Unstructured Adaptive Mesh MOL Solvers for Atmospheric Reacting Flow Problems

M.Berzins, A.S.Tomlin, S.Ghorai, I.Ahmad, and J.Ware

1 Introduction

In this chapter the method of lines is applied to computational models of reacting flow arising from atmospheric applications. These computational models describe the chemical transformations and transport of species in the troposphere and have an essential role in understanding the complex processes which lead to the formation of pollutants such as greenhouse gases, acid rain and photochemical oxidants. In order to make good comparisons with the limited experimental data available it is important to have a high degree of computational resolution, but at the same time to model emissions from many different sources and over large physical domains. This chapter is thus concerned with how to achieve this by using the method of lines combined with spatial mesh adaptation techniques.

Achieving high resolution in air pollution models is a difficult challenge because of the large number of species present in the atmosphere. The number of chemical rate equations which need to be solved rises with the number of species, and for high resolution 3-dimensional calculations, detailed chemical schemes can become prohibitively large. The range of reaction time-scales often leads to stiff systems of differential equations which require more expensive implicit numerical solvers. Previous work has shown (Talat, [31], Tomlin et al., [32, 33, 12] Hart et al.,[13]) 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. A review paper by Peters et al. [22] highlights the importance of developing more efficient grid systems for the next generation of air pollution models in order to "capture important smaller scale atmospheric phenomena".

In general 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., [15], Rajaona et al.,[24], Sunderam et al., [30], Sillman et al.,[26]). 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.[32]) 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. The primary reason is that away from

concentrated sources such models use large grids of up to 50 kilometres. Since dispersion can carry species distances of hundreds of kilometres from the source, such predescribed telescopic gridding models could still lead to inaccurate downwind profiles as the plumes travel into those areas with larger grids. This is a particular problem when modelling species such as ozone, where the chemical time-scale of pollutant formation is such that the main pollution episodes occur at very long distances downwind of the sources of photochemical precursors. The regions of steep spatial gradients of species such as ozone will move with time according to the wind-field present and the spatial distribution of emissions. A reliable solution can only be obtained if the mesh can be refined accordingly. The fine scale grids used in present regional scale models are of the order of 10-20km. For a power plant plume with a width of approximately 20km, it is impossible to resolve the fine structure within the plume using grids of this size. Furthermore, to refine the mesh a priori, according to the path of the plume, would be an impossible task since the plume position is a complicated function of many factors, including reaction, deposition and transport. There is a need for the application of methods which can refine the grid according to where the solution requires it i.e. time-dependent adaptive algorithms. While there have been some applications of adaptive grids for environmental modelling, e.g Skamarock et al.[27], as yet these methods have not been implemented in standard air quality models.

This chapter is based on the work done by the authors in applying adaptive gridding techniques, which automatically refine the mesh in regions of high spatial error, and illustrates the benefits this can bring over the telescopic approach in which mesh refinement is only used close to a pollution source. The first part of this chapter (Sections 2 to 4) describes the algorithms used and present results for the 1D hyperbolic conservation law with a nonlinear source term, of Leveque and Yee [18]. This deceptively simple problem may be used to show that spurious numerical solution phenomena, such as incorrect wave speeds may occur when insufficient spatial and temporal resolution are used. Sections 5 to 10 of the chapter will provide a summary of the results for more complex two dimensional atmospheric problems (see [32]) while three dimensional problems (see [33, 12]) are considered in Sections 10 to 14. The general approach used here is to employ positivity-preserving spatial discretization schemes in the method of lines to reduce the PDE to a system of ordinary differential equations (ODEs) in time. For reacting flow problems the numerical results will show that spatial mesh points should be chosen with great care to reflect the true solution of the PDE and to avoid generating significant but spurious numerical solution features. This is achieved here by using adaptive mesh algorithms, [3], to control the spatial discretisation error by refining and coarsening the mesh.

As reacting flow problems require the use of implicit methods to resolve the fast transients associated with some chemistry species, the cost of using implicit methods may be high unless great care is taken with numerical linear algebra. In the present work this is done by making use of an approach developed for atmospheric chemistry solvers, [35, 2]. This approach uses a Gauss-Seidel iteration applied to the source terms alone. The advective terms are effectively treated explicitly but without introducing a splitting error. In three dimensions because of the need to preserve positivity of the solution and to be more concerned about efficiency we have also used a more traditional operator splitting approach. In particular the overall conclusion to be drawn from the

computational evidence for one, two and three dimensional problems is that having good mesh resolution in certain parts of the solution domain is of critical importance with regard to obtaining a meaningful solution.

2 Spatial Discretisation and Time Integration

The 1D Leveque and Yee problem [18], is given by

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = -\psi(u) \quad x \in [0,\infty], \quad \psi(u) = \mu u(u-1)(u-0.5) \tag{1}$$

and is the linear advection with a source term that is "stiff" for large μ. The initial and boundary values (at $x = 0$) are defined by

$$u(x,0) = u_0(x) = u_L \; = \; 1, x \leq x_d; \; u_R \; = \; 0, \; x \; > \; x_d$$

where $x_d = 0.1$ or 0.3 in the cases considered here. The infinite domain will also be truncated to $[0,1]$ for the cases considered here, as this is sufficient to demonstrate the behaviour of the methods employed. A simple outflow boundary condition is then used at $x = 1$. The solution of equation (1) is a discontinuity moving with constant speed and has a potentially large source term that only becomes active at the discontinuity, [18].

Defining a spatial mesh $0 = x_1 < \ldots < x_N = 1$ and the vector of values $\underline{U}$ with components $U_i(t) \approx u(x_i,t)$ where $u(x,t)$ is the exact solution to the p.d.e. We define $U_i(t)$ as the exact solution to the o.d.e. system derived by spatial semi-discretization of the p.d.e. and given by

$$\dot{\underline{U}} \; = \; \underline{F}_N(t,\underline{U}(t)), \qquad \underline{U}(0) \;\; \textit{given}. \tag{2}$$

This true solution $[\underline{U}(t_n)]_{n=0}^{p}$ is approximated by $[\underline{V}(t_n)]_{n=0}^{p}$ at set of discrete time $0 \; = \; t_0 < t_1 < \ldots < t_p = t_e$ by the time integrator. The form of the o.d.e. system given by equation (2) at time t is given by

$$\underline{F}_N(t_n,\underline{U}(t_n)) = \underline{F}_N^f(t_n,\underline{U}(t_n)) + \underline{F}_N^s(t_n,\underline{U}(t_n)). \tag{3}$$

where the superscripts f and s denote the flow and source term parts of the function $\underline{F}$ as defined below. The function $\underline{F}_N^f(t_n,\underline{U}(t_n))$ is the second-order limited discretisation of the advective terms in equation (1) whose components are given by

$$F_j^f(t,\underline{U}(t)) \; = \; -\left[1 + \frac{(B(r_j,1)}{2} - \frac{B(r_{j-1},1)}{2r_{j-1}}\right] \frac{(U_j(t) - U_{j-1}(t))}{\triangle x} \quad . \tag{4}$$

The function B is a limiter such as that of van Leer: (see [3])

$$B(r_j,1) \; = \; \frac{r_j \; + \; |\, r_j \,|}{1 \; + \; r_j}, \text{ and } r_j \; = \; \frac{U_{j+1}(t) - U_j(t)}{U_j(t) - U_{j-1}(t)}. \tag{5}$$

The vector $\underline{F}_N^s(t,\underline{U}(t))$ represents the approximate spatial integration of the source term which is defined by $\frac{1}{\triangle x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \psi(U(x,t))dx$ and is evaluated by using the midpoint quadrature rule so that its jth component is:

$$F_j^s(t,U_j(t)) \;=\; \psi(U_j(t)). \tag{6}$$

The time integration method used here (mostly for simplicity of analysis) is the Backward Euler method defined by

$$\underline{V}(t_{n+1}) \;=\; \underline{V}(t_n) \;+\; \underline{F}_N(t_{n+1},\underline{V}(t_{n+1})). \tag{7}$$

In the case when a modified Newton method is used to solve the nonlinear equations at each timestep, this constitutes the major computational task of a method of lines calculation. In cases where large o.d.e. systems result from the discretization of flow problems with many chemical species the c.p.u. times may be excessive unless special iterative methods are used.

The approach of [4] is used to neglect the advective terms $J_f = \frac{\partial F^f}{\partial \underline{V}}$, and to concentrate on the Jacobian of the source terms $J_s \;=\; \frac{\partial F^s}{\partial \underline{V}}$ when forming the Newton iteration matrix. This approach, in the case when no source terms are present, corresponds to using functional iteration for the advective calculation, see [2, 4]. The matrix $I - \triangle t\gamma J_s$ is the Newton iteration matrix of that part of the o.d.e. system corresponding to the discretization of the time derivatives and the source terms alone. This matrix is thus block-diagonal with as many blocks as there are spatial elements and with each block having as many rows and columns as there are p.d.e.s. The fact that a single block relates only to the chemistry within one cell means that each block's equations may be solved independently by using a Gauss-Seidel iteration. This approach has been used with great success for atmospheric chemistry problems, [35]. The nonlinear equations iteration employed here may thus be written as

$$[I - \triangle t\; J_s]\;\; [\underline{V}^{m+1}(t_{n+1}) - \underline{V}^m(t_{n+1})] = \underline{r}\left(t_{n+1}^m\right) \tag{8}$$

where $\underline{r}\left(t_{n+1}^m\right) \;=\; -\underline{V}^m(t_{n+1}) + \underline{V}(t_n) + \triangle t\underline{F}_N(t_{n+1},\underline{V}^m(t_{n+1}))$. Providing that the iteration converges, this approximation has no adverse impact on accuracy. In order for this iteration to converge with a rate of convergence r_c it is necessary, [2], that

$$||\; [I - \triangle t J_s]^{-1}\; \triangle t\; J_f \;|| \;\; = \; r_c \qquad \text{where } \; r_c < 1. \tag{9}$$

Using the identity $\| ab \| \leq \| a \| \| b \|$, and defining J_f^* as $J_f^* = (\triangle x) J_f$ gives:

$$\frac{\triangle t}{\triangle x} \| \; J_f^* \; \| \;\; \leq \; r_c \; || \; [I - \triangle t J_s] \; || \; . \tag{10}$$

Hence the convergence restriction may be interpreteted as a CFL type condition. For example in the case of the p.d.e. in (1), $[I - \triangle t J_s]$ is a diagonal matrix with entries $1 + \triangle t\mu \; \frac{\partial \psi}{\partial V}$ where

$$\frac{\partial \psi}{\partial V} = p(V) \tag{11}$$

and where $p(V) = 3V^2 - 3V + 0.5$ gives a CFL type condition that allows larger timesteps as μ increases. The function $p(V)$ is bounded between the values 0.5 and -0.25 for solution values in the range $[0,1]$.

