spacing:

$$\beta \equiv \frac{2\Delta t K}{(\Delta \xi)^2} < 1.0 \quad (7-39a)$$

For non-uniform grid spacing, we use the CFL condition:

$$\max\{\beta_j\} \equiv \max\left\{\frac{\Delta t}{\Delta \xi_j} \left(\frac{K_{j-1/2}}{\Delta \xi_{j-1/2}} + \frac{K_{j+1/2}}{\Delta \xi_{j+1/2}} \right)\right\} < 1.0 \quad (7-39b)$$

to ensure positive definiteness of the semi-implicit solution. The internal time-step for vertical diffusion is thus determined in CMAQ with the following equation:

$$\Delta t_{\text{diff}} = \min\{\Delta t_j\} \quad (7-40)$$

where $\Delta t_j = \Delta \xi_j \left(\frac{K_{j-1/2}}{\Delta \xi_{j-1/2}} + \frac{K_{j+1/2}}{\Delta \xi_{j+1/2}} \right)^{-1}$

7.2.3 Flux Form Representation of Vertical Mixing

Vertical mixing can be represented in flux form as:

$$\frac{\partial q}{\partial t} = -\frac{\partial F^\xi}{\partial \xi} - F^\xi \frac{\partial [\ln(\sqrt{\hat{\gamma}} \bar{\rho})]}{\partial \xi}, \quad (7-41)$$

where $F^\xi = \hat{F}_{q_i}^3$ is the turbulent flux represented in the generalized vertical coordinate ξ , whose value increases monotonically with geometric height. Here, the flux should be parameterized in ξ coordinate instead of the generic height coordinate. The cross directional (w.r.t. generalized coordinate) diffusion terms, as well as the flux divergence due to grid spacing (second term in Equation 7-41), are neglected.

Thus, the numerical solver kernel for the flux form vertical diffusion should solve for:

$$\frac{\partial q}{\partial t} = -\frac{\partial F^\xi}{\partial \xi} \quad (7-42)$$

when the source term is zero. The flux form representation is extremely useful in describing the algorithms based on non-local closure. Non-local closure recognizes that larger-size eddies can

transport fluid across distances longer than the grid increment before the smaller eddies have a chance to cause mixing (Stull, 1988).

To represent the turbulent mass exchange with the transilient parameterization, the boundary layer height must coincide with the height of layer interfaces of the vertical grid. For most situations, the index for the boundary layer top (L_p) is less than the total number of model layers (i.e., $L_p < N$). With the transilient turbulence formulation, the new values of the trace species mixing ratio q due to turbulent mixing for a layer j at a future time ($t + \Delta t$) can be written as:

$$q_j(t + \Delta t) = \sum_{k=1}^{L_p} c_{jk}(t, \Delta t) q_k(t) \quad (7-43)$$

where c_{jk} are the components of a transilient matrix and subscripts j and k are indices of two different grid boxes (vertical layers) below boundary layer top in a column of atmosphere. If we consider turbulent mixing between grid boxes j and k , c_{jk} represents fraction of air mass ending in the grid box j that came from grid box k . The grid box j is considered as the “destination” box while grid box k is considered as the “source” box. Thus, the change in the tracer concentration due to turbulent mixing for grid box j at a time interval Δt is a simple matrix multiplication with concentration from the source cell. The transilient matrix representation is in fact applicable for any physical process that involves mass exchange among grid boxes in a column. For example, convective cloud mixing can be represented by a transilient matrix as well, similar to how mixing in a convective boundary layer is handled.

The mass conservation requirements provide constraints for the coefficients of the transilient matrix. The conservation of air mass requires that the sum over k of all mixing fractions be unity:

$$\sum_{k=1}^{L_p} c_{jk}(t, \Delta t) = 1 \quad (7-44)$$

and the conservation of trace gas amount requires that the sum over j of all mass-ratio weighted transilient coefficients be unity as well:

$$\sum_{k=1}^{L_p} \frac{\Delta \xi_j}{\Delta \xi_k} c_{jk}(t, \Delta t) = 1 \quad (7-45)$$

where $\Delta \xi_j / \Delta \xi_k$ represents the mass ratio (i.e., ratio of layer thicknesses for mixing ratio q) between the source and destination boxes. In order that transilient turbulence theory be useful, the coefficients should be determined using appropriate turbulent flux parameterizations.

Consider how the mixing coefficients are related with the turbulent flux representations. Because the transilient matrix describes the exchange of mass between grid boxes, the kinematic turbulent fluxes across the j -th level can be expressed, for a vertical layering with non-uniform grid spacing (Stull, 1993), for $2 \leq j \leq L_p$ as:

$$F_j(t, \Delta t) = F_{j-1}(t, \Delta t) + \frac{\Delta \xi_j}{\Delta t} \sum_{k=1}^{L_p} c_{jk}(t, \Delta t) [q_j(t) - q_k(t)] \quad (7-46a)$$

and for $j=1$ as:

$$F_1(t, \Delta t) = \frac{\Delta \xi_1}{\Delta t} \sum_{k=1}^{L_p} c_{1k}(t) [q_1(t) - q_k(t)] \quad (7-46b)$$

The concentration (mixing ratio) of the lowest model layer, taking into account the deposition flux, is:

$$\frac{\partial q_1}{\partial t} = -\frac{F_1 - F_s}{\Delta \xi_1} = -\frac{v_d}{h_{dep}} q_1 - \sum_{k=1}^{L_p} \frac{c_{1k}(t, \Delta t)}{\Delta t} [q_1(t) - q_k(t)] \quad (7-47)$$

where F_s is the flux at the surface.

The transilient turbulent representation, as shown in Equation 7-43, is an explicit expression in which the magnitudes of mixing coefficients depend on the numerical integration time-step. To use the transilient matrix representation as a general expression for the non-local turbulent closure methods, we need to relate Equation 7-43 with the semi-implicit representations of simple non-local closure algorithms such as in Blackadar (1978) and the Asymmetric Convection Model (Pleim and Chang, 1992). Using Blackadar's definition of m_{jk} , which is the fraction of mass exchange between two levels j and k per unit time, the turbulent fluxes at a given time t can be rewritten as:

$$F_j(t) = F_{j-1}(t) + \Delta \xi_j \sum_{k=1}^{L_p} m_{jk}(t) [q_j(t) - q_k(t)] \quad (7-48)$$

Note that $m_{jk} = c_{jk} / \Delta t$ for an explicit method. Substituting Equation 7-43 into the semi-implicit method, as in Equation 7-46, we get:

$$q_j^{n+1} = q_j^n - \frac{\Delta t}{\Delta \xi_j} \left[\vartheta (F_j^{n+1} - F_{j-1}^{n+1}) + (1 - \vartheta) (F_j^n - F_{j-1}^n) \right] \quad (7-49)$$

resulting in a general matrix equation:

$$\mathbf{P}\mathbf{q}^{n+1} = \mathbf{R}\mathbf{q}^n \quad (7-50)$$

where the coefficients of $\mathbf{P}$ and $\mathbf{R}$ are given as:

$$p_{jj} = 1 + \vartheta \Delta t \sum_{\substack{k=1 \\ k \neq j}}^{L_p} m_{jk} ; \quad p_{jk} = -\vartheta \Delta t m_{jk} \\ r_{jj} = 1 - (1 - \vartheta) \Delta t \sum_{\substack{k=1 \\ k \neq j}}^{L_p} m_{jk} ; \quad r_{jk} = (1 - \vartheta) \Delta t m_{jk}$$

If the matrix $\mathbf{P}$ is nonsingular, we have a general expression for the transilient turbulence:

$$\mathbf{q}^{n+1} = \mathbf{P}^{-1} \mathbf{R} \mathbf{q}^n = \mathbf{C} \mathbf{q}^n \quad (7-51)$$

where $\mathbf{C} = \mathbf{P}^{-1} \mathbf{R}$.

