



Resolution of pollutant concentrations in the boundary layer using a fully 3D adaptive gridding technique

S. Ghorai^a, A.S. Tomlin^{a,*}, M. Berzins^b

^a*Department of Fuel and Energy, University of Leeds, Leeds LS2 9JT, UK*

^b*School of Computer Studies, University of Leeds, Leeds LS2 9JT, UK*

Received 6 August 1999; received in revised form 7 December 1999; accepted 20 December 1999

Abstract

This paper investigates the solution of a 3D atmospheric dispersion problem using a time-dependent adaptive gridding technique. A cell-vertex finite volume scheme based on tetrahedral elements is used to solve the atmospheric diffusion equation. Preliminary studies of dispersion from a single source in stable, unstable and neutral boundary layers have been carried out. The results show the efficiency of using adaptive grids to represent accurately the structures of plumes in the boundary layer and also the advantage of this method compared to fixed methods for mesh refinement. Some comments about the interpolation of input data such as wind fields onto unstructured meshes are also made. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Adaptive meshes; Dispersion; Plume; Boundary layer; Unstructured

1. Introduction

The issue of mesh resolution in regional scale air pollution models is an important one since there are many examples of the detrimental effects of using coarse meshes on solution accuracy. Previous work has shown (Talat, 1997; Tomlin et al., 1999; Hart et al., 1998) that coarse horizontal resolution can have the effect of increasing horizontal diffusion to values many times greater than that described by models, resulting in the smearing of pollutant profiles and an underestimation of maximum concentration levels. For reactive pollutants the effects can be compounded by nonlinear chemical reactions and inaccurate predictions of secondary species budgets can occur. The effects of mesh resolution have been well noted by the atmospheric modelling community and attempts have been made to improve mesh resolution at the same time as trying to avoid excessive extra computational work. The usual approach is to use nested or telescopic grids where the mesh is refined in certain regions of the horizontal domain which are considered of

interest (Jacobs et al., 1995; Rajaona et al., 1998). This may include for example regions of high emissions such as urban areas, or close to regions where significant monitoring is taking place. Previous work has shown however (Tomlin et al., 1997) that such telescopic grids often cannot resolve plume structures occurring outside of the nested regions and that adaptive refinement in the horizontal domain can provide higher accuracy without entailing large extra computational costs.

In the vertical domain usually a stretched mesh is used, placing more solution points close to the ground. As in the horizontal domain, the resolution of the mesh in the vertical direction affects the vertical mixing of pollutant species. The use of adaptive meshes in the vertical domain has so far received little attention. This paper describes the application of a fully adaptive 3D unstructured dispersion code to test problems describing the dispersion of pollutants from a single source due to typical boundary layer wind profiles. The solution method is a 3D finite volume cell-vertex approach based on a tetrahedral mesh. The test problems have been designed to determine the importance of mesh structure on both horizontal and vertical mixing for typical meteorological conditions. The paper therefore concentrates

* Corresponding author. Fax: + 44-113-244-0572.

on simple scenarios and on the comparison between different mesh structures in order to highlight the solution accuracy of different modelling approaches. In particular, we compare a telescoping gridding method to a fully adaptive one. Section 2 describes the method of solution and Section 3 contains a description of the mesh adaption strategy. The flow integration scheme is described in Section 4 and 5 explains the implicit–explicit method used to solve the transport equation. Section 6 contains the test examples and finally, we draw conclusions in Section 7 regarding the importance of adaptive methods in solving 3D atmospheric flow problems.

2. Method of solution

The atmospheric diffusion equation in three space dimensions is given by

$$\frac{\partial \mathbf{c}}{\partial t} + \frac{\partial(u\mathbf{c})}{\partial x} + \frac{\partial(v\mathbf{c})}{\partial y} + \frac{\partial(w\mathbf{c})}{\partial z} = \frac{\partial}{\partial x} \left(K_x \frac{\partial \mathbf{c}}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_y \frac{\partial \mathbf{c}}{\partial y} \right) + \frac{\partial}{\partial z} \left(K_z \frac{\partial \mathbf{c}}{\partial z} \right) + \mathbf{S}, \quad (1)$$

where \mathbf{c} is a column vector representing the concentrations with the dimension of \mathbf{c} equal to the number of species. The scalars u, v, w are wind velocities and K_x, K_y, K_z are turbulent diffusivity coefficients in the x, y and z directions with respect to a cartesian coordinate system with the z -axis pointing upwards. The vector \mathbf{S} represent the source term which may also contain reaction terms which are usually nonlinear, and deposition terms. Our approach uses a fully 3D unstructured mesh based on tetrahedral elements. A cell-vertex finite volume scheme has been chosen so that the number of variables per partial differential equation is less than that of a cell centred scheme. The solution method consists of two major parts, a mesh adaption routine and a flow integration routine. The mesh adaption routine changes the connectivity in the data structure of the mesh in response to changes in the solutions. The flow integration routine advances the solution in time. These two major parts are described in detail in the next sections. Each component of the diffusion equations (1) is discretized using the same method. Hence, for simplicity, instead of treating the vector \mathbf{c} , we choose one of its components, c say, and describe its discretization.

3. Mesh adaption

The cell-vertex scheme approach is hierarchical in nature (Speares and Berzins, 1997; Biswas and Strawn,

1994) and is applicable to meshes constructed from tetrahedral-shaped elements. The basic mesh objects of *nodes, edges, faces* and *elements*, which together form the computational domain, map onto the data objects within the adaption algorithm tree data structure. The data objects contain all flow and connectivity information sufficient to adapt the mesh structure and flow solution by either local *refinement* or *derefinement* procedures. The mesh adaption strategy assumes that there exists a “good quality” initial unstructured mesh covering the computational domain. The refinement process adds nodes to this base level mesh by edge, face and element subdivision, with each change to the mesh being tracked within the code data structure by the construction of a data hierarchy. The derefinement is the inverse of refinement, where nodes, faces and elements are removed from the mesh by working back through the local mesh refinement hierarchy.

The main adaption is driven by refining and derefining element edges. Thus, if an edge is refined by the addition of a node along its length, then all the elements which share the (parent) edge under refinement must be refined. In the case of derefinement all the elements which share the node being removed must be derefined. Numerical criteria derived from the flow field will mark an edge to either refine, derefine or remain unchanged. It is necessary to make sure the edges targeted for refinement and derefinement pass various conditions prior to their adaption. These conditions effectively decouple the regions of mesh refinement from those of derefinement, meaning that for example, an element is not both derefined and refined in the same adaption step.

For reasons of both tetrahedral quality control and algorithm simplicity only two types of element subdivisions are used (Speares and Berzins, 1997). The first type of subdivision is called *regular subdivision* where a new node bisects each edge of the parent element resulting in eight new elements. The second type of dissection, *green subdivision*, introduces an extra node into parent tetrahedron, which is subsequently connected to all the parent vertices and any additional nodes which bisect the parent edges. Green refinement removes inconsistently connected or “hanging” nodes without the introduction of additional edge refinement. The green elements may be of poorer quality in terms of aspect ratio and so the green element may not be further refined. Fig. 1 demonstrates regular and green refinement for a tetrahedron. The five possible refinement possibilities (if all the edges are refined then the parent element is regularly refined) give rise to between 6 and 14 child green elements.