3 Space-Time Error Balancing Control.

Hyperbolic p.d.e.s are often solved by using a CFL condition to select the timestep. The topic of choosing a stable stepsize for such problems has been considered in detail by Berzins and Ware [6]. Although a CFL condition indicates when the underlying flow without reactions is stable, it is still necessary to get the required accuracy for the chemistry terms. In most time dependent p.d.e. codes either a CFL stability control is employed or a standard o.d.e. solver is used which controls the local error $\underline{l}_{n+1}(t_{n+1})$ with respect to a user supplied accuracy tolerance. Efficient time integration requires that the spatial and temporal errors are roughly the same order of magnitude. The need for spatial error estimates unpolluted by temporal error, requires that the spatial error is the larger of the two. One alternative approach developed by Berzins [3, 4] is to use a local error per unit step control in which the time local error (denoted by $\underline{le}(t)$) is controlled so as to be smaller than the local growth in the spatial error over the timestep (denoted by $\underline{est}(t)$). In the case of the Backward Euler method the standard local error estimate at t_{n+1} is defined as $\underline{le}(t_{n+1})$ and is estimated in standard ODE codes by

$$\begin{aligned} \underline{le}(t_{n+1}) &= \frac{\triangle t}{2}\left[\underline{F}_N(t_{n+1},\underline{V}(t_{n+1})) - \underline{F}_N(t_n,\underline{V}(t_n)\right] . \\ &\approx \frac{\triangle t^2}{2}\ddot{\underline{V}}(t_{n+1}) \end{aligned} \tag{12}$$

where the function $\underline{F}$ is defined by equation (2). The error control of [3] is defined by

$$\| \underline{le}_{n+1}(t_{n+1}) \| \leq \varepsilon \| \underline{est}(t_{n+1}) \| \tag{13}$$

where $0 < \varepsilon < 1$ is a balancing factor and $\underline{est}(t_{n+1})$ represents the local growth in time of the spatial discretisation error from t_n to t_{n+1}, assuming that the error is zero at t_n. Once the primary solution has been computed using the method of Section 2, a secondary solution is estimated at same time step with an upwind scheme of different order and a different quadrature rule for source term integration. The difference of these two computed solution is then taken as an estimate of the local growth in time of the spatial discretization error in the same way as in [3]. The primary solution $\underline{V}(t_{n+1})$ starting from $\underline{V}(t_n)$ is computed in the standard way as described in Section 2. The secondary solution $\underline{W}(t_{n+1})$ is computed by solving

$$\dot{\underline{W}}(t) = \underline{G}^f(t,\underline{W}(t)) + \underline{G}^s(t,\underline{W}(t)), \quad \underline{W}(t_n) = \underline{V}(t_n). \tag{14}$$

with initial value $\underline{V}_n$, where $\underline{G}^f$ and $\underline{G}^s$ are the first order advective term and the source terms which are evaluated using a linear approximation on each interval and the trapeziodal rule i.e.

$$G^f_j(t,W_j(t)) = -\frac{(W_j(t) - W_{j-1}(t)}{\triangle x},$$

$$G_j^s(t,W_j(t)) \quad = \quad \frac{1}{4}(\psi(W_{j-1}(t)) + 2\psi(W_j(t)) + \psi(W_{j+1}(t)). \tag{15}$$

Estimating $\underline{es}(t_{n+1})$ by applying the Backward Euler Method to (14) subtracted from (7) with one iteration of the modified Newton iteration of the previous section, as in [4], gives

$$\begin{aligned}[I - \triangle t J_s][\underline{es}(t_{n+1})] \quad &= \quad \triangle t\ [\underline{F}^f(t_{n+1},\underline{V}(t_{n+1})) - \underline{G}^f(t_{n+1},\underline{V}(t_{n+1})) \\ &+ \quad \underline{F}^s(t_{n+1},\underline{V}(t_{n+1})) - \underline{G}^s(t_{n+1},\underline{V}(t_{n+1}))\ ]\end{aligned} \tag{16}$$

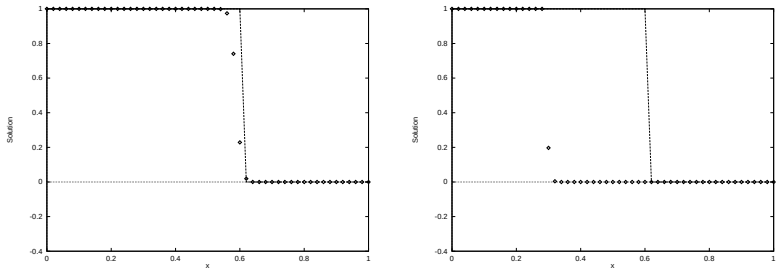
where $\underline{es}(t_{n+1}) \approx \underline{V}(t_{n+1}) - \underline{W}(t_{n+1})$.

4 Fixed and Adaptive Mesh Solutions

In the case of the problem defined by equation (1) comparisons were made between the standard local error control approach in which absolute and relative tolerances RTOL and ATOL are defined, (see Pennington and Berzins, [21]), and the new approach defined by (13). The choice of the parameter ε is an important factor in the performance of the second approach. In selecting this parameter the local growth in the spatial discretization error should dominate the temporal error and the work needed should not be excessive. Obviously the larger the value of ε the fewer ODE time steps there will be, and the smaller the value of ε the more steps there will be. A good compromise between efficiency and accuracy is to use ε in the range 0.1 to 0.3. The numerical experiments described by Ahmad [1] confirm the results of Berzins [3], although it is noted that for some combustion problems, ε may have to be reduced to below 0.1.

An important feature of solving the problem defined by equation (1) is that the numerical solution may move with an incorrect wave speed. Leveque and Yee [18] showed that the step size and the mesh size should be $O(\frac{1}{\mu})$, to avoid spurious solutions being generated. In order to illustrate these results we have taken $x_d = 0.3$ in equation (1), $\triangle x$ =0.02 and used a fixed time step $\triangle t$= 0.015. The product of time step $\triangle t$ and the reaction rate μ determines the stiffness of the system. Figure(1) shows the comparison of the computed solution and exact solution at $t = 0.3$ for μ =100, and 1000 ($\triangle t\mu$= 1.5 and 15) respectively. It is evident from Figure (1) that for smaller $\triangle t\mu$ the strategy works well and good results are obtained. When $\triangle t\mu = 15$, the discontinuity has stopped at x = 0.3 and when a trapezoidal quadrature rule was used for the source term, a large undershoot and overshoot occurred in the numerical solution. Leveque and Yees [18] pointed out that the source of difficulty is the discontinuity in the solution and that a much finer grid is needed there and suggested deploying a method that is capable of increasing the spatial resolution near the discontinuity rather than excessive refinement of the overall grid.

For this purpose a monitor function was used here to guide the decision as to where to refine or coarsen the mesh. A commonly used monitor function is the second spatial derivative which however tends to infinity around a shock [21] as the mesh is refined. In order to overcome this we have introduced a new monitor function based upon the local growth in time spatial error $\underline{est}$ as defined by equation (13). This leads to the use of local grid refinement, and with the help of the error balancing approach described in



Solution with $\triangle t\mu$ = 1.5 Solution with $\triangle t\mu$ = 15

Figure 1: Comparison between true solution(line) and numerical solution(dots) using local error control with 0.01 relative tolerance and 1×10^{-5} absolute tolerance.

Section 3 it is possible to create a new refined grid directly surrounding the location of the source. For this purpose we have modified the approach described by Pennington and Berzins [21]. The remesh routine bisects the mesh cell if the monitor function is too large or combines two cells into one if the monitor function is well below the required value. In the experiments here the remeshing routine is called on every second time step. The adaptive mesh initially starts with 26 points and when the error was larger that specified limit then the corresponding cell is subdivided into two with a 75 maximum points being allowed for the case shown in Figure 2, which shows the front moving correctly. The conclusion from these experiments is that for problems combining reaction type terms and advection operators the use of adaptive mesh techniques within a method of lines framework may be a critical factor is ensuring that a good numerical solution is obtained. The remainder of this chapter will show that this conclusion also applies to atmospheric modelling problems in two and three space dimensions.

5 Atmospheric Modelling Problem

In order to illustrate the application of the method of lines to atmospheric modelling problems, the model problem considered here involves the interaction of a power plant plume with background emissions. Such a power plant plume is a highly concentrated source of NOx (NO and NO_2) emissions which can be carried through the atmosphere for hundreds of kilometres, and so provides a stringent test of whether adaptive gridding methods can lead to more reliable results for complex multi-scale models. The test conducted here involves considering the interaction of the plume with its surroundings, and in the model we look at background scenarios of both clean and polluted air (Tomlin et al, [32]). The test case model covers a region of 300 x 500 km. To keep the model simple, and therefore reveal particular issues related to the mesh, we have used a reduced chemical scheme with idealised dispersion conditions. The domain is approximated by an unstructured triangular mesh in two space dimensions and by a tetrahedral mesh in three space dimensions. In both cases the mesh can then be adapted to higher and higher levels of refinement according to errors in solution components. The so-

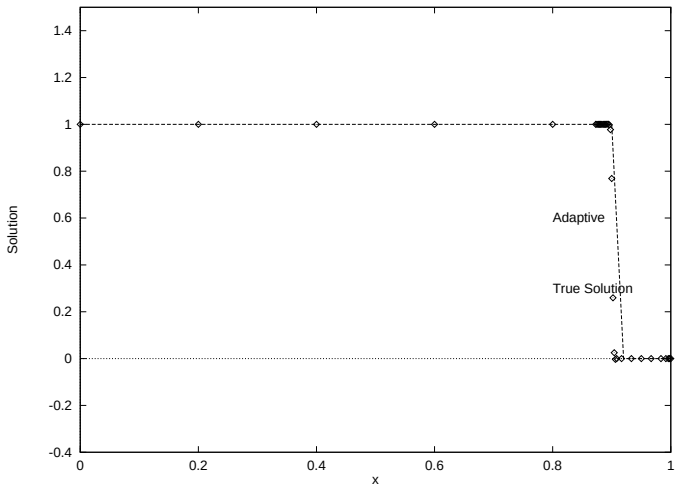


Figure 2: True Solution (lines) vs Adaptive Mesh Solution (dots), t=0.6.

lution technique is based on the spatial discretisation of a set of advection/diffusion equations on the unstructured mesh using a finite volume, flux-limited scheme.