The relationship between the coefficients of the transilient matrix and the mass exchange ratio among the grid boxes are somewhat complicated for the semi-implicit scheme. However, the semi-implicit scheme becomes attractive for the closure algorithms with sparse $\mathbf{P}$ matrix (i.e., when the matrix inversion is not so expensive) because it allows longer integration time-steps than the explicit method. In the following, we describe non-local flux-form atmospheric turbulence algorithms as a subset of the generalized transilient turbulence representations.

7.2.3.1 Blackadar Non-local Scheme

This scheme, first introduced by Blackadar (1978), has long been used as one of the PBL schemes in the Mesoscale Meteorology model generation 4 (MM4) and generation 5 (MM5). The Blackadar model is a simple non-local closure scheme designed to simulate vertical transport by large convective eddies during conditions of free convection. Therefore, this scheme is used only in the convective boundary layer and must be coupled with another scheme for non-convective conditions and above the boundary layer, such as K-theory. In general, the flux-form diffusion algorithm can be written for the lowest layer as:

$$\frac{\partial q_1}{\partial t} = -\frac{v_d}{h_{dep}} q_1 - \sum_{k=1}^{L_p} m_{1k}(t) [q_1(t) - q_k(t)] \quad (7-52a)$$

and for $2 \leq j \leq L_p$ as:

$$\frac{\partial q_j}{\partial t} = - \sum_{k=1}^{L_p} m_{jk}(t)[q_j(t) - q_k(t)] \quad (7-52b)$$

where m_{jk} is the rate of mass exchange between two grid boxes in a column of the atmosphere (below boundary layer top) per unit time. The convective mixing is assumed to be dominated by eddies of varying sizes but all having roots in the surface layer, each eddy exchanging a certain amount of its mass with the air around it as it ascends. The rate of change of mean potential temperature caused by the mass exchange in the mixed layer can be expressed as:

$$\frac{\partial \Theta}{\partial t} = M_u w(\xi)(\Theta_{sfc} - \Theta) \quad (7-53)$$

where $w(\xi)$ is a weight function that accounts for the variation of exchange rate with height.

The mass exchange rate, M_u , can be estimated from conservation of energy, which requires the heat flux at any level to satisfy the equation:

$$H = H_{sfc} - M_u \int_{\xi_{sfc}}^{\xi} C_{pd} \rho(\Theta_{sfc} - \Theta) w(\xi) J_{\xi} d\xi \quad (7-54)$$

where H_{sfc} is the sensible heat flux leaving the surface layer and C_{pd} is the specific heat at constant pressure. When the integration limit is extended to the top of the boundary layer, where H is assumed to be zero, we can estimate M_u with:

$$M_u = H_{sfc} / \int_{\xi_{sfc}}^{\xi_h} C_{pd} \rho(\Theta_{sfc} - \Theta) w(\xi) J_{\xi} d\xi \quad (7-55)$$

Usually the weight function w is approximated to be unity in the mixed layer. With the Blackadar scheme, the mixing algorithm is represented for the lowest model layer by:

$$\frac{\partial q_1}{\partial t} = - \frac{v_d}{h_{dep}} q_1 - \sum_{k=1}^{L_p} M_u \frac{\Delta \xi_k}{\Delta \xi_1} [q_1(t) - q_k(t)] \quad (7-56a)$$

and for $2 \leq j \leq L_p$ by:

$$\frac{\partial q_j}{\partial t} = - M_u \frac{\Delta \xi_1}{\Delta \xi_j} [q_j(t) - q_1(t)] \quad (7-56b)$$

where we used $m_{1k} = M_u \frac{\Delta \xi_k}{\Delta \xi_1}$, $m_{j1} = M_u \frac{\Delta \xi_1}{\Delta \xi_j}$, and all other components of m_{jk} are zero.

Finite difference representations of the above equations are:

$$\begin{aligned} \frac{q_1^{n+1} - q_1^n}{\Delta t} = & -\frac{v_d}{h_{dep}} [\vartheta q_1^{n+1} + (1 - \vartheta) q_1^n] - \vartheta \frac{M_u}{\Delta \xi_1} \left[(\xi_h - \Delta \xi_1) q_1^{n+1} - \sum_{k=2}^{L_p} \Delta \xi_k q_k^{n+1} \right] \\ & - (1 - \vartheta) \frac{M_u}{\Delta \xi_1} \left[(\xi_h - \Delta \xi_1) q_1^n - \sum_{k=2}^{L_p} \Delta \xi_k q_k^n \right] \end{aligned} \quad (7-57a)$$

where we used $\xi_h = \sum_{k=1}^{L_p} \Delta \xi_k$, and for $2 \leq j \leq L_p$:

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} = -\vartheta M_u \frac{\Delta \xi_1}{\Delta \xi_j} (q_j^{n+1} - q_1^{n+1}) - (1 - \vartheta) M_u \frac{\Delta \xi_1}{\Delta \xi_j} (q_j^n - q_1^n) \quad (7-57b)$$

After rearrangement, we obtain the following matrix equation:

$$\begin{pmatrix} d_1 & f_2 & \cdots & f_k & \cdots & f_{L_p} \\ e_2 & d_2 & 0 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \cdots & \cdots & 0 \\ e_j & \vdots & \vdots & d_j & \vdots & \vdots \\ \vdots & 0 & 0 & 0 & \ddots & 0 \\ e_{L_p} & \cdots & 0 & 0 & 0 & d_{L_p} \end{pmatrix} \begin{pmatrix} q_1^{n+1} \\ q_2^{n+1} \\ \vdots \\ q_j^{n+1} \\ \vdots \\ q_{L_p}^{n+1} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_j \\ \vdots \\ b_{L_p} \end{pmatrix} \quad (7-58)$$

where the elements are defined with:

$$d_1 = 1 + \vartheta \frac{v_d \Delta t}{h_{dep}} + \vartheta \frac{\Delta t M_u}{\Delta \xi_1} (\xi_h - \Delta \xi_1)$$

$$b_1 = 1 - (1 - \vartheta) \frac{v_d \Delta t}{h_{dep}} - (1 - \vartheta) \frac{\Delta t M_u}{\Delta \xi_1} (\xi_h - \Delta \xi_1) q_1^n + (1 - \vartheta) \frac{\Delta t M_u}{\Delta \xi_1} \sum_{k=2}^{L_p} \Delta \xi_k q_k^n$$

and for $2 \leq j \leq L_p$:

$$f_j = -\vartheta \left(\frac{\Delta t M_u}{\Delta \xi_j} \right) \Delta \xi_1$$

$$e_j = -\vartheta \left(\frac{\Delta t M_u}{\Delta \xi_j} \right) \Delta \xi_j$$

$$d_j = 1 + \vartheta \left(\frac{\Delta t M_u}{\Delta \xi_j} \right) \Delta \xi_j$$

$$b_j = \left[1 - (1 - \vartheta) \left(\frac{\Delta t M_u}{\Delta \xi_j} \right) \Delta \xi_j \right] q_j^n + (1 - \vartheta) \left(\frac{\Delta t M_u}{\Delta \xi_j} \right) \Delta \xi_j q_1^n$$

The numerical algorithm to solve the sparse matrix system is similar to the Thomas algorithm for the tridiagonal system. For the details of the numerical algorithm, refer to Appendix 7A.2.