The choice of adaption criteria is very important since it can produce a large or small number of nodes depending on the condition used to flag an edge for adaption. Also, when there are a large number of species, the choice of a given criteria might result in high resolution for some

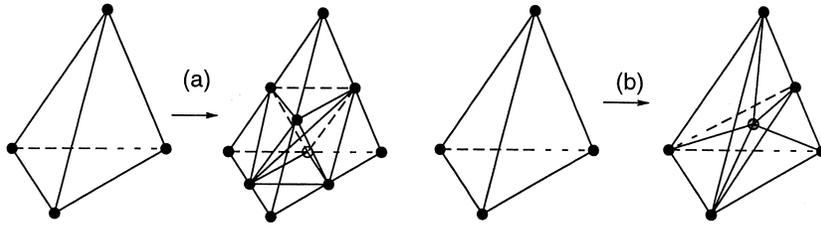


Fig. 1. (a) Regular refinement based on the subdivision of tetrahedron by dissection of interior diagonal (1 : 8) and (b) “green” refinement by addition of an interior node (1 : 6).

species but low resolution for the other species. Let 0 and i be the nodes for a given edge $e(0, i)$. We calculate $\text{tol } g$ and $\text{tol } c$ by

$$\text{tol } g = \frac{|(c)_0 - (c)_i|}{\text{dist}} \quad \text{and} \quad \text{tol } c = \frac{(c)_0 + (c)_i}{2},$$

where dist is the length of the edge $e(0, i)$. We refine the edge $e(0, i)$ if $\text{tol } g$ and $\text{tol } c$ exceed some tolerances, otherwise it is derefined. Also a maximum level of refinement is specified at the beginning so that if an edge is targeted for refinement but it is at the maximum level, then it is kept unchanged.

Suppose we have two edges with $\text{tol } g = 100$ and 200 . If we take the tolerance parameter, T_g say, for $\text{tol } g$ equal to 150 , then only the second edge is refined to maximum level. On the other hand, if $T_g = 50$, then both edges are refined to maximum level. We expect that the solution error for the edge with $\text{tol } g = 200$ is greater than the error in the edge with $\text{tol } g = 100$. It might be advantageous to use two sets of $T_g = 50$ and 150 . If $\text{tol } g > 150$, then we refine an edge to maximum level and if $50 < \text{tol } g < 150$, then we refine an edge to the level just lower than the maximum level. Thus the idea is to refine to the maximum level in the steepest gradient regions but to lower levels in the regions of less steep gradients.

4. Flow integration

The diffusion equation is discretized over special volumes that form the dual mesh. The dual mesh is formed by constructing non-overlapping volumes, referred to as dual cells, around each node. The dual mesh for a tetrahedral grid is constructed by dividing each tetrahedron into four hexahedra of equal volumes, by connecting the mid-edge points, face-centroids and the centroid of the tetrahedron. The control volume around a node 0 is thus formed by a polyhedral hull which is the union of all such hexahedra that share that node. The quadrilateral faces that constitute the dual mesh may not all be planer.

4.1. Flux evaluation using edge-based operation

The evaluation of flux around a dual cell can be cast in an edge-based operation. Let us discretize the divergence term

$$\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z}$$

over the control volume Ω_0 enclosing the node 0. This divergence form is converted to flux form using the Gauss divergence theorem:

$$\int_{\Omega_0} \left(\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} \right) d\Omega = \int_{\partial\Omega_0} (fn_x + gn_y + hn_z) dS = \sum_k (fS_x + gS_y + hS_z), \quad (2)$$

where the summation is over all the dual mesh faces that form the boundary of the control volume around the node 0 and the areas S_x, S_y, S_z are projections of the dual quadrilateral face.

Consider edge i , formed by nodes 0 and $N(i)$. The quadrilateral faces of the dual mesh that are connected to the edge at its mid-point P are shown in Fig. 2. The number of such quadrilateral faces attached to an edge depends on the number of neighbouring tetrahedra. There are four tetrahedra sharing the edge i in Fig. 2. The projected area, A_i , associated with the edge i is calculated

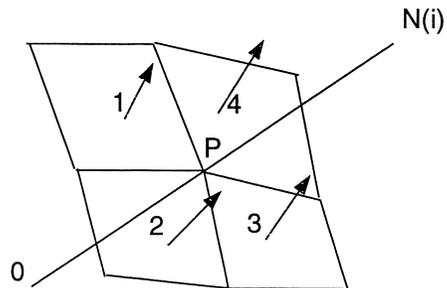


Fig. 2. Dual mesh faces attached to an edge.

in terms of the quadrilateral face areas, $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$, as

$$(A_i)_x = \sum_{j=1}^4 (a_j)_x, \quad (A_i)_y = \sum_{j=1}^4 (a_j)_y, \quad (A_i)_z = \sum_{j=1}^4 (a_j)_z. \quad (3)$$

The projections are computed so that the area vector points outward from the control volume surface associated with a node. The boundary of the control volume around the node 0 is formed by the union of all such areas \mathbf{A}_i associated with each edge i that share the node 0. The contribution of the edge i to the fluxes across the faces of the control volume surrounding the node 0 is given by

$$f_p(A_i)_x + g_p(A_i)_y + h_p(A_i)_z.$$

Hence Eq. (2) is replaced by

$$\int_{\Omega_0} \left(\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} \right) d\Omega = \sum_i (f_p(A_i)_x + g_p(A_i)_y + h_p(A_i)_z), \quad (4)$$

where the sum is over the edges that share the node 0. The fluxes are thus calculated on an edge-wise basis and conservation is enforced by producing a positive flux contribution to one node and an equal but opposite contribution to the other node that forms the edge.

4.2. Adjustments of wind field

In an atmospheric pollution model, we often use observed wind data which are not mass conservative. Even mass conserving wind data might not be mass conservative in the numerical sense when interpolated onto an unstructured grid. Thus, we want to adjust the wind data in such a way that the observed data are minimally changed while still satisfying the mass conservation property numerically. If u, v, w are the wind velocities, then they must satisfy

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \quad (5)$$

In the observed wind field data, usually the horizontal velocities u, v are given. A popular technique for obtaining the vertical velocity is the direct-differencing method. This procedure numerically calculates the vertical velocity taking $w = 0$ at $z = 0$ to initiate the calculation. This method is simple to use in regular cartesian meshes, since it produces one unknown in w per node. For unstructured meshes, the number of unknowns per node depends on the number of nodes sharing it. Here we enforce mass conservation using the variational calculus technique of Mathur and Peters (1990). The technique attempts to adjust the wind velocity in a manner such that the interpolated data are minimally changed in a least-squares

sense, and at the same time, the adjusted values satisfy the mass conservation constraint.