The atmospheric diffusion equation in three space dimensions is given by:

$$\frac{\partial c_s}{\partial t} = -\frac{\partial (u c_s)}{\partial x} - \frac{\partial (w c_s)}{\partial y} - \frac{\partial (v c_s)}{\partial z} + \frac{\partial}{\partial x}\left(K_x \frac{\partial c_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial c_s}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial c_s}{\partial z}\right) + R_s(c_1, c_2, \ldots, c_q) + E_s - (\kappa_{1s} + \kappa_{2s}) c_s, \qquad (17)$$

where c_s is the concentration of the s'th compound, u,w, are horizontal wind velocities, K_x and K_y are turbulent diffusivity coefficients and κ_{1s} and κ_{2s} are dry and wet deposition velocities respectively. E_s describes the distribution of emission sources for the s'th compound and R_s is the chemical reaction term which may contain nonlinear terms in c_s. For *npde* chemical species an *npde*-dimensional set of p.d.e.s is formed describing the rates of change of species concentration over time and space, where each is coupled through the nonlinear chemical reaction terms.

In the first instance the restriction to two space dimensions has the advantage that it is possible to concentrate on showing that standard adaptive numerical methods have the potential to reveal detail not previously observed in plume models. The extension to three dimensions will then show that the same conclusions can be drawn but that there are additional benefits from using mesh refinement vertically.

The simplified chemical mechanism used is shown in Table 1 of Tomlin et al. [32] and contains only 10 species. Despite its simplicity it represents the main features of a tropospheric mechanism, namely the competition of the fast equilibrating inorganic reactions:

$$\begin{aligned} & & \mathrm{O_2} & \\ \mathrm{NO_2} + \mathrm{h}\nu & \rightarrow & \mathrm{O_3} + \mathrm{NO} & \\ & & & \\ \mathrm{NO} + \mathrm{O_3} & \rightarrow & \mathrm{NO_2} + \mathrm{O_2}, & \end{aligned}$$

with the chemistry of volatile organic compounds (voc's), which occurs on a much slower time-scale. This separation in time-scales generates stiffness in the resulting equations. The voc reactions are represented by reactions of a single species, formaldehyde. This is unrealistic in terms of the actual emissions generated in the environment, but the investigation of fully speciated voc's is not the purpose of the present study. We therefore wished to include the minimum number of reactions which would lead to the generation of ozone at large distances from the NOx source. Deposition processes have not been included in the first instance.

In the work of Tomlin et al [32] the model was used to represent three separate scenarios of a plume of concentrated NOx emissions being dispersed through a background of clean and polluted air. Only one set of these results are shown here. This case represents a clean air situation where the background levels for NOx and voc's are low. Initial conditions for background concentrations are NO_2: 1.00×10^8 (molecule cm^{-3}), NO : 1.00×10^8 (molecule cm^{-3}), O_3: 5.00×10^{11} (molecule cm^{-3}), HCHO : 1.00×10^{10} (molecule cm^{-3}). Concentrations in the background change diurnally as the chemical transformations take place according to photolysis rates, temperature and concentration changes.

The power station was taken to be a separate source of NOx and this source was represented in a slightly different way. In this case the chimney region is treated as a subdomain and the concentration in the chimney set as an internal boundary condition. In terms of the mesh generation this ensures that the initial grid will contain more elements close to the concentrated emission source. This is similar in methodology to the telescopic approach. The concentration in the chimney corresponds to an emission rate of NOx of 400kg hr^{-1}. We have considered only 10% of the NOx to be emitted as NO_2.

A constant wind speed of $5ms^{-1}$ in the x-direction was used and the eddy diffusion parameters K_x and K_y was set at 300 m^2s^{-1} for all species.

6 Triangular Finite Volume Space Discretization Method.

The basis of the numerical method is the spatial discretisation of the p.d.e.s in equation (17) on unstructured triangular meshes as used in the software SPRINT2D (Berzins et al. [7]). The Method of Lines approach then leads to a system of o.d.e.s in time can then be solved as an initial value problem, and a variety of powerful software tools exist for this purpose (Berzins et al, [5]). For advection dominated problems it is important to choose a discretisation scheme which preserves the physical range of the solution.

Unstructured triangular meshes are popular with finite volume/element practitioners because of their ability to deal with general two-dimensional geometries. In terms of application to multi-scale atmospheric problems, we are not dealing with complex physical geometries, but unstructured meshes provide a good method of resolving

the complex structures formed by the interaction of chemistry and flow in the atmosphere and by the varying types of emission sources. The term unstructured represents the fact that each node in the mesh may be surrounded by any number of triangles whereas in a structured mesh this number would be fixed. The discretisation of advection/diffusion/reaction equations on unstructured meshes will now be discussed.

For systems of equations such as (17) it is useful to consider the advective and diffusive fluxes separately in terms of the discretisation. In the present work, a flux limited, cell-centered, finite volume discretization scheme of Berzins and Ware [6, 4] was chosen. This method enables accurate solutions to be determined for both smooth and discontinuous flows by making use of the local Riemann solver flux techniques (originally developed for the Euler equations) for the advective parts of the fluxes, and centered schemes for the diffusive part. The scheme used for the treatment of the advective terms is an extension to irregular triangular meshes of the nonlinear scheme described by Spekreijse [29] for regular Cartesian meshes. The scheme of Berzins and Ware has the desirable properties (see Chock [11]) of preserving positivity, eliminating spurious oscillations and restricting the amount of diffusion by the use of a nonlinear limiter function. Recent surveys of methods for the advection equation ([34], [36]) have suggested the use of a very similar scheme to Spekreijse for regular Cartesian meshes, preferring it to schemes such as Flux Corrected Transport.

To illustrate this method, consider the advection-reaction equation that extends equation (1) to two space dimensions:

$$\frac{\partial c}{\partial t} = -\frac{\partial uc}{\partial x} - \frac{\partial wc}{\partial y} + R(c) \ \ ,t \in (0,t_e) \ , \ (x,y) \in \Omega \tag{18}$$

with appropriate boundary and initial conditions. A finite volume type approach is adopted in which the solution value at the centroid of triangle i, (x_i,y_i), is c_i and the solutions at the centroids of the triangles surrounding triangle i are c_l, c_j and c_k. Integration of equation (18) on the ith triangle, which has area A_i, use of the divergence theorem, and the evaluation of the line integral along each edge by the midpoint quadrature rule gives an o.d.e. in time:

$$\begin{aligned}\frac{dc_i}{dt} = -\frac{1}{A_i}(&uc_{ik}\Delta y_{0,1} - vc_{ik}\Delta x_{0,1} + uc_{ij}\Delta y_{1,2} \\ &-vc_{ij}\Delta x_{1,2} + uc_{il}\Delta y_{2,0} - vc_{il}\Delta x_{2,0}) + R(c_i),\end{aligned} \tag{19}$$

where $\Delta x_{ij} = x_j - x_i$, $\Delta y_{ij} = y_j - y_i$. The fluxes uc_{ij} and vc_{ij} in the x and y directions respectively are evaluated at the midpoint of the triangle edge separating the triangles associated with c_i and c_j. These fluxes are evaluated by taking account of the flow directions with respect to the orientation of the triangle. This is achieved by using either the *left* or *right* solution values depending on the direction of advection and how each edge is aligned. These *left* and *right* solution values for each edge in a triangle are defined as the *left* solution value being that internal to the ith triangle, and the *right* solution value being that external to triangle i. Consider for example the case shown in Figure (3) when u is positive and $x_i < x_j$. This means that the x component of the advection is flowing from node i to node j, and so $c_{ij} = c^l_{ij}$. Similarly when v is positive

the y component of the wind is blowing from node k to node i and so $c_{ik} = c^r_{ik}$. Hence, equation (19) may be written as

$$\frac{dc_i}{dt} = -\frac{1}{A_i}(uc^l_{ik}\Delta y_{0,1} - vc^r_{ik}\Delta x_{0,1} + uc^l_{ij}\Delta y_{1,2} \\ -vc^l_{ij}\Delta x_{1,2} + uc^r_{il}\Delta y_{2,0} - vc^l_{il}\Delta x_{2,0}) + R(c_i), \tag{20}$$

A simple first-order scheme uses $c^l_{ij} = c_i$, $c^r_{ij} = c_j$ on the edge between triangles i and j. This scheme is too diffusive and so Berzins and Ware [6] use a complex interpolation scheme to obtain the *left* and *right* values on each edge. The interpolants in this second order scheme, use a constrained or limited form of the solution obtained from the six triangles surrounding an edge giving a ten triangle stencil for the discretization of the convective terms on each triangle. For example, the value c^l_{ij} is constructed by forming

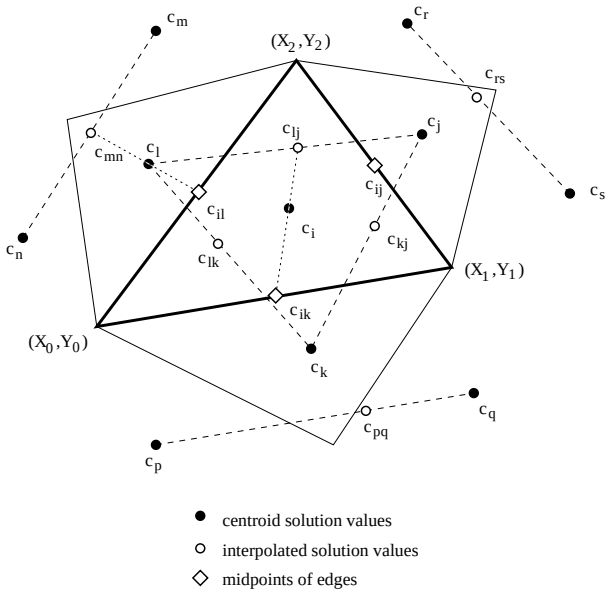


Figure 3: Interpolants used in irregular mesh flux calculation.