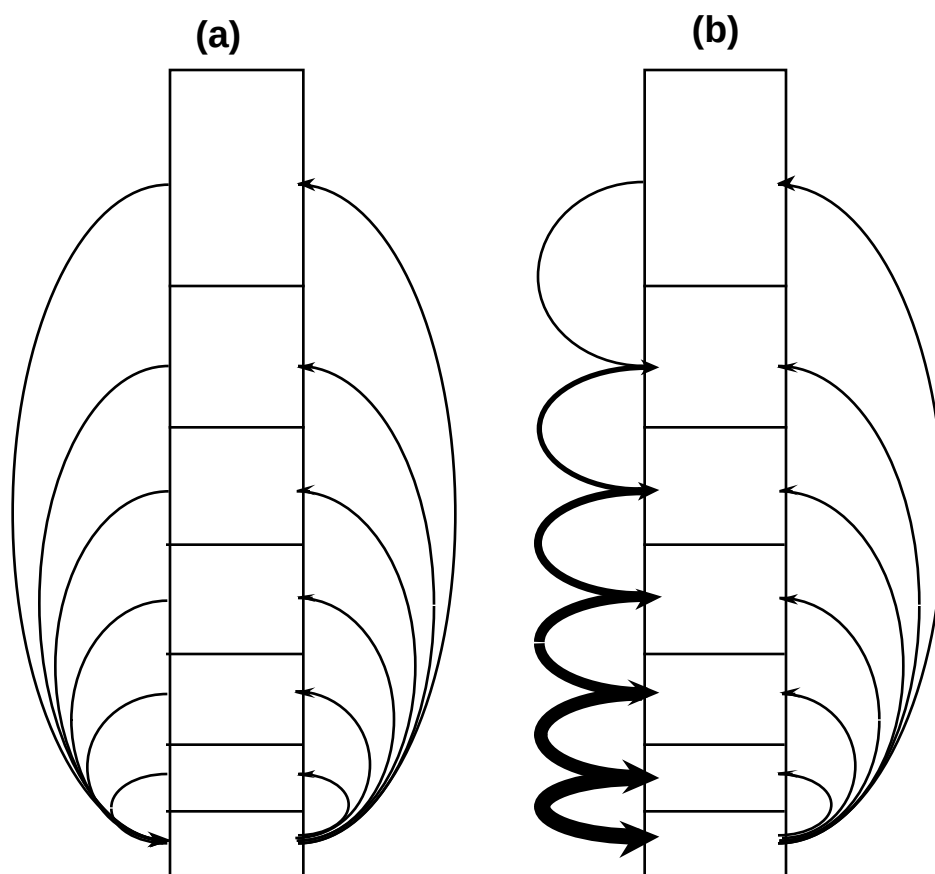


Figure 7-2. Schematics of the Blackadar Scheme (a) and the Asymmetric Convective Model (b)

7.2.3.2 Asymmetric Convective Mixing

The Asymmetric Convective Model (ACM), developed by Pleim and Chang (1992), is based on Blackadar's non-local closure scheme (Blackadar, 1978) but with a different scheme for

downward mixing in the convective boundary layer (CBL). Blackadar's scheme is based on the assumption that the turbulent mixing is isotropic (i.e., symmetric) in the CBL. However, observational evidence and large-eddy simulation modeling studies indicate that mixing processes in a convective boundary layer are essentially asymmetric (i.e., turbulence is anisotropic; Schumann, 1989) with fast upward buoyant plumes and slow broad compensatory subsidence. Therefore the direct, non-local downward transport of the Blackadar scheme is replaced with layer by layer subsidence which increases in mass flux as it descends, like a cascading waterfall (Figure 7-2). As with the Blackadar model, the ACM can only be used during convective conditions in the PBL. For other stability regimes, one needs to rely on other schemes such as K-theory.

Turbulent mixing in the PBL for any dynamic, or thermodynamic variables, or trace gas species concentrations can be represented in essentially the same way as in the above transilient parameterization. Also, the conceptual design of the ACM allows for considerable simplification. Because the mass influx to the lowest model layer is from the second layer only in ACM (refer to Figure 7-2), we can write the time rate change of mixing ratio as follows:

$$\frac{\partial q_1}{\partial t} = -\frac{v_d}{h_{dep}} q_1 - \frac{1}{\Delta \xi_1} [M_u (\xi_h - \xi_1) q_1(t) - M_{d2} \Delta \xi_2 q_2(t)] \quad (7-59a)$$

For $2 \leq j \leq L_p$:

$$\frac{\partial q_j}{\partial t} = M_u q_1 - M_{dj} q_j + M_{dj+1} \frac{\Delta \xi_{j+1}}{\Delta \xi_j} q_{j+1} \quad (7-59b)$$

where M_u represents upward mixing rate. M_{dj} represents downward mixing rate at layer j and is defined as:

$$M_{dj} = M_u \frac{\xi_h - \xi_{j-1}}{\Delta \xi_j} \quad (7-60)$$

The scheme can be represented in terms of transilient mixing rates (m_{jk}), as shown in general form in Equation 7-48:

$$\frac{\partial q_j}{\partial t} = -m_{j1} (q_j - q_1) - m_{j,j+1} (q_j - q_{j+1}) \quad (7-61)$$

Therefore, by equating Equations 7-59b and 7-61, we see that $m_{j1} = M_u \neq m_{1j}$, which shows that the transfer coefficients are asymmetric as expected, and $m_{j1} + m_{j,j+1} = M_{dj}$, which demonstrates a recursive relationship for M_{dj} :

$$M_{dj-1} = M_u + M_{dj} \frac{\Delta \xi_{j+1}}{\Delta \xi_j} \quad (7-62)$$

Finally, we rewrite the prediction equation in terms of the upward mixing rate:

$$\frac{\partial q_1}{\partial t} = -M_u \frac{\xi_h - \xi_1}{\Delta \xi_1} (q_1 - q_2) - \frac{v_d}{h_{dep}} q_1 \quad (7-63a)$$

and for $2 \leq j \leq L_p$:

$$\frac{\partial q_j}{\partial t} = M_u \left[q_1 - \left(\frac{\xi_h - \xi_{j-1}}{\Delta \xi_j} \right) q_j + \left(\frac{\xi_h - \xi_j}{\Delta \xi_{j+1}} \right) q_{j+1} \right] \quad (7-63b)$$

Note that this scheme does include the effects of vertical wind shear in generating turbulent mixing. The magnitude of the mixing rates of the transilient matrix is based on the conservation of sensible heat flux in the vertical direction:

$$M_u = H_{sfc} / [c_p \rho (\bar{\theta}_{sfc} - \theta_2) J_\xi (\xi_1 - \xi_h)] \quad (7-64)$$