Let u^0, v^0, w^0 be the interpolated wind velocity components and u, v, w be the corresponding adjusted values. The difference between the interpolated and adjusted field may be expressed in a least-squares sense as

$$\alpha_1^2(u - u^0)^2 + \alpha_2^2(v - v^0)^2 + \alpha_3^2(w - w^0)^2, \quad (6)$$

where α_1, α_2 and α_3 are weighting functions (Mathur and Peters, 1990). Let G denote the mass conservation constraint (Eq. (5)) to be satisfied, then the adjustment functional may be expressed as

$$I = \int_{\Omega} [(\alpha_1^2(u - u^0)^2 + \alpha_2^2(v - v^0)^2 + \alpha_3^2(w - w^0)^2) + \lambda G] d\Omega, \quad (7)$$

where the constraint G is introduced through the Lagrange multiplier λ . The resulting Euler–Lagrange equations obtained by setting $\delta I = 0$ can be expressed as

$$u_i = u_i^0 + \frac{1}{2\alpha_i^2} \frac{\partial \lambda}{\partial x_i}, \quad (8)$$

where $i = 1, 2, 3$ denote the x, y, z components, respectively. Introducing Eq. (8) into Eq. (5) we get

$$\begin{aligned} \frac{\partial}{\partial x} \left(\frac{\partial \lambda}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial \lambda}{\partial y} \right) + \frac{\partial}{\partial z} \left(\frac{\alpha_1^2}{\alpha_3^2} \frac{\partial \lambda}{\partial z} \right) \\ = -2\alpha_1^2 \left[\frac{\partial u^0}{\partial x} + \frac{\partial v^0}{\partial y} + \frac{\partial w^0}{\partial z} \right]. \end{aligned} \quad (9)$$

Eq. (9) is similar to the diffusion equation with diffusivities 1, 1 and α_1^2/α_3^2 and hence its discretization is similar to the diffusion scheme.

We have adjusted one-dimensional stable, neutral and unstable boundary layer wind velocities which are a function of z . The wind velocity is mass conservative analytically. It remains mass conservative in the numerical sense in the base mesh, since the unstructured base mesh is regular, but is not mass conservative once the grid is refined or derefined. A representative one-dimensional neutral boundary layer velocity is shown in Fig. 3(b). The velocity field has to be adjusted in the refinement region, but away from the refinement region the velocity remains almost unchanged. The base grid spacings in the vertical direction increase with height. Thus, the grid quality near the ground is worse due to the large aspect ratio of the tetrahedra. The velocity corrections decrease with height as the refinement region moves upwards. Suppose, we have a refinement region at 150 m height. The maximum corrections are 12, 14 and 0.06 cm s⁻¹, respectively, for the u, v and w components. For a refinement region at 600 m height the corresponding components are 11, 11 and 0.03 cm s⁻¹. Finally, the corresponding corrections

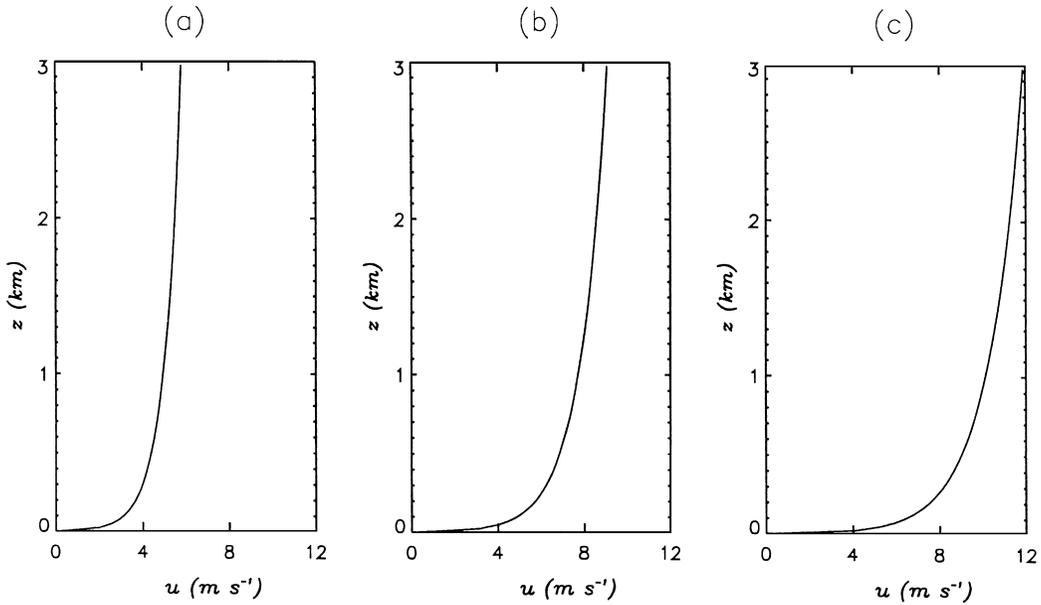


Fig. 3. A representative variation of wind with height for (a) stable, (b) neutral and (c) unstable boundary layers.

decreases to 0.3, 0.2 and 0.0002 cm s^{-1} at 1.8 km height. The neutral boundary layer velocity increases from 0 to 9 m s^{-1} as z increases from 0 to 3 km and so the velocity corrections are small.

Here we have used an idealized wind profile. In reality, the wind components are calculated directly from a meteorological model. A meteorological driver with nodes located at the vertices of the adaptive grid would be advantageous, but the time spent on solution may be prohibitively large. Extra effort would be required to discretize the equations for the meteorological model on an unstructured grid. The dispersion model described here would however work efficiently with an off-line model, where the meteorological variables are calculated on a regular interval of time (say 3 h) on a structured grid. The calculated wind fields would be interpolated onto the adaptive grid and adjustments made using the variational technique described in this section to ensure mass conservation.

4.3. The advection scheme

Our aim is to discretize the term

$$\int_{\Omega_0} \left[\frac{\partial(u c)}{\partial x} + \frac{\partial(v c)}{\partial y} + \frac{\partial(w c)}{\partial z} \right] d\Omega \equiv \int_{\Omega_0} \nabla \cdot \mathbf{F} c d\Omega,$$

where

$$\mathbf{F} = (u\hat{i} + v\hat{j} + w\hat{k}).$$

Using Eq. (4), the above equation can be written as

$$\sum_i [u_p (A_i)_x + v_p (A_i)_y + w_p (A_i)_z] (c)_p = \sum_i (\mathbf{F}_p \cdot \mathbf{A}_i) (c)_p, \tag{10}$$

where \mathbf{A}_i is called the edge-normal associated with the edge i and the sum is over all the edges sharing node 0 with control volume Ω_0 . Let $(c)_p^-$ denote the value of c at P interpolated from node 0 and $(c)_p^+$ denote the value at P interpolated from node i (refer to Fig. 2). Then an upwind version of Eq. (10) is given by

$$\sum_i \left[\frac{(\mathbf{F}_p \cdot \mathbf{A}_i + |\mathbf{F}_p \cdot \mathbf{A}_i|)}{2} (c)_p^- + \frac{(\mathbf{F}_p \cdot \mathbf{A}_i - |\mathbf{F}_p \cdot \mathbf{A}_i|)}{2} (c)_p^+ \right]. \tag{11}$$

Let \mathbf{r}_0 and \mathbf{r}_i denote the position vectors of the nodes 0 and i , respectively. If \mathbf{r}_p is the position vector of P , then

$$\mathbf{r}_p = \frac{\mathbf{r}_0 + \mathbf{r}_i}{2}.$$

The extrapolation formulae are

$$(c)_p^- = (c)_0 + \Phi_0 (\nabla c)_0 \cdot (\mathbf{r}_p - \mathbf{r}_0),$$

$$(c)_p^+ = (c)_i + \Phi_i (\nabla c)_i \cdot (\mathbf{r}_p - \mathbf{r}_i),$$

where Φ_0 and Φ_i are the limiter functions at nodes 0 and i , respectively. The limiter function at a node is chosen

such that the reconstructed value is bounded by the values at the node and its neighbours (Barth and Jespersen, 1989). Specifically, let us consider node 0. Let N_0 denote the set of neighbouring nodes to the node 0. First, we compute the maximum and minimum of all adjacent neighbours.