a linear interpolant using the solution values c_i, c_k and c_l at the three centroids. An alternative interpretation is that linear extrapolation is being used based on the solution value c_i and an intermediate solution value (itself calculated by linear interpolation) c_{lk} which lies on the line joining the centroids at which c_l and c_k are defined (see Figure 3) i.e.

$$c^l_{ij} = c_i + \Phi(S_{ij})\; d_{ij,i}\; \frac{c_i - c_{lk}}{d_{i,lk}}, \tag{21}$$

where the argument S is a ratio of solution gradients defined in a similar way to the ratio r_j in equation (5), see [6] and the generic term $d_{a,b}$ denotes the positive distance

between points a and b. For example $d_{ij,i}$ denotes the positive distance between points ij and i, see Figure (3), as defined by

$$d_{i,ij} = \sqrt{(x_i - x_{ij})^2 + (y_i - y_{ij})^2}\,, \tag{22}$$

where (x_{ij},y_{ij}) are the co-ordinates of c_{ij} . In order to preserve positivity in the numerical solution, the limiter function Φ is used and has to satisfy $\Phi(S)/S \leq 1$, see [6]. These conditions are satisfied, for example, by a modified van Leer limiter defined by:

$$\Phi(S) = (S + |S|)/(1 + Max(1, |S|)) \ . \tag{23}$$

The value c^r_{ij} is defined in a similar way using the centroid values c_j, c_s and c_r . This scheme is of second order accuracy, see [6]. The diffusion terms are discretised using a finite volume approach to reduce the integrals of second derivatives to the evaluation of first derivatives at the midpoints of edges. These first derivatives are then evaluated by differentiating a bilinear interpolant based on four mid-point values, see [7]. The boundary conditions are implemented by including them in the definitions of the advective and diffusive fluxes at the boundary.

7 Time Integration.

A method of lines approach with the above spatial discretization scheme results in a system of o.d.e.s in time which are integrated using the code SPRINT [5] with the Theta or BDF options which are specially designed for the solution of stiff systems with moderate accuracy and automatic control of the local error in time. Once the p.d.e.s have been discretised in space we are left with a large system of coupled o.d.e.s of dimension $N = m \times npde$ where m is the number of triangles in the mesh, and $npde$ the number of species. These equations may now be written in the same form as equation (2) as

$$\dot{\underline{c}} \ = \ \underline{F}_N \ (\ t, \ \underline{c}(t) \ ), \ \ \underline{c}(0) \ \text{ given}\,, \tag{24}$$

where, in the case of a single species, the vector, $\underline{c}(t)$, is defined by
$\underline{c}(t) \ = \ \left[c(x_1,y_1,t), \ldots, c(x_N,y_N,t)\ \right]^T$. The point x_i,y_i is the centre of the i th cell and $C_i(t)$ is defined as a numerical approximation to the exact solution to the p.d.e. evaluated at the centroid i.e. $c(x_i,y_i,t)$. The method of lines approach is used to numerically integrate equation (24) thus generating an approximation, $\underline{C}(t)$, to the vector of exact p.d.e. solution values at the mesh points, $\underline{c}(t)$.

The Theta method (Berzins and Ware, [6]), which has been used for the experiments described here, defines the numerical solution at $t_{n+1} = t_n + \triangle t$, where $\triangle t$ is the time step size, as denoted by $\underline{C}(t_{n+1})$, by:

$$\underline{C}(t_{n+1}) \ = \ \underline{C}(t_n) \ + \ (1-\theta)\triangle t\, \dot{\underline{C}}(t_n) + \theta \ \triangle t\, \underline{F}_N(t_{n+1}, \underline{C}(t_{n+1})), \tag{25}$$

in which $\underline{C}(t_n)$ and $\dot{\underline{C}}(t_n)$ are the numerical solution and its time derivative at the previous time t_n and $\theta = 0.55$. This system of equations is solved by using the approach described in Section 2. In this case the matrix J_s is block-diagonal with as many block

as there are triangles and with each block having as many rows and columns as there are p.d.e.s. The fact that the blocks relate only to the chemistry plus source/sink terms within each cell, the equations may be solved independently using LU decomposition, or even more efficiently by using Gauss Seidel iterations, see Verwer [35]. This approach may also be interpreted as approximating the flow term $[I - \triangle t \theta J_f]$ by the identity matrix, as is done when using functional iteration with the Theta method applied to flow alone (Berzins, [3]). Since the spatial discretization method connects each triangle to as many as ten others it follows that the matrix $\left[I - \triangle t \theta \, J_f\right]$ may have a much more complex sparsity pattern than that of the block-diagonal matrix $[I - \triangle t \theta \, J_s \,]$. Approximating the matrix $\left[I - \triangle t \theta \, J_f\right]$ by the identity matrix (Berzins and Ware, [6]) thus eliminates a large number of the full Jacobian entries. Moreover the use of Gauss Seidel iteration makes it possible to solve these problems without any matrices being stored. This approach is particularly useful in three space dimensional problems.

The original approach of Berzins [3] was only extended to source-term problems by Ahmad and Berzins [1]. As a consequence the calculations performed by Tomlin et al., [32], used the standard local error approach given by:

$$|| \ \underline{le}(t_{n+1}) \ || \ < \ TOL. \tag{26}$$

where $\underline{le}$ is the local error defined as in equations (12) and (13).

8 Mesh Generation and Adaptivity.

The initial unstructured meshes used in SPRINT2D are created from a geometry description using the Geompack (Joe and Simpson, [16]) mesh generator. These meshes are then refined and coarsened by the Triad adaptivity module which uses data structures to enable efficient mesh adaptation.

Since the initial mesh is unstructured we have to be very careful in choosing a data structure which provides the necessary information for refining and derefining the mesh. When using a structured mesh it is possible to number mesh vertices or elements explicitly. This is not possible for unstructured meshes and therefore the data structure must provide the necessary connectivity. The important factor is to maintain the quality of the triangle as the mesh is refined and coarsened. This is achieved using a tree-like data structure with a method of refinement based on the regular subdivision of triangles. These may later be coalesced into the parent triangle when coarsening the mesh. This process is called local h-refinement, since the nodes of the original mesh do not move and we are simply subdividing the original elements. Three examples of adaptive meshes for a single moving front at different times are shown in Figure 4. These meshes show how the adaptive mesh follows the front as it moves in time across the spatial domain. Similar procedures are extensively used with a wide range of both finite element and volume methods for a very broad range of physical problems. Once a method of refinement and derefinement has been implemented, it remains to decide on a suitable criterion for the application of the adaptivity. The ideal situation would be that the decision to refine or derefine would be made on a fully automatic basis

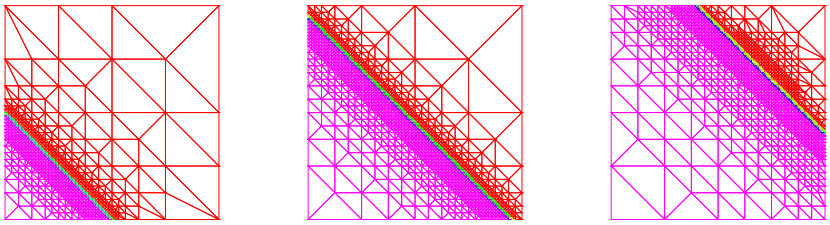


Figure 4: Sequence of Refined Meshes

with no user input necessary. In practice a combination of an automatic technique and some knowledge of the physical properties of the system is used. The technique used in this work is based on the calculation of spatial error estimates. Low and high order solutions are obtained and the difference between them gives the spatial error, as in Section 3 and in [3] but without the extension to source terms in [1]. The algorithm can then choose to refine in regions of high spatial error by comparison with a user defined tolerance. For the ith p.d.e. component on the jth triangle, a local error estimate $e_{i,j}(t)$ is calculated from the difference between the solution using a first order method and that using a second order method. For time dependent p.d.e.s this estimate shows how the spatial error grows locally over a time step. A refinement indicator for the jth triangle is defined by an average scaled error ($serr_j$) measurement over all $npde$ p.d.e.s using supplied absolute and relative tolerances:

$$serr_j = \sum_{i=1}^{npde} \frac{e_{i,j}(t)}{atol_i/A_j + rtol_i \times C_{i,j}}, \tag{27}$$

where $atol$ and $rtol$ are the absolute and relative error tolerances. This formulation for the scaled error provides a flexible way to weight the refinement towards any p.d.e. error. An integer refinement level indicator is calculated from this scaled error to give the number of times the triangle should be refined or derefined. Since the error estimate is applied at the end of a time-step it is too late to make the refinement decision. Methods are therefore used for the prediction of the growth of the spatial error using linear or quadratic interpolants. The decision about whether to refine a triangle is based on these predictions, and the estimate made at the end of the time-step can be used to predict errors at future time-steps. Generally it is found that large spatial errors coincide with regions of steep spatial gradients. The spatial error estimate can also be used to indicate when the solution is being solved too accurately and can indicate which regions can be coarsened.

For applications such as atmospheric modelling it is important that a maximum level of refinement can be set, to prevent the code from adapting to too high a level in regions with concentrated emissions. This is especially important around point or highly concentrated area sources. Here, because of the nature of the source, steep

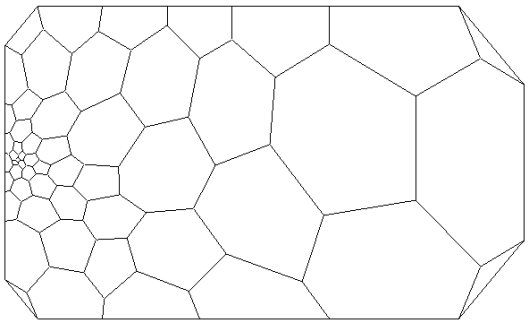


Figure 5: The structure of the level 0 mesh. The length of the domain is 300km and the width 200km. The smallest and largest mesh lengths are approximately 5 and 60km respectively for the level zero domain.

spatial gradients are likely to persist down to very high levels of refinement. This would have the consequence that the number of elements on which the p.d.e.s had to be discretised would become prohibitively large. For the following test problems the maximum level of refinment was therefore limited to level 3.