Finite difference representations of the above equations are:

$$\begin{aligned} \frac{q_1^{n+1} - q_1^n}{\Delta t} = & - \left(\frac{v_d}{h_{dep}} + M_u \frac{\xi_h - \xi_1}{\Delta \xi_1} \right) [\vartheta q_1^{n+1} + (1 - \vartheta) q_1^n] \\ & + M_u \frac{\xi_h - \xi_1}{\Delta \xi_1} [\vartheta q_2^{n+1} + (1 - \vartheta) q_2^n] \end{aligned} \quad (7-65a)$$

and for $2 \leq j \leq L_p$:

$$\begin{aligned} \frac{q_j^{n+1} - q_j^n}{\Delta t} = & M_u [\vartheta q_1^{n+1} + (1 - \vartheta) q_1^n] \\ & - \left(\frac{\xi_h - \xi_{j-1}}{\Delta \xi_j} \right) [\vartheta q_j^{n+1} + (1 - \vartheta) q_j^n] + \left(\frac{\xi_h - \xi_j}{\Delta \xi_{j+1}} \right) [\vartheta q_{j+1}^{n+1} + (1 - \vartheta) q_{j+1}^n] \end{aligned} \quad (7-65b)$$

This results in a sparse matrix of the form:

$$\begin{pmatrix} d_1 & c_1 & \cdots & 0 & \cdots & 0 \\ e_2 & d_2 & c_2 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \cdots & 0 \\ e_j & \vdots & \vdots & d_j & c_j & \vdots \\ \vdots & 0 & 0 & 0 & \ddots & c_{L_p-1} \\ e_{L_p} & \cdots & 0 & 0 & 0 & d_{L_p} \end{pmatrix} \begin{pmatrix} q_1^{n+1} \\ q_2^{n+1} \\ \vdots \\ q_j^{n+1} \\ \vdots \\ q_{L_p}^{n+1} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_j \\ \vdots \\ b_{L_p} \end{pmatrix}, \quad (7-66)$$

where:

$$\begin{aligned} d_1 &= 1 + \left(\frac{v_d}{h_{dep}} + M_u \frac{\xi_h - \xi_1}{\Delta \xi_1} \right) \vartheta \Delta t; \\ c_1 &= -M_u \frac{\xi_h - \xi_1}{\Delta \xi_1} \vartheta \Delta t; \\ b_1 &= [1 - \left(\frac{v_d}{h_{dep}} + M_u \frac{\xi_h - \xi_1}{\Delta \xi_1} \right) (1 - \vartheta) \Delta t] q_1^n + M_u \frac{\xi_h - \xi_1}{\Delta \xi_1} (1 - \vartheta) \Delta t q_2^n; \\ e_j &= -M_u \vartheta \Delta t; \\ d_j &= 1 + M_u \left(\frac{\xi_h - \xi_j}{\Delta \xi_{j+1}} \right) \vartheta \Delta t; \\ c_j &= -M_u \left(\frac{\xi_h - \xi_j}{\Delta \xi_j} \right) \vartheta \Delta t; \text{ and} \\ b_j &= [1 - M_u \frac{\xi_h - \xi_{j-1}}{\Delta \xi_j} (1 - \vartheta) \Delta t] q_j^n + M_u \frac{\xi_h - \xi_j}{\Delta \xi_{j+1}} (1 - \vartheta) \Delta t q_{j+1}^n + M_u (1 - \vartheta) \Delta t q_1^n. \end{aligned}$$

The numerical algorithm solving the sparse matrix system is presented in Appendix 7A.3.

7.2.3.3 Transient Turbulence Parameterization

The general computational paradigm for the transient turbulence parameterization has been presented above. In order to use the transient turbulence concept for mixing trace gases, one needs to know the mass exchange coefficient matrix. This is the closure problem with this parameterization. A couple of methods have been presented in the literature. One method is based on the TKE equation (Stull and Driedonks, 1987, and Raymond and Stull, 1990) and the other is the one based on non-local Richardson number (Zhang and Stull, 1992). In the following we describe briefly the TKE based scheme and discuss its associated difficulties.

The horizontally homogeneous form of the TKE equation, Equation 7-30a is given as:

$$\frac{\partial E}{\partial t} = -\overline{u''w''} \frac{\partial u}{\partial z} - \overline{v''w''} \frac{\partial v}{\partial z} + \frac{g}{\Theta_o} \overline{\theta''w''} - \varepsilon. \quad (7-67)$$

Note that the pressure and turbulent transport terms have been ignored. After normalizing with E , the non-local analogy of the finite difference form of Equation 7-67 can be written as:

$$\frac{\Delta_t E_{jk}}{E_{jk}} = \left[\frac{(-\overline{u''w''})_{jk}}{E_{jk}} \left(\frac{\Delta u}{\Delta z} \right)_{jk} + \frac{(-\overline{v''w''})_{jk}}{E_{jk}} \left(\frac{\Delta v}{\Delta z} \right)_{jk} + \frac{g}{\Theta_j} \frac{(\overline{\theta''w''})_{jk}}{E_{jk}} - \frac{\varepsilon_{jk}}{E_{jk}} \right] \Delta_t t, \quad (7-68)$$

where the symbol Δ_t represents temporal change while Δ represents spatial gradient. To close the system, the unknown parameters are written in terms of known parameters by introducing three scaling parameters T_o , Ri_c , and D , which are the time scale of turbulence, the critical Richardson number, and dissipation factor, respectively. Thus, weighted kinematic fluxes can be written as:

$$\frac{(-\overline{u''w''})_{jk}}{E_{jk}} = T_o \left(\frac{\Delta u}{\Delta z} \right)_{jk}, \quad (7-69a)$$

$$\frac{(-\overline{v''w''})_{jk}}{E_{jk}} = T_o \left(\frac{\Delta v}{\Delta z} \right)_{jk}, \quad (7-69b)$$

$$\frac{(-\overline{\theta''w''})_{jk}}{E_{jk}} = \frac{T_o}{Ri_c} \left(\frac{\Delta \Theta}{\Delta z} \right)_{jk}. \quad (7-69c)$$

Then Equation 7-68 can be rewritten as:

$$Y_{jk} = \frac{\Delta_t E_{jk}}{E_{jk}} = \frac{T_o \Delta_t t}{(\Delta z)_{jk}^2} \left[(\Delta u)_{jk}^2 + (\Delta v)_{jk}^2 - \frac{g}{Ri_c \Theta_j} (\Delta \Theta)_{jk} (\Delta z)_{jk} \right] - \frac{\varepsilon_{jk} \Delta_t t}{E_{jk}}. \quad (7-70)$$

To use Equation 7-70 for the generalized coordinate ξ , the corresponding layer heights $(\Delta z)_{jk}$ should be computed for $(\Delta \xi)_{jk}$.