$$(c)_0^{\min} = \min_{i \in N_0} ((c)_0, (c)_i),$$

$$(c)_0^{\max} = \max_{i \in N_0} ((c)_0, (c)_i)$$

and require that

$$(c)_0^{\min} \leq (c)_p^- \leq (c)_0^{\max},$$

where $(c)_p^-$ is computed at each of the edges sharing the node 0. Considering the edge sharing the node 0 and i , we calculate ϕ_i according to

$$\phi_i = \begin{cases} \min\left(1, \frac{(c)_0^{\max} - (c)_0}{(\nabla c)_0 \cdot (\mathbf{r}_p - \mathbf{r}_0)}\right) & \text{if } (\nabla c)_0 \cdot (\mathbf{r}_p - \mathbf{r}_0) > 0, \\ \min\left(1, \frac{(c)_0 - (c)_0^{\min}}{(\nabla c)_0 \cdot (\mathbf{r}_p - \mathbf{r}_0)}\right) & \text{if } (\nabla c)_0 \cdot (\mathbf{r}_p - \mathbf{r}_0) < 0, \\ 1 & \text{if } (\nabla c)_0 \cdot (\mathbf{r}_p - \mathbf{r}_0) = 0. \end{cases} \quad (12)$$

Now, we calculate Φ_0 by

$$\Phi_0 = \min_i \{\phi_i\},$$

where the minimum is taken over all the edges that share the node 0. The calculation of limiter functions and gradients at the nodes is not on a node-by-node basis which is CPU intensive. Instead, they are calculated in an edge-based operation. The time step for the advection scheme is chosen so that it satisfies the CFL condition (Wierse, 1997). Again, let us consider the node 0. We define S_0^1 and S_0^2 as follows:

$$S_0^1 = \sum_{\substack{i \in N_0 \\ (c_s)_p^- \geq (c_s)_0}} - \frac{(\mathbf{F}_p \cdot \mathbf{A}_i - |\mathbf{F}_p \cdot \mathbf{A}_i|)}{2} + \sum_{\substack{i \in N_0 \\ (c_s)_p^- \leq (c_s)_0}} \frac{(\mathbf{F}_p \cdot \mathbf{A}_i + |\mathbf{F}_p \cdot \mathbf{A}_i|)}{2},$$

$$S_0^2 = \sum_{\substack{i \in N_0 \\ (c_s)_p^- < (c_s)_0}} - \frac{(\mathbf{F}_p \cdot \mathbf{A}_i - |\mathbf{F}_p \cdot \mathbf{A}_i|)}{2} + \sum_{\substack{i \in N_0 \\ (c_s)_p^- > (c_s)_0}} \frac{(\mathbf{F}_p \cdot \mathbf{A}_i + |\mathbf{F}_p \cdot \mathbf{A}_i|)}{2}.$$

The time step at node 0 is Δt_0 and is given by

$$\Delta t_0 = \min\left(\frac{V_0}{S_0^1}, \frac{V_0}{S_0^2}\right)$$

and in a similar way, the time step at every node is calculated. The minimum of the time steps over all the vertices constitutes the time step for the advection scheme. Again this computation can be cast into an edge-based operation.

4.4. Diffusion scheme

The diffusion term is discretized as

$$\int_{\Omega_0} \left[\frac{\partial}{\partial x} \left(K_x \frac{\partial c}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_y \frac{\partial c}{\partial y} \right) + \frac{\partial}{\partial z} \left(K_z \frac{\partial c}{\partial z} \right) \right] d\Omega \equiv \int_{\Omega_0} \nabla \cdot \mathbf{R} d\Omega,$$

where

$$\mathbf{R} = K_x \frac{\partial c}{\partial x} \hat{i} + K_y \frac{\partial c}{\partial y} \hat{j} + K_z \frac{\partial c}{\partial z} \hat{k}.$$

Here K_x, K_y, K_z are the turbulent diffusivity coefficients. Using Eq. (4), the above equation can be written as

$$\sum_i \left[\left(K_x \frac{\partial c}{\partial x} \right)_p (A_i)_x + \left(K_y \frac{\partial c}{\partial y} \right)_p (A_i)_y + \left(K_z \frac{\partial c}{\partial z} \right)_p (A_i)_z \right], \quad (13)$$

where \mathbf{A}_i is the edge-normal associated with the edge i and the sum is over all the edges sharing the node 0 with control volume Ω_0 . The first term in Eq. (13) can be expressed as

$$\left(K_x \frac{\partial c}{\partial x} \right)_p = (K_x)_p \left(\frac{\partial c}{\partial x} \right)_p$$

with similar expressions for the other two terms.

Thus, we need to calculate the edge gradient term $(\nabla c)_p$ for the edge $e(0, i)$. For this, we again apply the Gauss divergence theorem over a control volume. For edge $e(0, i)$, this control volume is the union of elements sharing the edge $e(0, i)$. Once we calculate $(\nabla c)_p$, its components are substituted in Eq. (13) and the same is done for all the edges that share the node 0. The discretized version of Eq. (13) can be expressed as

$$\sum_i a_i (c)_i - a_0 (c)_0, \quad (14)$$

where the sum is over the nodes that share the node 0 and

$$a_0 = \sum_i a_i.$$

Thus, we need to store only the values of a_i and these values can be attached to the edge pointer. Importantly,

the values of a_i depend only on the mesh geometry and turbulent diffusivity coefficients. This is very useful and results in the use of less CPU time since the values of a_i between two time steps remain the same as long as there is no mesh adaption between these steps.

5. Solution of the atmospheric diffusion equation

Although in two space dimensional calculations we have used sophisticated space–time error control techniques (Berzins et al., 1998; Tomlin et al., 1997), the computational cost and the need to evaluate spatial mesh adaption has led us to focus here on less costly methods. It is thus primarily for computational efficiency that, in common with many others, we have used an operator splitting technique. In this approach, the chemistry is decoupled from the transport. The nonlinear chemistry part gives rise to stiff ordinary differential equations. Here we consider solutions for the transport only, the solutions for the chemistry will be discussed elsewhere.

If c^n denotes the species concentration at time level n , then the species concentration at the next time step is given by

$$c^{n+1} = c^n + \tau g(c) + \tau f(c) + \tau S, \quad (15)$$

where τ is the time step and $g(c)$ is the advection operator and $f(c)$ is the diffusion operator. In a fully explicit scheme, f and g are evaluated using values at the time level n . However, the time restriction for stability due to vertical diffusion is severe since the grid spacings along the vertical can be small. Hence we use an implicit–explicit formulation for Eq. (15), where the advection is evaluated explicitly and the diffusion is calculated implicitly. Again let us consider node i and let $N(i)$ be the set of nodes sharing the node i . The discretized form of the advection–diffusion equation for c at the node i is given by

$$\left(\frac{1}{\tau} + a_i\right)(c^{n+1})_i = \sum_{j \in N(i), j \neq i} a_j (c^{n+1})_j + Q_i^n, \quad (16)$$

where i is varied over all the nodes and

$$Q_i^n = \left[\frac{c^n}{\tau} + g(c^n) + S \right]_i.$$

The time step τ is chosen as the time step due to advection only. Its value mainly depends on the wind speed and the vertical mesh spacings near the source. For the base mesh (described in the next section) used in the test examples, τ is ≈ 35 s for the stable boundary layer but decreases to ≈ 18 s for the unstable boundary layer. Thus, the time step is smaller for higher wind speed and vice versa. The system of equations given by Eq. (16) is solved using the Gauss–Seidel iteration technique with

over-relaxation and the iteration is stopped when the relative error is less than some prescribed tolerance. The advantage of this method is its computational efficiency. The disadvantage is that we are introducing an extra time integration and splitting error which is not easily quantified.