9 Single Source Pollution Plume Example

The example used here to illustrate the effectiveness of the adaptive mesh is that of a single plume pollution source. In this case the initial two dimensional mesh was generated with only 100 elements. It is difficult to relate the size of unstructured meshes directly to regular rectangular ones, but our original mesh was comparable to the size of mesh generally used in regional scale atmospheric models, the largest grid cell being approximately 60km along its longest edge. Close to the chimney the mesh was refined to elements of length 5km ensuring that it would be refined to a reasonable resolution in this region of steep gradients. If we allow the mesh to refine two levels then the smallest possible mesh size close to the chimney will be 1.25km in length. Spatial errors in the concentration of NO were chosen as the criterion from which to further refine the mesh. Test runs showed that regions of high spatial error coincided with steep spatial gradients. The mesh can therefore be considered to adapt around steep NO concentration gradients. Each run was carried out over a period of 48 hours starting from midnight on day 1, so that the diurnal variations could be observed. We present here only a selection of the results which illustrate the main features relating to the mesh adaptation. Figures 5 and 6 allow a comparison to be made between the structure of the base mesh and a mesh that has been adapted up to level 2 at 14.00 on day 2. In these figures the sides of the polygons represent the distance between cell centres on the triangular mesh. The main area of mesh refinement is along the plume edges close to the chimney, indicating that there is a high level of structure in these regions. On the coarse mesh the plume is dispersed over a much larger area than on the fine mesh and most of the plume structure is lost. Close to the stack the concentration of O_3 is much

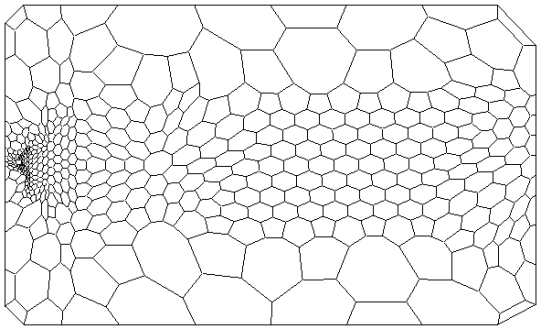


Figure 6: The structure of the level 2 adaptive mesh.

lower than that in the background because of high NOx concentrations. The inorganic chemistry is dominant in this region and the ozone is consumed by the reaction: NO + $O_3 \rightarrow NO_2 + O_2$.

In Figure 7 we present a cross plume profile of the NO_2 concentrations at a distance of 10km downwind of the chimney stack for case A at the same time as the previous figure. The figure clearly shows the features at the edge of the plume which are revealed by the adaptive solution. From the base mesh, where the distance between elements along the y-axis close to the stack is 20km, it appears that the concentration of NO_2 rises to a peak in the centre of the plume. If the mesh is refined to higher levels then we start to see the true structure of the plume emerging. With a level 3 solution we can see that the peak concentrations are actually found along the edges of the plume and that the concentration of NO_2 drops to very low levels at the plume centre. From the area under these curves it is found that there is a 30% difference between the overall level 0 and the level 3 concentrations. This shows that not only the peak concentrations, but the total integrated concentrations are very different for the different levels of mesh adaptation. It is clear therefore that using a very coarse grid in regions of steep spatial gradients can lead to an over estimate of total pollutant concentrations for systems with nonlinear chemical schemes.

Figures 8, 9 show that in the case considered here the plume is over-dispersed in the level 0 case and the spatial distribution of ozone is therefore inaccurately represented. For the clean air case, the levels of ozone drop considerably in the plume compared to the background since the levels of NO are much higher there. For the level 0 case these lowered concentrations spread over very large distances owing to the over-dispersion of the plume. The location of reduced/raised concentrations will therefore be incorrect for the level 0 results in all three cases. For each scenario the level 0 solution leads to a smoothing out of the ozone profiles so that the true structure caused by the interaction of the plume with background air is missed.

The striking result is that the adaptive solution reveals features such as peak levels of NO_2 and O_3 which could not be detected using a coarse mesh. The change in mesh refinement also resulted in a change in overall or integrated concentration levels. This indicates that due to strongly nonlinear terms in the chemical reaction rates, the source terms in the p.d.e. will be mesh dependent. Without using a fine mesh over the whole

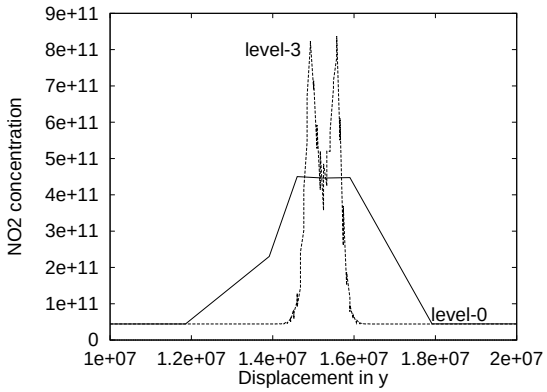


Figure 7: Cross plume NO_2 profiles 10km from stack in molecules cm^{-3}, showing how the level 3 solution captures the structure of the plume.

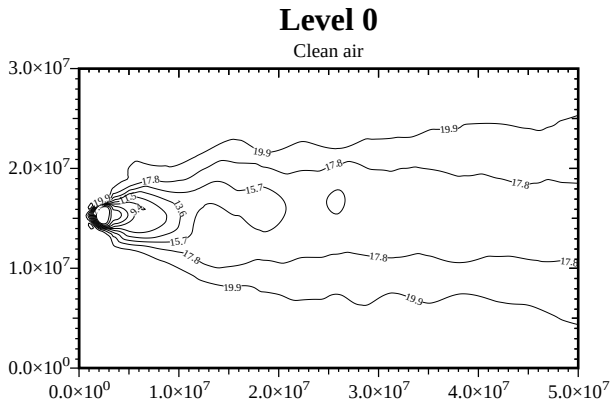


Figure 8: Ozone contours for case C, clean air, level 0 calculation.

domain so that the concentrations in neighbouring cells differ only very little, the effects of this nonlinearity could be quite significant. To reduce the effects it is important to refine the mesh at least in regions of steep spatial gradients. This has been partially addressed by the telescopic methods presently used in air quality models. However, the present test case has shown that steep gradients can occur at long distances downwind from the source, for example the change in ozone concentrations along the edges of the plume. Adaptive algorithms seem to present a successful method of achieving accuracy in such regions and can do so in an automatic way. The main limitation of the above approach is that only two space dimensions have been considered. The next issue to be resolved is whether mesh adaptation is necessary in the vertical direction and how appropriate a method of lines approach is in three space dimensions. These are the issues considered in the next five sections.

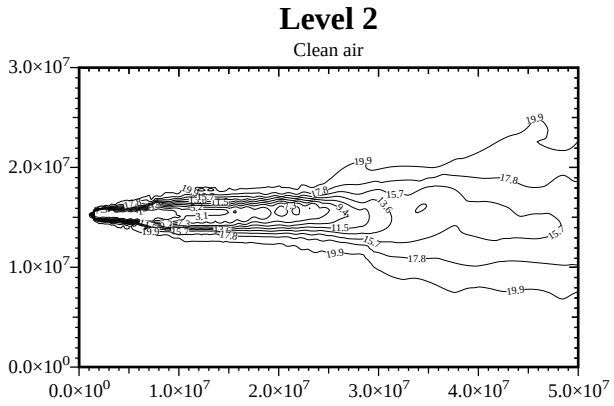


Figure 9: Ozone contours for case C, clean air, level 2 calculations.

10 Three Space Dimensional Computations

The standard approach with three space dimensional atmospheric dispersion problems is that 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.

In the work described here we have used two approaches for solving three space-dimensional atmospheric dispersion problems. Both approaches use a fully 3D unstructured mesh based on tetrahedral elements. The first approach is described in [17] and is the closer of the two approaches to the two dimensional case described in Section 6 above in that a cell-centred finite volume scheme is used for the spatial discretization. In this case a conventional method of lines approach is used based on a modified version of the SPRINT time integration package. The linear algebra approach of Section 7 is used with a simple first order spatial discretization approach. The disadvantage of this approach is that it requires a much larger number of unknowns for a given mesh than if a cell-vertex approach is used with the solution unknowns being positioned at the nodes of the mesh. The price that is paid for this reduction in the number of unknowns is an increase in the complexity of the discretization method. There is also the well-known difficulty that the cell-vertex discretization may not preserve the positivity of the solution on certain meshes due to the discretization of the diffusion operator [9]. Although it may be possible to address this issue within a method of lines framework, the need to preserve positivity and the different timescales needed for advection and chemistry have led us to employ an operator splitting approach.

The next section describes the 3-D unstructured mesh discretization method and the flow integration scheme which advances the solution in time. Section 12 contains the mesh adaptation strategy which changes the connectivity in the data structure of the mesh in response to changes in the solutions. Section 13 explains the implicit-explicit method used to solve the transport equation. is described. Section 14 contains the

test examples which have been designed to determine the importance of mesh structure on both horizontal and vertical mixing for typical meteorological conditions. The test problem describes the dispersion of pollutants from a single source due to typical boundary layer wind profiles. Finally, we draw conclusions in Section 15 regarding the importance of adaptive mesh method in solving 3-D atmospheric flow problems.

11 Three Space Dimensional Discretization

The atmospheric 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 hexehedra 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. Each component of the diffusion equations (17) is discretized using same method. Hence, for simplicity, instead of treating the vector $\underline{c}$, we choose one of its component, c say, and describe its discretization.