Since we are dealing with fraction of masses that are coming from and going to different layers (i.e., $j \neq k$), Y_{jk} in the above equation is for off-diagonal elements only. Diagonal elements in Equation 7-70, Y_{jj} , represent mass of air that remain in each layer without interaction with other layers. Observations during convective conditions (Ebert et al., 1989) indicated that turbulent eddies cause a well-mixed rather than convectively overturned boundary layer. This requires

that the values of the mixing potential elements (Y_{jk}) should increase monotonically from the upper right-most element towards the main diagonal. Further, to account for the subgrid scale mixing in each layer, an independent parameter Y_{ref} was introduced. Thus, the diagonal elements can be written as:

$$Y_{jj} = \text{MAX}(Y_{j,j-1}, Y_{j,j+1}) + Y_{ref} \quad (7-71)$$

given values for T_o , Ri_c , and D . Usually, Y_{ref} is estimated based on observations (Stull, 1988). Finally, the off-diagonal elements of the transilient matrix are estimated with:

$$c_{jk} = \frac{Y_{jk}}{\|\mathbf{Y}\|_{\infty}} \quad (7-72)$$

where $\|\mathbf{Y}\|_{\infty}$ is the infinite norm of matrix $\mathbf{Y}$, $\max\{\|\mathbf{Y}\|\}$. The formulation presented in Raymond and Stull (1990) and Alapaty et al. (1997) includes the additional weighting based on the mass in the layer for irregularly spaced grids. However, we believe that Equation 7-72 should be valid even for irregularly spaced grids when the constraint Equation 7-45 is satisfied. Also, Stull (1993) states that the formulation Equation 7-72 causes too much mixing near the surface and inclusion of the mass weighting in the formulation exacerbates the problem further. The diagonal elements of the transilient matrix can be computed by rewriting Equation 7-44 as:

$$c_{jj} = 1 - \sum_{\substack{k=1 \\ k \neq j}}^{L_p} c_{jk} \quad (7-73)$$

Once the transilient matrix is determined, the concentration due to turbulent processes in the boundary layer can be estimated from Equation 7-43. The difficulties associated with this parameterization are:

- The scheme still depends on many free parameters (T_o , Ri_c , D , and Y_{ref}) and they control the behavior of the mixing algorithm; and
- The time-steps should be such that the trace species mixing ratio cannot be negative because Equation 7-43 is written in explicit form. Although explicit methods do not require matrix inversion, the time-step must be small enough to ensure positivity and numerical stability of the solution.

7.3 Horizontal Mixing Algorithms

Unfortunately, our understanding of horizontal turbulence is limited due to the lack of adequate turbulence measurements as well as the scale dependency of the problem. In earlier days of atmospheric modeling, the horizontal diffusion process was often ignored because the numerical diffusion associated with the advection algorithms used was large. For problems with large scales, such as regional to global transport studies, the coarse grid resolutions did not require an explicit implementation of horizontal diffusion. However, with the advent of very accurate (i.e., less diffusive) numerical advection schemes and the emerging need for high resolution grids for urban scale problems, a good algorithm for horizontal diffusion is required. The skill needed is balancing numerical diffusion associated with the advection schemes with the added explicit diffusion to model horizontal diffusion in the atmosphere. A fundamental problem is that we do not know much about the expected magnitude of the actual horizontal diffusion. In this section we will describe the numerical algorithm for the horizontal diffusion implemented in the CMAQ system.

The horizontal diffusion process in the curvilinear coordinate system (See Equation 6-25' in Chapter 6) is given as:

$$\left. \frac{\partial c_i^*}{\partial t} \right|_{hdiff} = \left. \frac{\partial (\sqrt{\hat{\gamma}} \bar{\rho} \bar{q}_i)}{\partial t} \right|_{hdiff} = -\hat{\nabla}_s \bullet \left[\sqrt{\hat{\gamma}} \bar{\rho} \hat{\mathbf{F}}_{q_i} \right] \quad (7-74)$$

There are not many choices for the horizontal diffusion parameterizations. Frequently, the horizontal turbulent fluxes are parameterized using eddy diffusion theory. The contributions of the off-diagonal diffusion terms show up explicitly as in Equation 6-14 in Chapter 6. Often, these off-diagonal terms are neglected in air quality simulations, and in the CMAQ implementation, we solve for diagonal terms only:

$$\left. \frac{\partial c_i^*}{\partial t} \right|_{hdiff} = \frac{\partial}{\partial \hat{x}^1} \left[\sqrt{\hat{\gamma}} \bar{\rho} (\hat{K}^{11} \frac{\partial \bar{q}_i}{\partial \hat{x}^1}) \right] + \frac{\partial}{\partial \hat{x}^2} \left[\sqrt{\hat{\gamma}} \bar{\rho} (\hat{K}^{22} \frac{\partial \bar{q}_i}{\partial \hat{x}^2}) \right]. \quad (7-75)$$

The contravariant eddy diffusivity components are related to the Cartesian counterparts as $\hat{K}^{11} = mK_{xx}$ and $\hat{K}^{22} = mK_{yy}$. In practice, for Eulerian air quality modeling, we do not distinguish between eddy diffusivities in two different horizontal directions (i.e., $K_{xx} = K_{yy} = K_H$). For a Lagrangian simulation of atmospheric turbulence, the longitudinal (following the plume movement) and lateral (perpendicular to the plume movement) dispersion are treated differently according to characteristics from the isotropic turbulence analysis. Often the horizontal eddy diffusivity in the Cartesian coordinates is parameterized with the magnitude of the deformation in

the gridded wind field. For that case, one must be careful whether the wind data are represented in Cartesian coordinates or in the transformed coordinates.

Unlike the vertical diffusion case, we do not separate the grid divergence term from the diffusion equation. An explicit solution method for Equation 7-75 is:

$$(\rho^*)_{l,m}^{n+1} q_{l,m}^{n+1} = (\rho^*)_{l,m}^n q_{l,m}^n + \frac{\Delta t}{(\Delta \hat{x}^1)^2} \left[(\rho^*)_{l+1,m}^n \overline{\hat{K}_{l+1,m}^{11}} (q_{l+1,m}^n - q_{l,m}^n) - (\rho^*)_{l,m}^n \overline{\hat{K}_{l,m}^{11}} (q_{l,m}^n - q_{l-1,m}^n) \right] \\ + \frac{\Delta t}{(\Delta \hat{x}^2)^2} \left[(\rho^*)_{l,m+1}^n \overline{\hat{K}_{l,m+1}^{22}} (q_{l,m+1}^n - q_{l,m}^n) - (\rho^*)_{l,m}^n \overline{\hat{K}_{l,m}^{22}} (q_{l,m}^n - q_{l,m-1}^n) \right], \quad (7-76)$$

where $\overline{\hat{K}_{l,m}^{11}} = \frac{1}{2}(\hat{K}_{l,m+1}^{11} + \hat{K}_{l,m}^{11})$ and $\overline{\hat{K}_{l,m}^{22}} = \frac{1}{2}(\hat{K}_{l+1,m}^{22} + \hat{K}_{l,m}^{22})$. At the boundary cells, a zero-gradient boundary condition (Neumann) is applied. Because Equation 7-76 is an explicit scheme, the time-step should be chosen to prevent numerical instability and to maintain positivity. With an appropriate Courant number for horizontal diffusion, β_{hdiff} , the time-step can be determined with:

$$\Delta t|_{hdiff} = \beta_{hdiff} \frac{(\Delta x)^2}{\max_{\forall (l,m)} (\hat{K}_{l,m}^{11}, \hat{K}_{l,m}^{22})}. \quad (7-77)$$

At present $\beta_{hdiff} = 0.3$ and a uniform eddy diffusivity $K_H|_{\Delta x=4km} = 2000 \text{ m}^2/\text{s}$ is used for the 4-km grid resolution. To compensate for larger grid sizes for coarser grids, the eddy diffusivity is modified to give:

$$K_H|_{\Delta x} = \frac{(4000)^2}{(\Delta x)^2} K_H|_{\Delta x=4km} \quad (7-78)$$

where Δx is in meters.