The error due to splitting can be reduced if the splitting time step is small or the splitting is done at level of the nonlinear equations (see Tomlin et al., 1997). The simulation of reactive flow problems using these techniques will be discussed in greater detail elsewhere.

6. Test examples

The advection scheme has previously been tested by advecting a puff of NO around a horizontal circle without any diffusion (Tomlin et al., 1999). The results showed that the peak almost remains constant suggesting that very little artificial diffusion has taken place for refined meshes. Here we consider the solution of the combined advection–diffusion problem with a source term which relates to the long-range transport of a passive species from an elevated point source. The background concentration of NO is 7.5×10^{10} molecules cm^{-3} . The horizontal dimensions of the domain are 96 and 48 km along the x and y -axis, respectively. The vertical height of the domain is 3 km. We consider a point source at (6, 24, 0.24) km location with an NO emission rate of 1.98×10^{24} molecules s^{-1} . For simplicity, we consider constant wind direction along the x -axis.

We consider three different wind velocity and vertical diffusion profiles which are representative of stable, neutral and unstable boundary layers. The corresponding velocities and vertical diffusions are shown in Figs. 3 and 4 (Seinfeld, 1986). The horizontal diffusion coefficients K_x and K_y are kept constant and equal to $50 \text{ m}^2 \text{ s}^{-1}$. The initial tetrahedral mesh is generated by dividing the whole region into cuboids and then subdividing a cuboid into six tetrahedral elements. The cuboids are 4 and 4 km along the x and y -axis, respectively. The vertical height is divided into nine layers and the layers are placed at 0, 0.206, 0.460, 0.767, 1.13, 1.54, 2.0, 2.45 and 3 km heights, respectively.

We compute the solutions on the adaptive grid and also check the accuracy against a reference solution. The reference solution is obtained on a fixed grid generated from the base mesh by refining all the edges (to level 3) which lie inside a box lying along the x -axis through the source. We also compute the solution on a telescopic grid with refinement around the source and compare the solution with the adaptive and reference solution. The vertical turbulent diffusivity coefficient is small for the stable boundary layer. Thus the concentration does not mix much above the source height. The height of the

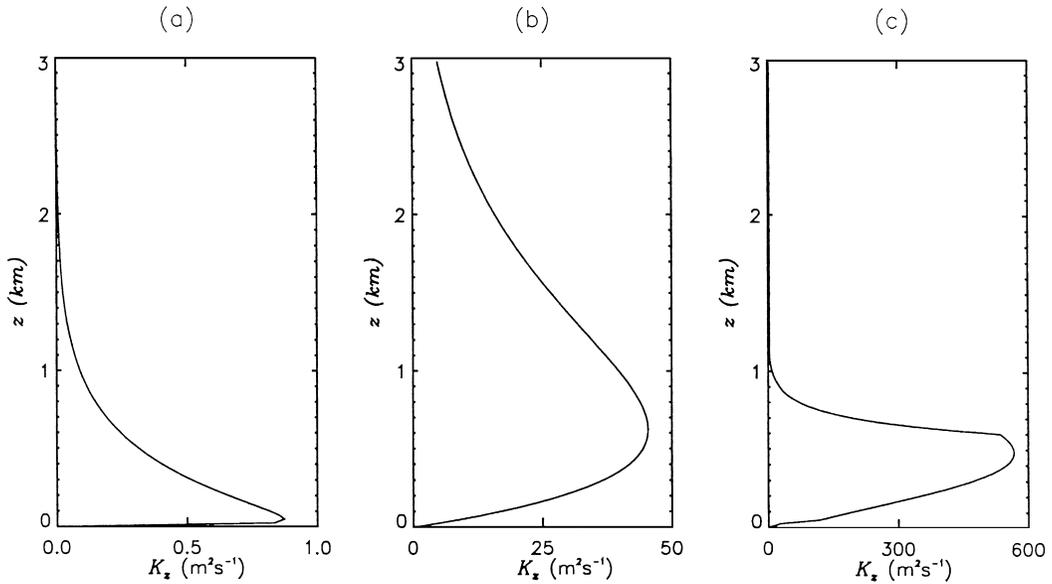


Fig. 4. A representative variation of vertical diffusion with height for (a) stable, (b) neutral and (c) unstable boundary layers.

reference box is 1/2 km and the width is 10 km for the stable boundary layer. On the other hand, the pollutant becomes well mixed above the source height for the neutral and unstable boundary layers. Thus a box of width 10 km and height 1 km is chosen for these conditions. The total number of nodes in the reference grid is 114,705 for the stable layer and 142,247 for the neutral and unstable boundary layers. The initial grid for the adaptive solution is generated by refining a region around the point source. The refinement region lies horizontally within 3 km of the point source and vertically within 300 m of the source. The initial number of nodes is 6442 for all three boundary layers. The number of nodes for the telescopic method remains 6442 throughout the simulation period. On the other hand, the adaptive grid is refined/derefinned as the solution advances. The time step τ for the implicit–explicit scheme is small (usually less than 1 min) due to small vertical spacings near the ground level which effect the CFL condition. Instead of carrying out the adaption after every time step (which is CPU intensive), the adaption is carried out approximately every 20 min. This prevents large amount computational effort being used to perhaps refine very few tetrahedra each time step and does not significantly affect solution accuracy.

6.1. Grid adaption

Three sets of tolerance parameters are chosen for the adaptive grid method for each boundary layer profile as described below. Let $TOLg$ be the maximum value of $tolg$ outside the source region. The refinement criteria of

the edges are

- (a) Refine edges to level 3 if $tolc > 9 \times 10^{10}$ and $tolg > 0.002 \times TOLg$.
- (b) Refine edges to level 2 if $tolc > 9 \times 10^{10}$ and $tolg > 0.00002 \times TOLg$.
- (c) Refine edges to level 1 if $tolc > 9 \times 10^{10}$ and $tolg > 0.000001 \times TOLg$.

for the stable boundary layer.

The corresponding criteria for the neutral and unstable boundary layers are

- (a) Refine edges to level 3 if $tolc > 10^{11}$ and $tolg > 0.01 \times TOLg$.
- (b) Refine edges to level 2 if $tolc > 10^{11}$ and $tolg > 0.0005 \times TOLg$.
- (c) Refine edges to level 1 if $tolc > 10^{11}$ and $tolg > 0.00005 \times TOLg$.