11.1 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 Gauss divergence theorem:

$$\begin{aligned}\int_{\Omega_0}(\frac{\partial f}{\partial x}+\frac{\partial g}{\partial y}+\frac{\partial h}{\partial z})\,d\Omega &= \int_{\partial\Omega_0}(f\,n_x+g\,n_y+h\,n_z)\,dS \\ &= \sum_k (f\,S_x+g\,S_y+h\,S_z), \end{aligned} \tag{28}$$

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 Figure 10. The number of such quadrilateral faces attached to an edge depends on the number of tetrahedra neighbours to that edge. There are four tetrahedra sharing the edge i in Figure 10. The projected area, A_i, associated with the edge i is calculated in terms of the quadrilateral face areas, a_1,a_2,a_3,a_4, as

$$(A_i)_x=\sum_{j=1}^{4}(a_j)_x,\quad (A_i)_y=\sum_{j=1}^{4}(a_j)_y,\quad (A_i)_z=\sum_{j=1}^{4}(a_j)_z. \tag{29}$$

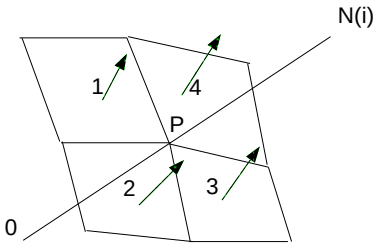


Figure 10: Dual mesh faces attached to an edge.

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 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 equation (28) is replaced by

$$\int_{\Omega_0} (\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z})\,d\Omega = \sum_i (f_p\,(A_i)_x + g_p\,(A_i)_y + h_p\,(A_i)_z), \tag{30}$$

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 equally opposite contribution to the other node that forms the edge.

11.2 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 conservative property numerically. If u, v, w are the wind velocities, then they must satisfy

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \tag{31}$$

Here we enforce mass conservation using the variational calculus technique of Mathur and Peters [20]. The technique attempts to adjust the wind velocity in a manner such that the interpolated data are minimally changed in a least square sense, and at the same time, the adjusted values satisfy the mass conservation constraint. The details are provided by Ghorai et al. [12].

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. The wind velocity remains mass conservative in the numerical sense in the base mesh since the unstructured base mesh is regular. But the wind field is not numerically mass conservative once the grid is refined (derefined). A representative one dimensional neutral boundary layer velocity is shown in Figure 11(b). The velocity field is adjusted in the refinement region but away from the refinement region, the velocity almost remains unchanged.

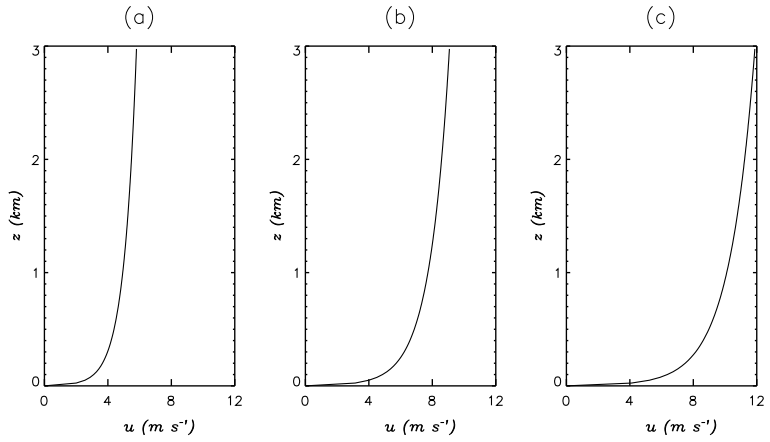


Figure 11: A representative variation of wind with height for (*a*) stable, (*b*) neutral and (*c*) unstable boundary layers.

The base grid spacings along the vertical increases upwards. Thus the grid quality near the ground is worse due to large aspect ratio of the tetrahedron. The velocity corrections decrease upwards as the refine region moves upwards. Suppose we have a refine region at 150 meters height. The maximum corrections are 12, 14 and $0.06\,\mathrm{cm\,s^{-1}}$ respectively for the u, v and w components. For a refine region at 600 meters height the corresponding components are 11, 11 and $0.03\,i\mathrm{cm\,s^{-1}}$. And finally, the corresponding corrections decreases to 0.3, 0.2 and $0.0002\,\mathrm{cm\,s^{-1}}$ i at 1.8 km height. The neutral boundary layer velocity increases from 0 to $9\,\mathrm{m\,s^{-1}}$ as z increases from 0 to 3 km and so the velocity corrections are small.

11.3 Advection scheme

The discretization of the term

$$\int_{\Omega_0} \left[\frac{\partial (uc)}{\partial x} + \frac{\partial (vc)}{\partial y} + \frac{\partial (wc)}{\partial z} \right] d\Omega \equiv \int_{\Omega_0} \nabla . F \, d\Omega,$$

where

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

is done by using an algorithm used is based on that of Barth and Jespersen [8] and uses equation (30), to rewrite the equation above as an edge based computation.

$$\sum_i \left[u_p\left(A_i\right)_x + v_p\left(A_i\right)_y + w_p\left(A_i\right)_z\right]\left(c\right)_p = \sum_i \left(\underline{F}_p.\underline{A}_i\right)\left(c\right)_p, \tag{32}$$

where $\underline{A}_i$ is called 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. Evaluation of this expression is by using the upwind limited approach of Barth and Jesperson [8]. The values of limiter functions and gradient at the nodes are not calculated on a node by node basis (which is CPU intensive), instead they are calculated in an edge-based operation, [8, 9]. The time step for the advection scheme is chosen so that it satisfies the CFL condition (Wierse), [37]. The minimum of the time steps over all the vertices constitute the time step for the advection scheme. Again this computation can be cast into an edge-based operation.

11.4 Diffusion scheme

The diffusion term is discretized using the standard linear finite element method or the equivalent cell-vertex method described by Barth. Again the key feature is that the calculation of the diffusion terms is reordered so that it involves edge gradient terms, see Barth [9]. The disadvantage of the standard approach is that it does not preserve positivity of the solution for certain meshes, see Barth [9]. Very recent work has provided methods that begin to address this issue, [23].

12 Mesh Adaptation

The cell-vertex scheme approach is hierarchical in nature, [10, 28], 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 adaptation algorithm tree data structure. The data objects contains all flow and connectivity information sufficient to adapt the mesh structure and flow solution by either local *refinement* or *derefinement* procedures. The mesh adaptation 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 adaptation 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 adaptation. 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 adaptation step.

For reasons of both tetrahedral quality control and algorithm simplicity only two types of element subdivisions are used (Speares and Berzins, [28]). 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. The 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. 12 demonstrates regular and green refinement for a tetrahedron. The five

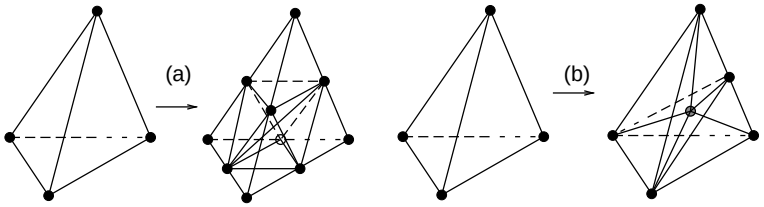


Figure 12: (*a*) Regular refinement based on the subdivision of tetrahedron by dissection of interior diagonal (1:8) and (*b*) "green" refinement by addition of an interior node (1:6).

possible refinement possibilities (if all the edges are are refined then the parent element is regularly refined) give rise to between 6 and 14 child green elements.

The choice of adaptation criteria is very important since it can produce either large or small number of nodes depending on the condition used to flag a edge for the adaptation. Also, when there are a large number of species, the choice of a given criteria might result in high resolution for some species but low resolution for the other species. Let 0 and i be the nodes for a given edge $e(0,i)$. We calculate $tolg$ and $tolc$ by

$$tolg = \frac{\left|(c)_0 - (c)_i\right|}{\text{dist}} \quad \text{and} \quad tolc = \frac{(c)_0 + (c)_i}{2},$$

where dist is the length of the edge $e(0,i)$. We refine the edge $e(0,i)$ if $tolg$ and $tolc$ exceed some tolerances, otherwise it is derefined. Also a maximum level of refinement is specified at the beginning so that if a edge is targeted for refinement but it is in the maximum level, then it is kept unchanged.

Suppose we have two edges with $tolg = 100$ and 200. If we take the tolerance parameter, T_g say, for $tolg$ 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 edge with $tolg = 200$ is greater than the error in the edge with $tolg = 100$. It might be advantageous to use two sets of $T_g = 50$ and 150. If $tolg > 150$, than we refine an edge to maximum level and if $50 < tolg < 150$, then

we refine an edge to the level just lower than the maximum levels. 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.

13 Time Integration for 3D Problems

Although in two space dimensional calculations we have used sophisticated space-time error control techniques (Berzins et al.[7], Tomlin et al.[32]), the need to preserve positivity, to reduce computational cost and the need to take into account the different timescales needed for the integration of advection and chemistry has led us to use an operator splitting technique. In this approach, the chemistry is decoupled from the transport. The main reason for the use of this is that it is much easier to ensure positivity of the solution components. The nonlinear chemistry part gives rise to stiff ordinary differential equations. We solve the chemistry part using the SPRINT time integration methods (Berzins et al.,[7]) and also using Gauss-Seidel iteration of Verwer [35]. The transport step is considered first. If $\underline{c}^n$ denotes the species concentration at time level n, then the species concentration at the next time step is given by

$$\underline{c}^{n+1} = \underline{c}^n + \triangle t \underline{g}(c) + \triangle t \underline{f}(c) + \underline{S}, \tag{33}$$

where $\triangle t$ is the time step and $\underline{g}(c)$ is the advection operator and $\underline{f}(c)$ is the diffusion operator. In a fully explicit scheme, $\underline{f}$ and $\underline{g}$ 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 equation (33), 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

$$(\frac{1}{\triangle t} + a_i)(\underline{c}^{n+1})_i = \sum_{j \in \mathrm{N(i)},\, j \neq i} a_j \, (\underline{c}^{n+1})_j + Q_i^n, \tag{34}$$

where i is varied over all the nodes and

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

The time step $\triangle t$ is chosen to equal to the time step due to advection only. The value of time step 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, $\triangle t$ is $\approx$ 35 s for the stable atmospheric boundary layer but decreases to $\approx$ 18 s for the unstable atmospheric boundary layer. Thus the time step is smaller for higher wind speed and vice versa. The system of equations given by equation (34) 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