Obviously, the above parameterization is too simple to be realistic in a variety of atmospheric conditions. Also, depending on the numerical advection algorithms chosen, the artificial diffusivity can be quite different. This calls for several in-depth studies on following two major issues:

- (1) Quantification of realistic horizontal sub-grid scale diffusion.

The simplest approach is to assume a space independent diffusivity (e.g., $K_h = 50 \text{ m}^2/\text{s}$). Smagorinsky (1963) formulated a horizontal diffusivity that accounts for diffusion due to

distortion or stress in the horizontal wind field. For plumes which are several kilometers or more across, the Briggs (1973) parameterizations of horizontal diffusion define the diffusivity as a constant times the transport wind speed. The constant is usually based on the landuse (i.e., urban or rural) and the stability class (i.e., stable through unstable). However, it is difficult to quantify what the horizontal eddy diffusivity should be appropriate for a variety of atmospheric conditions without more detailed wind field and turbulence information.

(2) Maintaining appropriate horizontal diffusion in the presence of numerical diffusion.

Most methods for simulating advective transport in current models yield an effective numerical diffusivity much larger than physical horizontal diffusivities (Yamartino et al., 1992). Thus, the physical process may be outweighed by the numerical errors in the model. A re-assessment of this issue is required when the resolution of the model changes or when the method for simulating advection is updated. For idealized concentrations and wind fields distributions, we may be able to quantify the magnitude of the numerical diffusion in an advection scheme. However, for the more general applications, estimating the magnitude of numerical diffusion with a specific advection scheme is almost impossible. Refer to Odman (1997) for methodologies that quantify numerical diffusion errors associated with advection algorithms.

7.4 Conclusions

In this chapter, we have described numerical advection and diffusion algorithms. It has two purposes: to provide a description of the algorithms currently implemented in CCTM, and to describe the fundamental formulations that would guide future implementation of advection and diffusion modules. We encourage the development of algorithms in conservation (i.e., flux) forms to ensure compatibility of new modules with existing ones.

Because of the concerns over the non-monotonicity of BOT and YAM schemes and the mass conservation problem and diffusive nature of BOT-M, we have used PPM for a number of demonstration executions (Byun et al., 1998). Similar testing with BOT and YAM is underway. We intend to integrate other methods into the CCTM at a later time.

Also, we have identified several aspects in vertical and horizontal diffusion algorithms that require additional quantitative studies:

- Effects of the parameterization for the free troposphere;
- Importance of coordinate divergence term for vertical diffusion, in particular for the height-based constant coordinates;
- Characteristics among competing algorithms for the vertical diffusion, such as TKE and transilient turbulence schemes; and

- Practical and theoretical concerns with the horizontal diffusion algorithms.

7.5 References

Alapaty, K., R. Mathur, and D.W. Byun, 1996: A modeling study of vertical diffusion of passive and reactive tracers using local- and nonlocal-closure boundary layer schemes. *Air Pollution Modeling and Its Application XI*, ed. S.E. Gryning and F. Schiermeier. 433-442.

Alapaty, K.A., J.E. Pleim, S. Raman, D.S. Niyogi, and D.W. Byun. 1997: Simulation of atmospheric boundary layer processes using local- and nonlocal-closure schemes. *J. Applied Meteor.* Vol. **36**. 214-233.

Blackadar, A. K., 1978: Modeling pollutant transfer during daytime convection, *Preprints, Fourth Symposium on Atmospheric Turbulence, Diffusion, and Air Quality*, Reno, Am. Meteor. Soc., 443-447.

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

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

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

Briggs, G. A., 1973: Diffusion estimates for small emissions. ATDL Contribution No. 79 [Available from Atmospheric Turbulence and Diffusion Laboratory, Oak Ridge, Tenn.]

Brooks, A. N., and T. J. R. Hughes, 1982: Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.* **32**, 199-259.

Burk, S.D., and W.T. Thompson, 1989: A vertically nested regional numerical weather prediction model with second-order physics. *Mon. Wea. Rev.* **117**, 2305-2324.

Businger J. A., J. C. Wyngaard, Y. Izumi, and E. F. Bradley, 1971: Flux profile relationships in the atmospheric surface layer, *J. Atmos. Sci.*, **28**, 181-189.

Byun, D. W., 1999a: Dynamically consistent formulations in meteorological and air quality models for multi-scale atmospheric applications: I. Governing equations in a generalized coordinate system. *J. Atmos. Sci.*, (in print).

Byun, D. W., 1999b: Dynamically consistent formulations in meteorological and air quality models for multi-scale atmospheric applications: II. Mass conservation issues. *J. Atmos. Sci.*, (in print).

Byun, D. W., and R. L. Dennis, 1995: Design artifacts in Eulerian air quality models: Evaluation of the effects of layer thickness and vertical profile correction on surface ozone concentrations. *Atmos. Environ.*, **29**, 105-126.

Byun, D.W., and S.-M. Lee, 1999: Mass conservative numerical integration of trace species conservation equation: I Experiment with idealized two-dimensional flows. (in preparation)

Byun, D.W., J. Young., G. Gipson., J. Godowitch., F. Binkowsk., S. Roselle, B. Benjey, J. Pleim, J. Ching., J. Novak, C. Coats, T. Odman, A. Hanna, K. Alapaty, R. Mathur, J. McHenry, U. Shankar, S. Fine, A. Xiu, and C. Jang, 1998: Description of the Models-3 Community Multiscale Air Quality (CMAQ) model. Proceedings of the American Meteorological Society 78th Annual Meeting, Phoenix, AZ, Jan. 11-16, 1998.

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

Chang, J.S., R.A. Brost, I.S.A. Isaksen, S. Madronich, P. Middleton, W.R. Stockwell, and C.J. Walcek, 1987: A three-dimensional Eulerian acid deposition model: Physical concepts and formulation, *J. of Geophys. Res.*, **92**, 14,681-700.

Childs, P. N., and K. W. Morton, 1990: Characteristic Galerkin methods for scalar conservation laws in one dimension. *SIAM J. Numer. Anal.* **27**, 553-594.

Chock, D. P., and A. M. Dunker, 1983: A comparison of numerical methods for solving the advection equation. *Atmos. Environ.* **17**(1), 11-24.

Chock, D. P., 1985: A comparison of numerical methods for solving the advection equation - II. *Atmos. Environ.* **19**(4), 571-586.

Chock, D. P., 1991: A comparison of numerical methods for solving the advection equation-III. *Atmos. Environ.* **25A**(5/6), 853-871.

Chock, D. P., 1999: Personal communication.

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

Dabdub, D., and Seinfeld, J. H., 1994: Numerical advective schemes used in air quality models - sequential and parallel implementation. *Atmos. Environ.* **28**(20), 3369-3385.