The total number of nodes generated by the adaptive grid method are 60,000, 51,000 and 52,000 for the stable, neutral and unstable boundary layers, respectively. The adaptive grid refinement in the vertical plane downwind along the plume centre-line is shown in Fig. 5. The concentration is confined near the ground level due to small vertical diffusion for the stable case. This produces high spatial gradients within this region and grid refinement is highest near the ground. Since the vertical diffusion for the other two cases is larger compared to the stable boundary layer, the grid refinement extends to

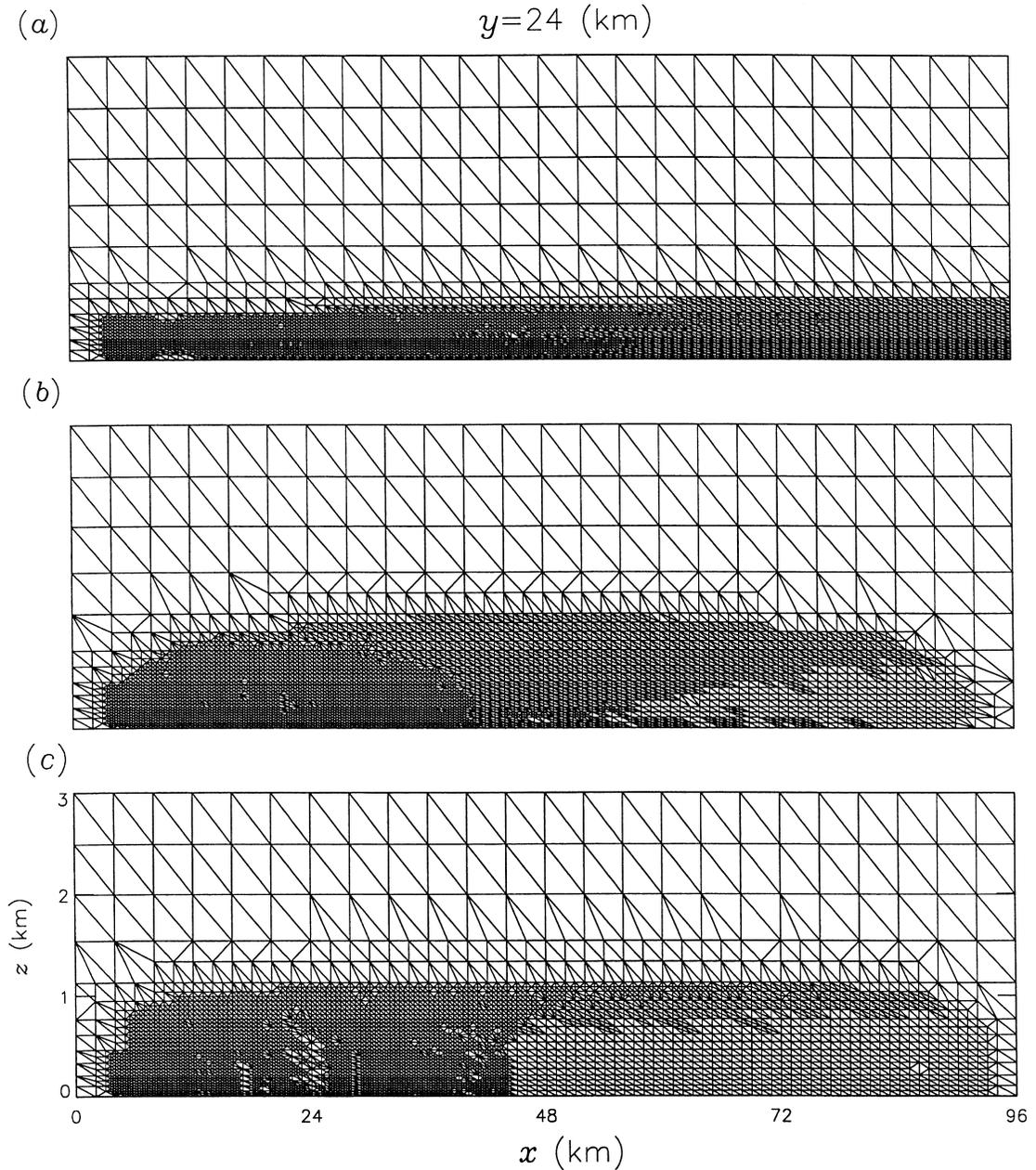


Fig. 5. Grid refinement in the vertical plane through the source along the downwind direction for the (a) stable, (b) neutral and (c) unstable boundary layers.

almost 1 km from the ground level. It is also interesting to note that at large distances downwind from the source, the adaptive technique places more mesh points at the top of the boundary layer domain. This reflects the steep gradients found here due to a significant drop in the vertical diffusion coefficient K_z . This result may have significance for models attempting to represent boundary

layer transport and mixing since the usual approach to vertical meshing is to place a greater number of mesh points close to the ground and not the top of the boundary layer. For the unstable boundary layer (see Fig. 5(c)), the concentration becomes uniformly mixed below the inversion layer but very little diffusion is taking place above the inversion layer. The gradient is high near the

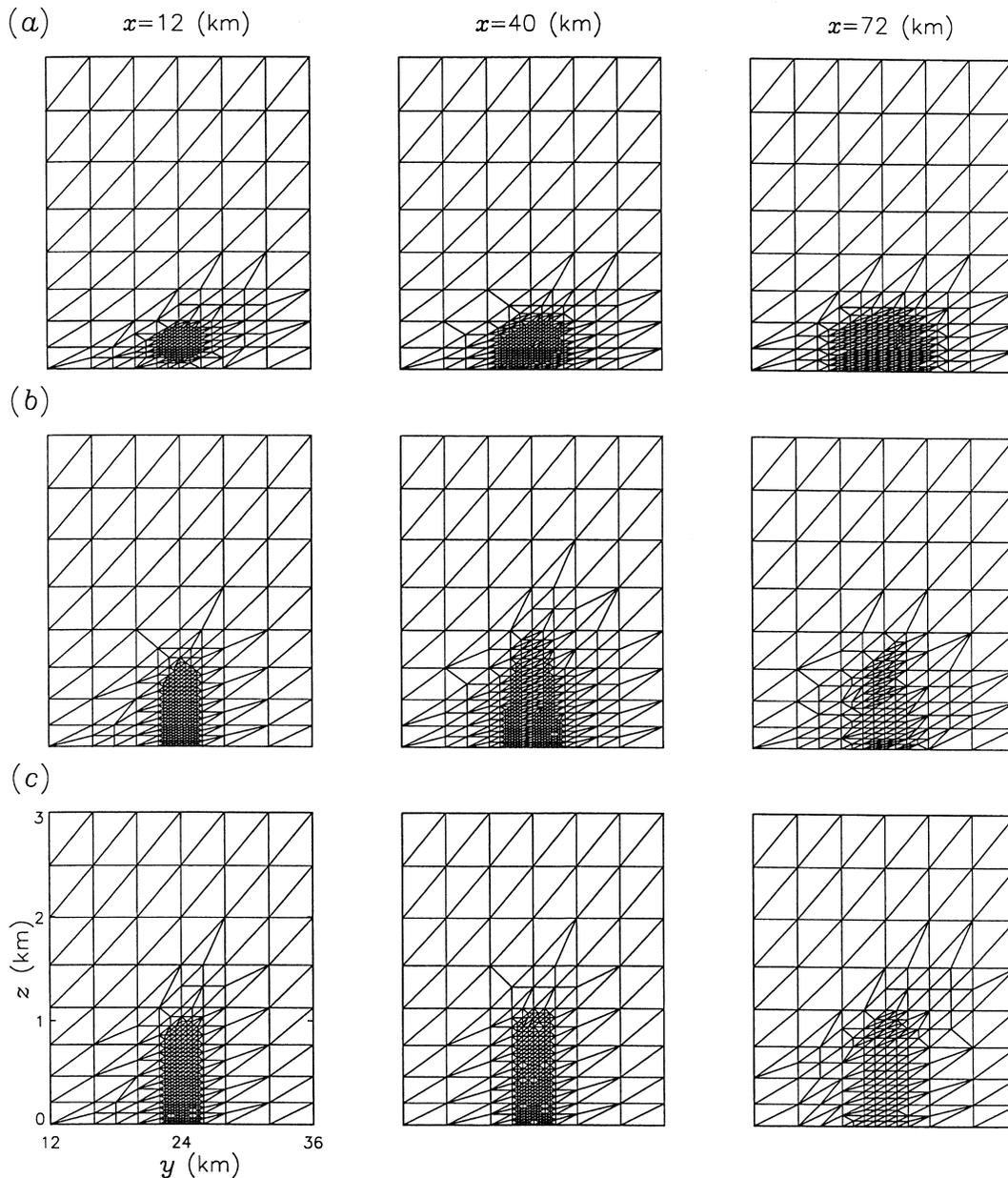


Fig. 6. Grid refinement in the vertical plane at three cross-wind directions for the (a) stable, (b) neutral and (c) unstable boundary layers.

inversion layer compared to the gradient near the ground. Thus the edges near the inversion layer refine to a higher level than the edges near ground level.