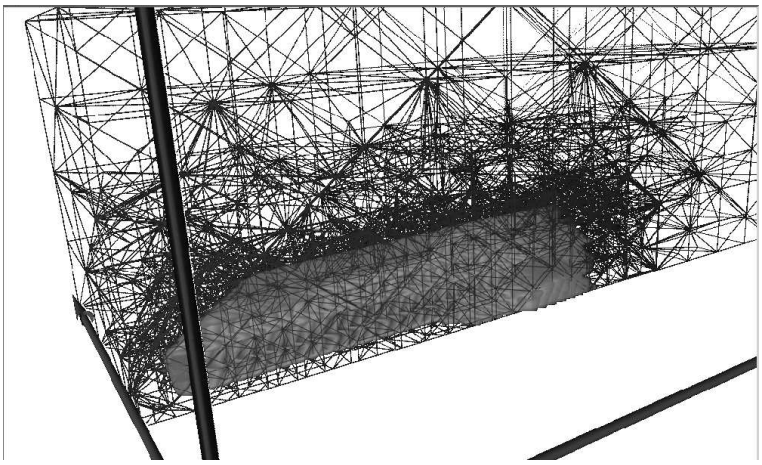


Figure 13: A representative mesh for the 3D atmospheric dispersion problem.

extra time integration and splitting error which is not easily quantified. In future work we will revisit this issue of a standard method of lines approach versus the operator splitting approach used here.

14 Three Dimensional Test Examples

The advection scheme has been tested by advecting a puff of NO around a horizontal circle without any diffusion [33]. 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 km 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 a 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 Figure 11 and Figure 14 from Seinfeld, [25].

The horizontal diffusion coefficients K_x and K_y are kept constant and equal to 50 m^2 s^{-1}. The initial tetrahedral mesh is generated by dividing the whole region into cuboids and then subdividing a cuboid into 6 tetrahedral elements. The cuboids are 4 km 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.

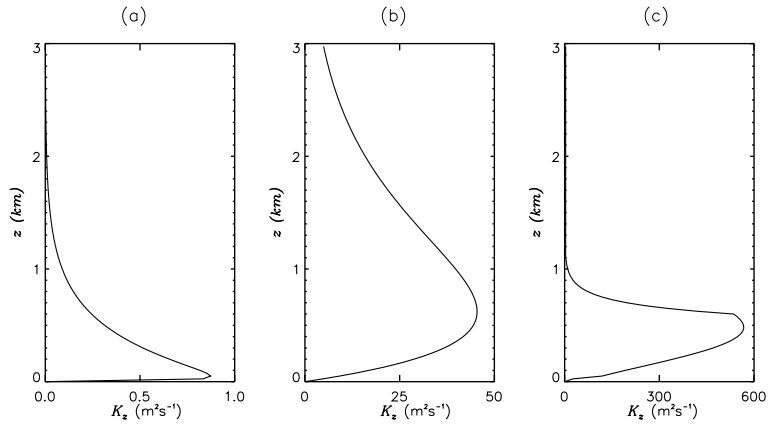


Figure 14: A representative variation of vertical diffusion with height for (*a*) stable, (*b*) neutral and (*c*) unstable boundary layers.

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 and confined very near to the ground level for the stable boundary layer. Thus the concentration does not mix much above the source height. The height of the 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 the neutral and unstable boundary layers. 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 a 3 km circle with the point source as the centre and it lies vertically within 300 meters from the source. The initial number of nodes is $6,442$ for all the three boundary layers. The number of nodes for the telescopic method remains $6,442$ throughout the simulation period. On the other hand, the adaptive grid is refined/derefined as the solution advances. The time step $\triangle t$ for the implicit-explicit scheme is small (usually less than 1 minute) due to small vertical spacings near the ground level which effect the CFL condition. Instead of carrying out the adaptation after every time step (which is CPU intensive), the adaptation is carried out approximately every 20 minutes. This prevents large amount computational effort being used to perhaps refine very few tetrahedra each time step and does not significantly affect solution accuracy.

14.1 Grid adaptation

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 values 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 centreline is shown in Fig. 15. 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 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. 15(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 inversion layer compared to the gradient near the ground. Thus the edges near the inversion layer refine to higher level than the edges near the ground level.

The adaptive grid refinement at three different locations in the cross-wind direction is shown by Ghorai et al.[12]. 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 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.

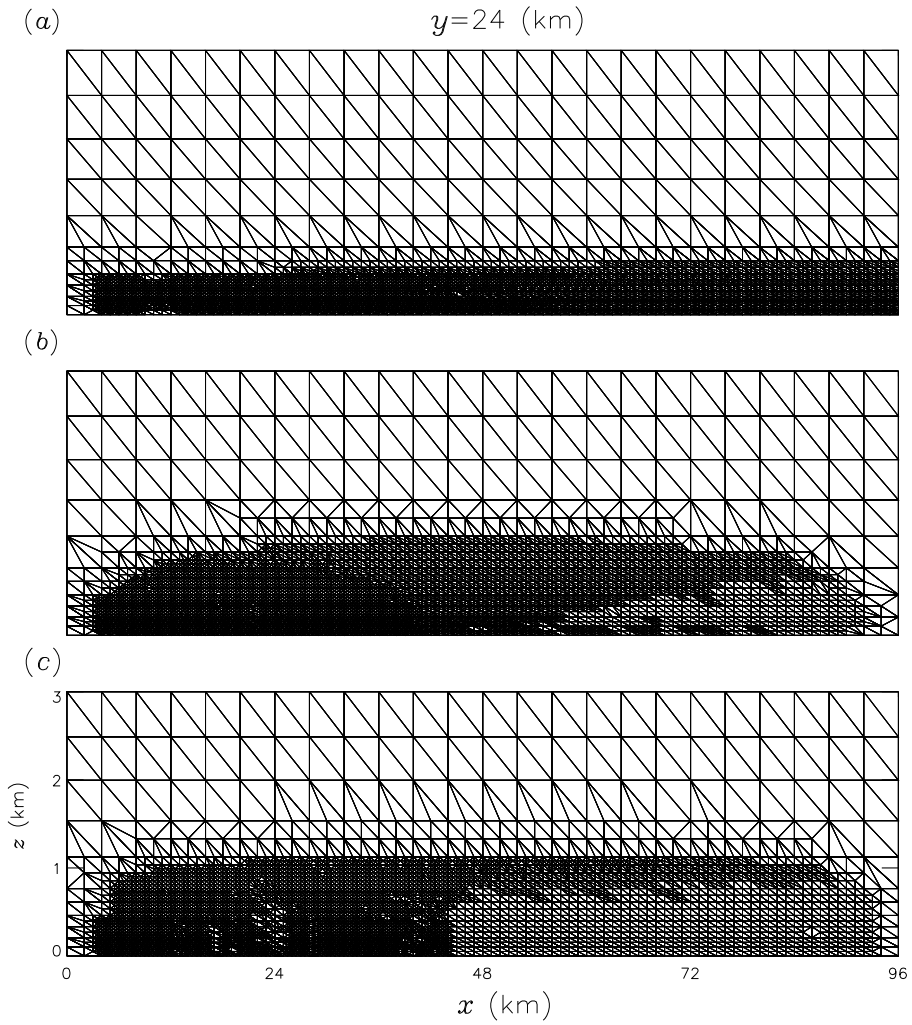


Figure 15: Grid refinement in the vertical plane through the source along the downwind direction for the (*a*) stable, (*b*) neutral and (*c*) unstable boundary layers.

14.2 Downwind concentration

The solutions downwind along the plume centre-line in the ground level are shown in Fig. 16. The maximum 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 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 a Origin2000 com-

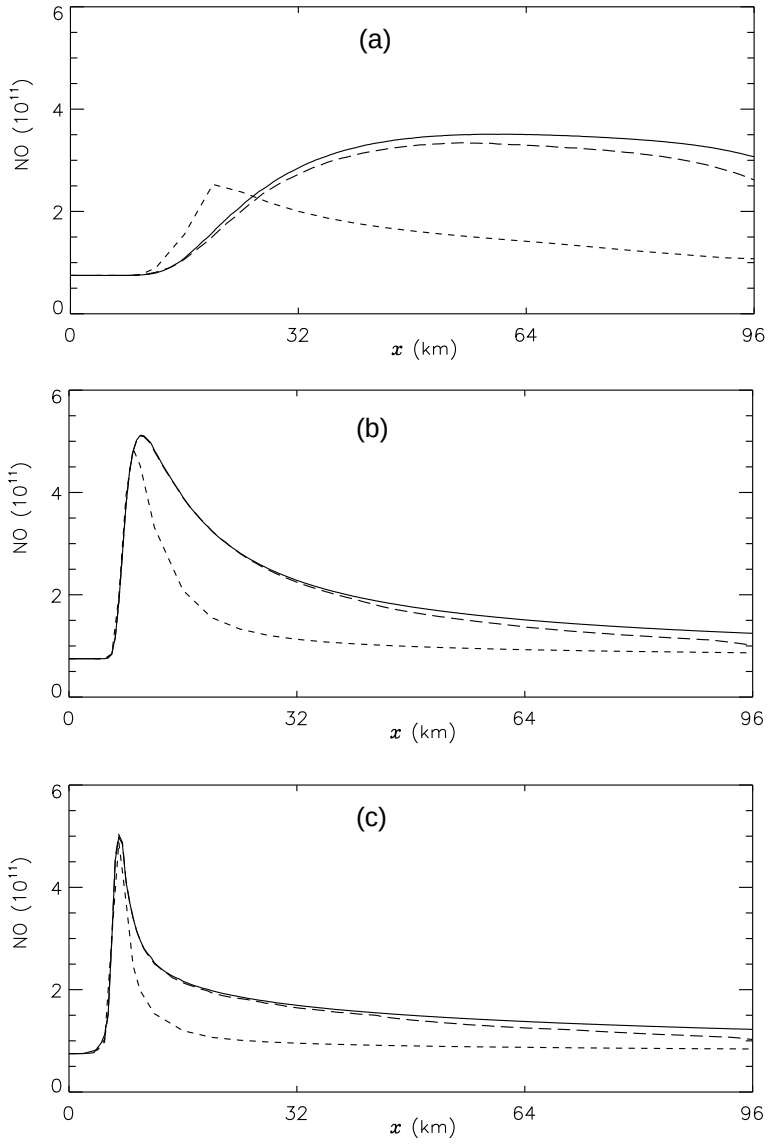


Figure 16: Comparison of the solution along the plume centre-line in the ground level for the (*a*) stable, (*b*) neutral and (*c*) unstable boundary layers. The solid, dotted and dashed lines correspond the solutions in the reference, telescopic and adaptive grids.

puter. For the neutral boundary layer, the total CPU times are approximately 1, 7 and 25 hours for the telescopic, adaptive and reference grids respectively. Thus the adaptive method is efficient compared to the other methods and achieves greater accuracy in a

reasonable time.