Detering, H.W., and D. Etling, 1985: Application of E-ε turbulence model to the atmospheric boundary layer. *Bound.-Layer Meteor.*, **33**, 113-133.

Ebert, E.E., U. Schumann, and R. B. Stull, 1989: Nonlocal turbulence mixing in the convective boundary layer evaluated from large-eddy simulation. *J. Atmos. Sci.*, **46**, 2178-2207.

Flatoy, F., 1993: Balanced wind in advanced advection schemes when species with long lifetimes are transported. *Atmos. Environ.* **27A**(12), 1809-1819.

Hass, H., H.J. Jacobs, M. Memmescheimer, A. Ebel, and J.S. Change, 1991: Simulation of a wet deposition case in Europe using the European Acid Deposition Model (EURAD). *Air Pollution Modelling and Its Applications*, Vol. VIII (edited by van Dop H. and D.G. Steyn), Plenum Press, 205-213.

Holtslag, A. A., E. I. F. de Bruin, and H.-L. Pan, 1990: A high resolution air mass transformation model for short-range weather forecasting. *Mon. Weather Rev.*, **118**, 1561-1575.

Leith, C. C., 1965: Numerical simulation of the earth's atmosphere. A chapter in *Methods in Computational Physics. Vol. 4, Applications in Hydrodynamics*. p.1. ed., B. Alder, Academic Press, New York.

Löhner, R., K. Morgan, J. Peraire, and M. Vahdati, 1987: Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier-Stokes Equations. *Finite Elements in Fluids. Vol. 7*. John Wiley & Sons. 105-121.

Mellor, G.L. and T. Yamada, 1974: A hierarchy of turbulence closure models for planetary boundary layers, *J. Atmos. Sci.*, **31**, 1791-1806.

Mellor, G.L. and T. Yamada, 1982: Development of a turbulence closure model for geophysical fluid problems. *Reviews of Geophysics and Space Physics*, **20**, 851-875.

Odman, M. T., 1997: A quantitative analysis of numerical diffusion introduced by advection algorithms in air quality models. *Atmos. Environ.* **31**, 1933-1940.

Odman, M. T., 1998: Research on Numerical Transport Algorithms for Air Quality Simulation Models. EPA Report. EPA/660/R-97/142. [Available from National Exposure Research Laboratory, U.S. Environmental Protection Agency, Research Triangle Park, NC 27711].

- Odman, M. T., and A. G. Russell, 1993: A nonlinear filtering algorithm for multi-dimensional finite element pollutant advection schemes. *Atmos. Environ.* **27A**, 793-799.
- Odman, M. T., A. Xiu, and D. W. Byun, 1995: Evaluating advection schemes for use in the next generation of air quality modeling systems, In *Regional Photochemical Measurement and Modeling Studies*, eds A.J. Ranzieri and P.A. Solomon, Vol. III, Air & Waste Management Association, Pittsburgh, PA. 1386-1401.
- Oran, E., and J. Boris, 1987: Numerical Solution of Reactive Flow, New York, Elsevier.
- Otte, M.J. and J.C. Wyngaard, 1996: A general framework for an “unmixed layer” PBL model. *J. Atmos. Sci.*, **53**, 2652-2670.
- Pleim, J.E. and J. Chang, 1992: A non-local closure model for vertical mixing in the convective boundary layer. *Atmos. Envi.*, **26A**, 965-981.
- Raymond, W.H. and R.B. Stull, 1990: Application of transilient turbulence theory to mesoscale numerical weather forecasting. *Mon. Weather Rev.*, **118**, 2471-2499.
- Rood, R. B., 1987: Numerical advection algorithms and their role in atmospheric transport and chemistry models. *Rev. Geophysics* **25**(1), 71-100.
- Schumann, U., 1989: Large-eddy simulation of turbulent diffusion with chemical reactions in the convective boundary layer. *Atmos. Envi.*, **26A**, 965-981.
- Smagorinsky, J., 1983: General circulation experiments with the primitive equations: 1. The basic experiment. *Mon. Wea. Rev.* **91**, 99-164.
- Stull, R. B., 1988: *An Introduction to Boundary Layer Meteorology*. Kluwer Academic, 666 pp.
- Stull, R. B., 1993: Review of non-local mixing in turbulent atmosphere: Transilient turbulence theory. *Bound.-Layer Meteor.*, **62**, 21-96.
- Stull, R. B., and A.G.M. Driedonk, 1987: Applications of the transilient turbulence parameterization to atmospheric boundary layer simulations. *Bound.-Layer Meteor.*, **40**, 209-239.
- Tremback, G. J., Powell, J., Cotton, W. R. and Pielke, R. A., 1987: The forward-in-time upstream advection scheme: extension to higher orders. *Mon. Wea. Rev.* **115**, 540-555.

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

Yamartino, R.J., J.S. Scire, G.R. Carmichael, and Y.S. Chang, 1992: The CALGRID mesoscale photochemical grid model - Part I. Model formulation, *Atmos. Environ.* **26A**, 1493-1512.

Yanenko, N. N., 1971: *The Method of Fractional Steps*. Springer-Verlag, New York, 160pp.

Zhang, Q., and Stull, R. B. Stull, 1992: Alternative nonlocal descriptions of boundary-layer evolution. *J. Atmos. Sci.*, **49**, 2267-2281.

This chapter is taken from *Science Algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) Modeling System*, edited by D. W. Byun and J. K. S. Ching, 1999.

Appendix 7A Numerical Solvers for Diffusion Equations

In this appendix we describe numerical procedures for eddy diffusion, the Blackadar mixing scheme, and the asymmetric convective model.

7A.1 Stability of Tridiagonal Solver

The eddy diffusion formulation requires the solution of the linear equation

$$\mathbf{A}\mathbf{q} = \mathbf{b} \quad (7A-1)$$

where:

$$\mathbf{A} = \begin{pmatrix} d_1 & c_1 & 0 & 0 & \cdots & 0 \\ a_2 & d_2 & c_2 & 0 & \cdots & 0 \\ 0 & a_3 & d_3 & c_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & a_{n-1} & d_{n-1} & c_{n-1} \\ 0 & \cdots & 0 & 0 & a_n & d_n \end{pmatrix};$$

$$\mathbf{q} = (q_1 \quad q_2 \quad q_3 \quad \cdots \quad q_n)^T; \text{ and}$$

$$\mathbf{b} = (b_1 \quad b_2 \quad b_3 \quad \cdots \quad b_n)^T.$$

The system Equation 7A-1 can be solved by the Thomas algorithm (Gaussian elimination of a tridiagonal matrix without pivoting) followed by back substitution. Assume that the following stage of the elimination has been reached:

$$\alpha_{j-1}q_{j-1} + c_{j-1}q_j = \beta_{j-1} \quad (7A-2a)$$

$$a_jq_{j-1} + d_jq_j + c_jq_{j+1} = b_j \quad (7A-2b)$$

where $\alpha_1 = d_1$ and $\beta_1 = b_1$.