The adaptive grid refinement at three different locations in the cross-wind direction is shown in Fig. 6. The concentration gradients remain high for the stable case but low for the neutral and unstable cases far downwind from the source. Thus the edges for the stable boundary layer, far downwind the source, are refined to

higher level than for the neutral and unstable cases. The gradients are high near the source for all the three cases and the edges are refined to the maximum level for all of them.

6.2. Downwind concentration profiles

The solutions downwind along the plume centre-line in the ground level are shown in Fig. 7. The maximum

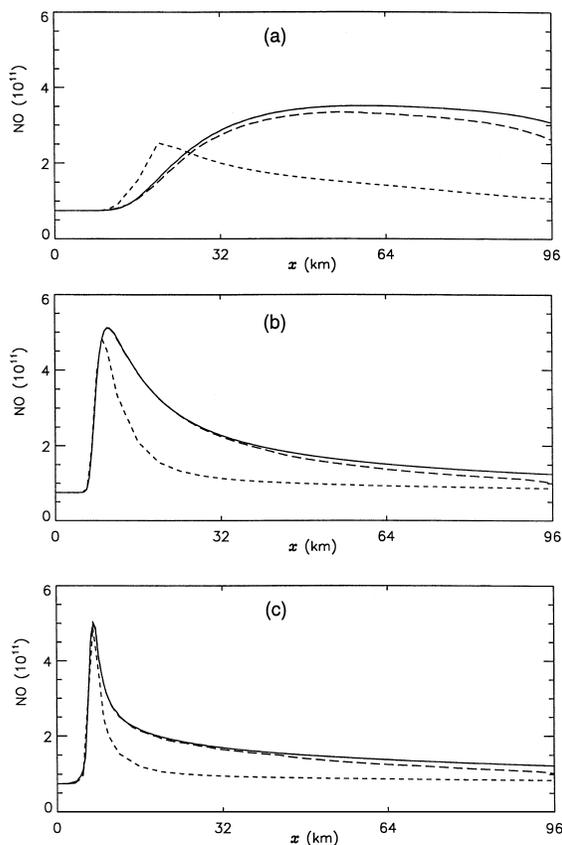


Fig. 7. Comparison of solutions (molecules cm^{-3}) along the plume centre-line in the ground level for the (a) stable, (b) neutral and (c) unstable boundary layers. The solid, dotted and dashed lines correspond to solutions in the reference, telescopic and adaptive grids, respectively.

relative errors with respect to reference solutions are 16, 20 and 20% approximately for the stable, neutral and unstable boundary layers, respectively. The maximum errors for the neutral and unstable cases occur far downwind of the source where the magnitude of the concentrations are small. The solution on the telescopic grid is accurate near the source region only, due to the refinement in this region. Far downwind from the source, the solution on the telescopic grid differs widely from the reference solution.

The programs have been run serially on an Origin2000 computer. For the stable boundary layer the times for the telescopic, adaptive and reference grids are approximately 45 min, 7 and 27 h, respectively. The corresponding times are about 1, 10 and 46 h for the neutral boundary layer. For the unstable boundary layer, these are approximately 2, 26 and 66 h, respectively. Thus the adaptive method is efficient compared to the reference solution and achieves greater accuracy than the telescopic method in a reasonable time. Note that the computing time

increases with an increase in the extent of vertical diffusion.

6.3. Cross-wind and vertical concentration profiles

We plot the concentrations along the cross-wind and vertical directions at 34 km downwind from the source in Fig. 8. The higher values of vertical diffusion, K_z , enhances vertical mixing for the neutral and unstable cases. Thus, the maximum in the vertical concentration profiles occur at the ground level for the neutral and unstable cases. Since, K_z is small for the stable boundary layer, the mixing along the vertical is relatively small. Thus the maximum in the vertical profile remains at the height of the source. For unstable conditions with an inversion layer above we would expect to see uniform concentration below the inversion layer with a very steep gradient across it. This profile is shown by both the reference and adaptive solutions (see Fig. 8(c)). The telescopic solution however enhances mixing both in the horizontal and vertical directions allowing a greater flux of pollutants through the inversion layer than determined by the vertical diffusion profile alone.

7. Discussions and conclusions

In this paper we have presented a solution method for the atmospheric diffusion equation based on an unstructured, 3D adaptive mesh. The test cases have demonstrated that adaptive methods can give much improved accuracy when compared to telescopic refinement methods particularly at large distances from the source. Adaptive refinement methods are capable of using less mesh points than using fixed refined meshes since they are able to place mesh points where the solution requires them rather than in pre-defined locations where they may not be necessary for solution accuracy. There is an extra cost with the adaptive codes, that of periodically refining/coarsening the mesh, but this cost is small if the mesh is not adapted every time-step. The results shown here demonstrate that adapting every 20 min provides solution profiles which are very close to the fully refined reference solution. The CPU times are very much lower for the adaptive mesh.

Some care must be taken when using emission data with the adaptive grid. Elevated sources (such as chimneys) are treated as point sources and cause no major difficulties as they are easily distributed to the nearest nodes. If emissions from ground level are distributed uniformly then we calculate the flux contribution from the emissions by multiplying the ground level area of a control volume by the emission rate. On the other hand, if the emissions are nonuniform, the following procedure is adopted. First, the ground level is divided into areas with uniform emissions in each of them. The