15 Discussions and conclusions

In this paper we have described a method of lines approach to the solution of transient reacting flow problems. In particular, the atmospheric diffusion equation was solved by using unstructured, adaptive meshes with the method of lines in both two and three space dimensions. However because of effieciency and positivity considerations, the three space dimensional case was solved by using operator splitting. The single most important conclusion is that there are key features of plume characteristics which cannot be represented by the coarse meshes generally used in regional scale models.

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. The adaptive mesh methods may also use 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. However, there is an extra cost with the adaptive codes, that of periodically refining/coarsening the mesh. In particular, the test cases 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 and of 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. The results here demonstrate that for neutral it unstable boundary layers 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 affect 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.

Our general conclusion is that the adaptive method of lines approach works well for two space dimensional problems and in those cases it is possible to use standard codes providing that it is possible to make use of sophisticated linear algebra methods which are tailored to the problem. In the case of three dimensional problems however it seems more necessary to use tailor-made codes either based on the method of lines as in [17] or using the operator splitting approach described here. Very recent work by Verwer and others has suggested that the approach we used in two dimensions should also be used in three space dimensions rather than introducing an operator splitting error. The challenge is now to implement this in a sufficiently efficient way to make the method

of lines competitive with operator splitting in terms of efficiency.
Acknowledgements–This research has been supported by grants from NERC and from the Pakistan Government for one of us (IA). The funding for the SPRINT2D and 3D software has come from Shell Global Solutions. These calculations have been carried out on an Origin2000 machine with support from a JREI grant. We would also like to thank our many colleagues who helped on this project such as G.Hart, J.Smith, M.Pilling and many others.

References

[1] I.Ahmad and M.Berzins MOL Solvers for Hyperbolic PDEs with Source Terms. Submitted to special issue of *Mathematics and Computers in Simulation*.

[2] I. Ahmad and M. Berzins. An algorithm for ODEs from atmospheric dispersion problems. *Applied Numerical Mathematics*, (1997), **25** 137–149.

[3] M. Berzins. Temporal error control for convection-dominated equations in two space dimensions. *SIAM Journal on Scientific and Statistical Computing*, **16**, (1995), 558–580, .

[4] M. Berzins and J. M. Ware. Solving convection and convection reaction problems using the M.O.L. *Applied Numerical Mathematics*, **20** (1996), 83–99, .

[5] Berzins M., Dew P.M., and Furzeland R.M. Developing software for time-dependent problems using the method of lines and differential algebraic integrators. *Applied Numerical Mathematics* **5**, (1989), 375-390.

[6] Berzins M. and Ware J.M. Positive cell-centered finite volume discretization methods for hyperbolic equations on irregular meshes. *Applied Numerical Mathematics* **16**, (1995), 417-438.

[7] Berzins, M., Fairlie, R., Pennington, S. V., Ware, J. M. and Scales, L. E. SPRINT2D: Adaptive Software for PDEs. *ACM Transactions on Mathematical Software* **24**, (1998), 475-499.

[8] Barth, T.J. and Jesperson, D.C. The Design and Application of Upwind Schemes on Unstructured Meshes. *AIAA-89-0366*, (1989), 9-12.

[9] T.J. Barth, Numerical Aspects of Computing Viscous High Reynolds Number Flows on Unstructured Meshes. AIAA Paper 91-0721, 29th Aerospace Sciences Meeting, January 7-10, (1991), Reno Nevada.

[10] Biswas, R. and Strawn, R.C. A new procedure for dynamic adaption of 3-D *unstructured grids. Applied Numerical Mathematics* **13**, (1994), 437-452.

[11] Chock D.P. A comparison of numerical methods for solving the advection equation III. *Atmos. Env.* **25A**, (1991), 553-571.

[12] S.Ghorai A.S.Tomlin and M.Berzins " Resolution of pollutant concentrations in the boundary layer using a fully 3-D adaptive gridding technique." *Atmospheric Environment* **34**, 18, (2000), 2851-2863.

[13] Hart G., Tomlin A.S., Smith J. and Berzins M. Multi-scale atmospheric dispersion modelling by use of adaptive gridding techniques. *Environmental Monitoring and Assessment* **52**, (1998), 225-238.

[14] Hov, Ø., Zlatev, Z., Berkowicz, R., Eliassen, A. and Prahm, L. P. Comparisons of numerical techniques for use in air pollution models with non-linear chemical reactions. *Atmospheric Environment* **23**, (1989), 967-983.

[15] Jacobs H.J., Feldman H., Kass H., and Messesheimer M. The use of nested models for air pollution studies: an application of the EURAD model to a SANA episode. *Journal of Applied Meteorology* **34**, (1995), 1301-1319.

[16] Joe B. and Simpson R.B. Triangular meshes for regions of complicated shape, *Int. J. Numer. Meth. Eng.* **23**, (1991), 987-997.

[17] C.R. Johnson, M. Berzins, L. Zhukov, and R. Coffey. SCIRun: Application to Atmospheric Dispersion Problems Using Unstructured Meshes. pp. 111-122 in Numerical Methods for Fluid Dynamics VI, ed. M.J.Baines, ICFD, Wolfson Building, Parks Road, Oxford. ISBN 0 9524929 11, (1998).

[18] R. J. Leveque and H.C. Yee. A study of numerical methods for hyperbolic conservation laws with stiff source terms. *Journal of Computational Physics*, **86** (1990), 187–210.

[19] Liu S.C., Trainer M., Fehsenfeld F.C., Parrish D.D., Williams E.J., Fahey D.W., Hubler G., and Murphey P.C.) Ozone production in the rural troposphere and the implications for regional and global ozone distributions *J. Geophys. Res.* **92**, (1987), 4191-4207.

[20] Mathur, R. and Peters, L.K. Adjustment of wind fields for application in air pollution modelling. *Atmospheric Environment* **24A**, (1990),1095-1106.

[21] S. V. Pennington and M. Berzins. New NAG library software for first-order partial differential equations. *ACM Transactions on Mathematical Software* **20**, (1994), 63–99.

[22] Peters L.K., Berkovitz C.M., Carmichael G.R., Easter R.C., Fairweather G., Ghan S.J., Hales J.M., Leung L.R., Pennell W.R., Potra F.A., Saylor R.D. and Tsang T.T. The current and future direction of Eulerian Models in simulating the tropospheric chemistry and transport of trace species: a review *Atmospheric Environment* **29**, (1995), 189-222.

[23] M. Putti and C. Cordes, Finite Element Approximation of the Diffusion Operator on Tetrahedra, *SIAM Journal on Scientific Computing* **19**,(1998), 1154-1168, .

[24] Rajaona T., Ciccoli M.C., and Coppalle A. Local refinement around a pollution source using a domain decomposition method. In *Proc. Int. Conf. on Air Pollution Modelling and Simulation*, Paris, France (1998).

[25] Seinfeld, J.H. (1986) Air Pollution. *Wiley, New York.*

[26] Sillman S., Logan J. A., Wofsy S. C. *J. Geophys. Res.* **95**, (1990), 1837

[27] Skamarock W., Oliger J. and Street R.L. Adaptive Grid Refinement for Numerical Weather Prediction. *Journal of Computational Physics*, **80**, (1989), 27-60.

[28] Speares, W. and Berzins, M. A 3D unstructured mesh adaptation algorithm for time-dependent shock-dominated problems. *International Journal for numerical Methods in Fluids* **25**, (1997), 81–104.

[29] Spekreijse S. Multigrid solution of monotone second order discretizations of hyperbolic conservation laws. *Math. Comp.*, **47**, 179, (1987), 135-155.

[30] Sunderam S., Logan J.A. and Wofsy. S.C. A Regional Scale Model for Ozone in the United States With Subgrid Representation of Urban and Power Plant Plumes. *Journal of Geophysical Research*, **95**, (1990), 5731-5748.

[31] Talat, O. A quantitative analysis of numerical diffusion introduced by advection algorithms in air quality models. *Atmospheric Environment* **31**, (1997), 1933-1940.

[32] Tomlin, A.S., Berzins. M., Ware. J., Smith, J. and Pilling, M.J. On the use of adaptive gridding methods for modelling chemical transport from multi-scale sources. *Atmospheric Environment* **31**, (1997) 2945-2959.

[33] Tomlin, A.S., Ghorai, S., Hart, G. and Berzins, M. The use of 3-D adaptive unstructured meshes in pollution modelling. i In *Large-Scale Computations in Air Pollution Modelling*, ed. Z.Zlatev et al., (1999) Kluwer Academic Publishers.

[34] VanLoon M. Numerical methods in smog prediction. *PhD. Thesis*, (1996) CWI Amsterdam.

[35] Verwer, I. (1994) Gauss-Seidel iteration for stiff ODEs from chemical kinetics. *SIAM Journal on Scientific computing* **15** 1243-1250.

[36] Vreugdenhil C. B. and Koren B. Numerical methods for advection-diffusion problems. *Notes on numerical fluid mechanics* **45**, (1993), Vieweg, Braunschweig/Weisbaden, ISBN 3-528-07645-3.

[37] Wierse, M. A new theoretically motivated higher order upwind scheme on unstructured grids of simplices. *Advances in Computational Mathematics* **7**, (1997), 303-335.