For $j=2,3,\dots,n-1$, eliminating q_{j-1} from Equations 7A-2a,b leads to:

$$\alpha_jq_j + c_jq_{j+1} = \beta_j \quad (7A-3a)$$

where

$$\alpha_j = d_j - \frac{a_jc_{j-1}}{\alpha_{j-1}} \quad (7A-3b)$$

$$\beta_j = b_j - \frac{a_j \beta_{j-1}}{\alpha_{j-1}} \quad (7A-3c)$$

The last pair of simultaneous equations are:

$$\alpha_{n-1} q_{n-1} + c_n q_n = \beta_{n-1} \quad (7A-4a)$$

$$a_n q_{n-1} + d_n q_n = b_n \quad (7A-4b)$$

Eliminating q_{n-1} gives:

$$\alpha_n q_n = \beta_n \quad (7A-5a)$$

and with Equation 7A-3a, we can obtain the solution by back substitution, i.e., $j = n-1, n-2, \dots, 1$:

$$q_j = \frac{\beta_j + c_j q_{j+1}}{\alpha_j} \quad (7A-5b)$$

The algorithm described above is stable for the tridiagonal system if:

- (i) $d_j > 0$, $a_j < 0$, and $c_j < 0$;
- (ii) $d_j > -(a_{j+1} + c_{j-1})$ for $j=1, 2, \dots, n-1$, defining $c_o = a_n = 0$; and
- (iii) $d_j > -(a_j + c_j)$ for $j=1, 2, \dots, n-1$, defining $a_1 = c_{n-1} = 0$.

The first two conditions ensure that the forward elimination is stable and the first and third conditions ensure that the back substitution is stable.

To prove that the forward elimination procedure is stable, it is necessary to show that the moduli of the multipliers $m_j = -a_j / \alpha_{j-1}$ used to eliminate $q_1, q_2, \dots$ are positive and less than or equal to one. From Equation 7A-2, we get:

$$\alpha_j = d_j + m_j c_{j-1} \quad (7A-6a)$$

and

$$m_{j+1} = -\frac{a_{j+1}}{\alpha_j} = \frac{-a_{j+1}}{d_j + m_j c_{j-1}} \quad (7A-6b)$$

Then, since $d_1 > -a_2 > 0 = c_o$:

$$0 < m_2 < -\frac{a_2}{d_1} < 1 \quad (7A-7a)$$

Similarly, we have:

$$0 < m_3 < \frac{-a_3}{d_2 + m_2 c_1} < \frac{-a_3}{d_2 + c_1} < \frac{-a_3}{-(a_3 + c_1) + c_1} = 1 \quad (7A-7b)$$

since $d_2 > -(a_3 + c_1)$. In this way, we can show that $0 < m_j < 1$ for $j=1, 2, \dots, n-1$.

For back substitution, we can write:

$$q_j = \frac{1}{\alpha_j}(\beta_j - c_j q_{j+1}) = p_{j+1} q_{j+1} + \frac{\beta_j}{\alpha_j} \quad (7A-8)$$

with $a_1 = c_{n-1} = 0$ for $j=1, 2, \dots, n-1$.

There will be no build-up of errors in the back substitution process if $|p_{j+1}| < 1$, where:

$$p_{j+1} = \frac{-c_j}{\alpha_j} = \frac{-c_j}{d_j + a_j p_j} \quad (7A-9a)$$

Now, $0 < p_2 = -c_1 / d_1 < 1$, since $a_1 = 0$ and $d_1 > -c_1$ by hypothesis. Then:

$$p_3 = \frac{-c_2}{d_2 + a_2 p_2} \quad (7A-9b)$$

As $-c_2 > 0$, $0 < p_2 < 1$, and $d_2 > -a_2 > 0$, it follows that:

$$0 < p_3 < \frac{-c_2}{d_2 + a_2} < \frac{-c_2}{-(a_2 + c_2) - a_2} = 1 \quad (7A-10)$$

Similarly, we can show that $0 < p_j < 1$ for $j=1, 2, \dots, n-1$.

7A.2 Solver for Blackadar Scheme

The Blackadar scheme requires solving the sparse linear matrix equation $\mathbf{Aq}=\mathbf{b}$ of the form

$$\mathbf{A} = \begin{pmatrix} d_1 & f_2 & f_3 & \cdots & f_{n-1} & f_n \\ e_2 & d_2 & 0 & 0 & \cdots & 0 \\ e_3 & 0 & d_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ e_{n-1} & \cdots & 0 & 0 & d_{n-1} & 0 \\ e_n & \cdots & 0 & 0 & 0 & d_n \end{pmatrix} \quad (7A-11)$$

The solver for this system uses a similar numerical procedure as for the tridiagonal system. The elements of the linear set of equations are related as follows:

$$d_1 q_1 + \sum_{j=2}^n f_j q_j = b_1 \quad (7A-12a)$$

and for $2 \leq j \leq n$:

$$e_j q_1 + d_j q_j = b_j \quad (7A-12b)$$

By substituting q_j with q_1 for each j , we get:

$$q_1 = \frac{b_1 - \sum_{j=2}^n b_j f_j / d_j}{d_1 - \sum_{j=2}^n e_j f_j / d_j} \quad (7A-13a)$$

and for $2 \leq j \leq n$:

$$q_j = (b_j - e_j q_1) / d_j \quad (7A-13b)$$

Note that Equation 7A-13b involves neither a forward nor a backward substitution loop.

7A.3 Solver for Asymmetric Convective Model (ACM)

The Asymmetric Convective Model requires solving the linear matrix equation $\mathbf{Aq}=\mathbf{b}$ with a sparse matrix of the form:

$$\mathbf{A} = \begin{pmatrix} d_1 & c_2 & 0 & \cdots & 0 & 0 \\ e_2 & d_2 & c_2 & 0 & \cdots & 0 \\ e_3 & 0 & d_3 & c_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ e_{n-1} & \cdots & 0 & 0 & d_{n-1} & c_{n-1} \\ e_n & \cdots & 0 & 0 & 0 & d_n \end{pmatrix} \quad (7A-14)$$

The solver for this system is based on a numerical procedure similar to the tridiagonal solver. The relation among the variables q_i are given as:

$$d_1 q_1 + c_1 q_2 = b_1 \quad (7A-15a)$$

$$e_n q_1 + d_n q_n = b_n \quad (7A-15b)$$

and for $2 \leq j \leq n-1$:

$$e_j q_1 + d_j q_j + c_j q_{j+1} = b_j \quad (7A-15c)$$

Assume that the following stage of the elimination has been reached for $1 \leq j \leq n-1$:

$$\alpha_j q_1 + c_j q_{j+1} = \beta_j \quad (7A-16a)$$

where:

$$\alpha_j = e_j - \frac{d_j}{c_{j-1}} \alpha_{j-1} \quad (7A-16b)$$

$$\beta_j = b_j - \frac{d_j}{c_{j-1}} \beta_{j-1} \quad (7A-16c)$$

with $\alpha_1 = d_1$ and $\beta_1 = b_1$. From Equation 7A-16a for $j = n-1$ and Equation 7A-15b, q_1 is found to be:

$$q_1 = \frac{d_n \beta_{n-1} - b_n c_{n-1}}{d_n \alpha_{n-1} - e_n c_{n-1}} \quad (7A-17a)$$

and all other q_i s are computed with:

$$q_{j+1} = \frac{\beta_j - \alpha_j q_1}{c_j} \quad (7A-17b)$$

for $1 \leq j \leq n - 1$. The final substitution stage can be implemented either in a forward or backward sweep.