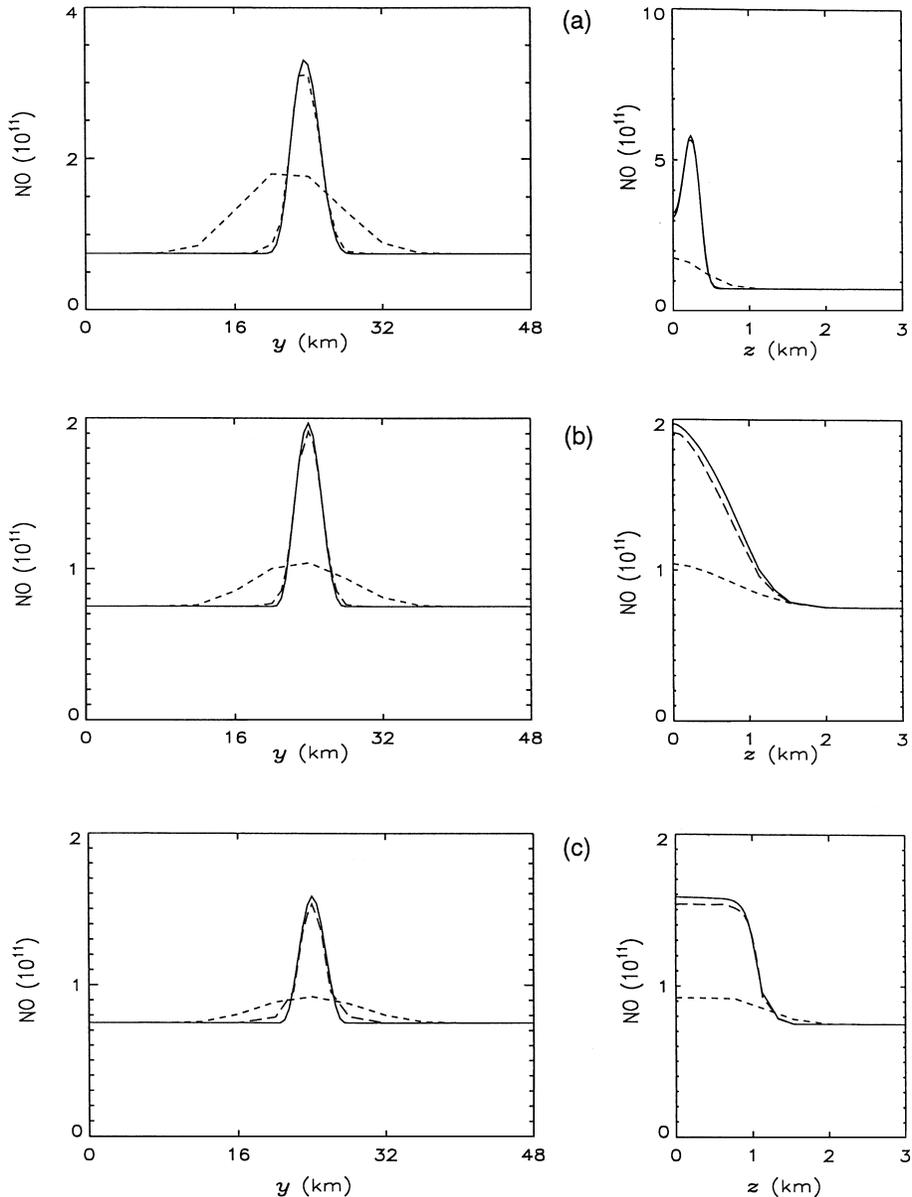


Fig. 8. Comparison of solutions (molecules cm^{-3}) along the cross-wind direction in the ground level and along the vertical in the plane through the source for the (a) stable, (b) neutral and (c) unstable boundary layers. The solid, dotted and dashed lines correspond to solutions in the reference, telescopic and adaptive grids, respectively.

proportion of each of these areas which overlaps a given control volume on the unstructured mesh is then determined. The flux contribution can then be calculated in a straightforward way.

The results here for 3D problems confirm findings in 2D that telescopic methods which use a priori refinement provide artificial diffusion away from emission sources which smears the plume profiles. For regional scale dis-

persion modelling where meteorological conditions vary with time, telescopic methods are not capable of following plume boundaries. Further, the test cases used here have demonstrated some important consequences of vertical mesh resolution for boundary layer pollutant dispersion.

It is usual in tropospheric dispersion models to stretch the mesh in the vertical domain and place more solution

points near to the ground. Close to ground level sources this often makes sense since it gives a better resolution of the initial stages of vertical mixing. It also corresponds to meshes commonly used in meteorological codes and better represents deposition to the ground. However, at large distance from their sources pollutants can become well mixed close to the ground and the important feature is their escape from the boundary layer to higher levels of the troposphere. For neutral-to-unstable conditions vertical diffusion profiles can vary dramatically with height, with large gradients in pollution profiles occurring across the inversion layer. The results here demonstrate that under such conditions solution accuracy requires refined meshes not close to the ground but close to the inversion height where steep gradients can occur. The use of coarse meshes in this region could have a significant effect on the prediction of pollutants mixing out of the boundary layer for these conditions and may be a source of error in regional scale pollution dispersion models. In a realistic boundary layer model vertical mixing profiles will change during the diurnal cycle making the a priori choice of vertical mesh structure difficult. Adaptive refinement would seem to be the simplest method for resolving such phenomena since the choice of mesh is made naturally according to the solution structure resulting from different stability conditions. The issue of using unstructured meshes which do not match those of meteorological drivers can be solved since errors due to the interpolation of wind fields are small and are outweighed by improvements in solution accuracy. Results for 3D models seem therefore to confirm those in 2D and the ability to use adaptive vertical mesh structure would seem to be important for the resolution of boundary layer pollutants.

Acknowledgements

This research has been supported with funding from the NERC. The calculations have been carried out on an Origin2000 machine with support from EPSRC through a JREI grant.

References

- Berzins, M., Fairlie, R., Pennington, S.V., Ware, J.M., Scales, L.E., 1998. SPRINT2D: Adaptive Software for PDEs. *ACM Transactions on Mathematical Software* 24, 475–499.
- Barth, T.J., Jespersen, D.C., 1989. The design and application of upwind schemes on unstructured meshes. *AIAA-89-0366*, 9–12.
- Biswas, R., Strawn, R.C., 1994. A new procedure for dynamic adaptation of 3-D unstructured grids. *Applied Numerical Mathematics* 13, 437–452.
- Hart, G., Tomlin, A.S., Smith, J., Berzins, M., 1998. Multi-scale atmospheric dispersion modelling by use of adaptive gridding techniques. *Environmental Monitoring and Assessment* 52, 225–238.
- Jacobs, H.J., Feldman, H., Kass, H., Messesheimer, M., 1995. The use of nested models for air pollution studies: an application of the EURAD model to a SANA episode. *Journal of Applied Meteorology* 34, 1301–1319.
- Mathur, R., Peters, L.K., 1990. Adjustment of wind fields for application in air pollution modelling. *Atmospheric Environment* 24A, 1095–1106.
- Rajaona, T., Ciccoli, M.C., Coppalle, A., 1998. Local refinement around a pollution source using a domain decomposition method. *Proceedings of the International Conference on Air Pollution Modelling and Simulation*, Paris, France.
- Seinfeld, J.H., 1986. *Air Pollution*. Wiley, New York.
- Speares, W., Berzins, M., 1997. A 3D unstructured mesh adaptation algorithm for time-dependent shock-dominated problems. *International Journal for Numerical Methods in Fluids* 25, 81–104.
- Talat, O., 1997. A quantitative analysis of numerical diffusion introduced by advection algorithms in air quality models. *Atmospheric Environment* 31, 1933–1940.
- Tomlin, A.S., Berzins, M., Ware, J., Smith, J., Pilling, M.J., 1997. On the use of adaptive gridding methods for modelling chemical transport from multi-scale sources. *Atmospheric Environment* 31, 2945–2959.
- Tomlin, A.S., Ghorai, S., Hart, G., Berzins, M., 1999. The use of 3-D adaptive unstructured meshes in pollution modelling. In: Zlatev, Z., et al. (Eds.), *Large-Scale Computations in Air Pollution Modelling*. Kluwer, Dordrecht.
- Wierse, M., 1997. A new theoretically motivated higher order upwind scheme on unstructured grids of simplices. *Advances in Computational Mathematics* 7, 303–335.