# Mesh Quality and Moving Meshes for 2D and 3D Unstructured Mesh Flow Solvers.

M.Berzins, P.K.Jimack and M.Walkley
*CPDEs Unit, School of Computer Studies, University of Leeds*
*Woodhouse Lane, Leeds LS2 9JT, U.K.*
L.J.K.Durbeck
*Computer Science, University of Utah, Salt Lake City, Utah, USA.*

**Abstract**

The issue of mesh quality for the solution of problems with anisotropic solutions on unstructured triangular and tetrahedral meshes is considered. A three dimensional time-dependent problem from atmospheric dispersion is described and used to motivate a discussion of mesh quality. A survey of recent research in the development of finite element methods describes work on moving mesh methods, on anisotropic meshing algorithms and on the provision of appropriate error estimates. As such error estimates are presently not always available, one option is to use local error estimates for transient problems. One such approach is described and is used to decide how to adapt the mesh. A visualization system for identifying poor quality elements is described and used to find the poor quality elements in the atmospheric dispersion problem.

## 1 Introduction

The issue of what is an appropriate spatial mesh is as old as the finite element method itself, but the increasingly complex nature of 3D applications may involve dealing with multicomponent problems with time dependence, turbulence and anisotropy to name but some of many possible complications.

A relatively simple example which is useful to illustrate the difficulties is the following 3D advection-reaction problem. The example is a model of atmospheric dispersion from a power station plume which is a concentrated source of NOx emissions, [31]. The photo-chemical reaction of this NOx with polluted air leads to the generation of ozone at large distances downwind from the source. An accurate description of the distribution of pollutant concentrations is needed over large spatial regions in order to compare with field measurement calculations. The present trend is to use models incorporating an ever larger number of reactions and chemical species in the atmospheric chemistry model. The complex chemical kinetics in the atmospheric model gives rise to abrupt and sudden changes in the concentration of the chemical species in both space and time. These changes must be matched by changes in the spatial mesh and the timesteps if high resolution is required, [31]. This application is modelled by the atmospheric diffusion equation in three space dimensions given by:

$$\frac{\partial c_s}{\partial t} + u\frac{\partial c_s}{\partial x} + v\frac{\partial c_s}{\partial y} + w\frac{\partial c_s}{\partial z} = D + R_s + E_s - \kappa_s c_s, \tag{1}$$

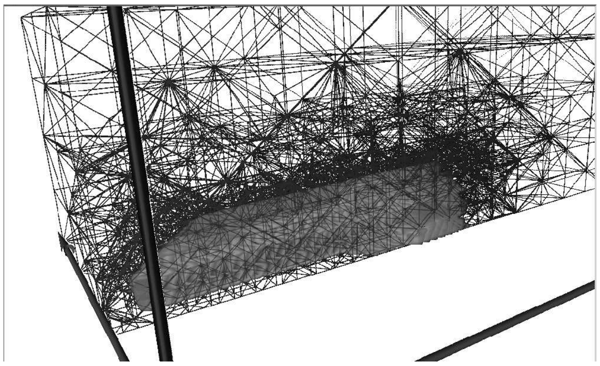


Figure 1: Tetrahedral Mesh for Reacting Flow P.D.E.

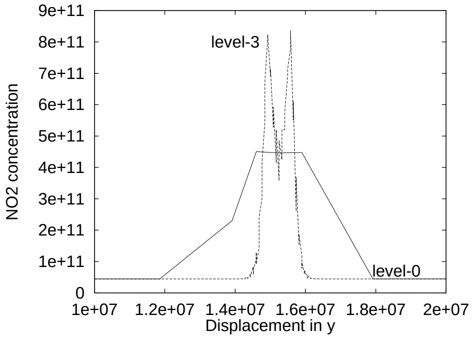



Figure 2: Cross section of plume

where $c_s$ is the concentration of compound $s$ , $u, v$ and $w$, are constant wind velocities and $\kappa_s$ is the sum of the wet and dry deposition velocities. $E_s$ describes the distribution of emission sources for compound $s$ and $R_s$ is the chemical reaction term which may contain nonlinear terms in $c_s$. $D$ is the diffusion term which is set to zero here. For n chemical species a set of n partial differential equations (p.d.e's) is formed where each is coupled to the others through the nonlinear chemical reaction terms.

The test case model covers a region of 300 km x 500 km and is a three-dimensional form of that used by [31] and although far from detailed, does represent the main features which would commonly be found in an atmospheric model including slow and fast nonlinear chemistry, concentrated source terms and advection. Although the chemical reaction terms involve only 7 species they still represent the main features of a tropospheric mechanism, namely the competition of the fast inorganic reactions given by $NO_2 \overset{O_2}{\rightarrow} O_3 + NO$ and $NO + O_3 \rightarrow NO_2 + O_2$ 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 reaction rate constants and the background concentrations that form the initial conditions for the model are given by Tomlin et al. [31]. These concentrations will then change diurnally as the chemical transformations take place. The power station is taken to be the only source of NOx and this source is treated by setting 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. The concentration in the chimney corresponds to an emission rate of NOx of 400kghr$^{-1}$ with only 10% of the NOx being emitted as $NO_2$. A constant wind speed of 5ms$^{-1}$ in the x-direction with y and z components of one tenth of this value is assumed.

Spatial discretisation of the model atmospheric diffusion equation on unstructured tetrahedral meshes reduces the set of p.d.e's in four independent variables to a system of ordinary differential equations (o.d.e's) in one independent variable: time. This system of o.d.e's can then be solved as an initial value problem. For advection-dominated problems it is important to choose a discretisation scheme which preserves the physical range of the solution [30]. The method used here is the cell-centered finite volume discretisation scheme used by [30] which enables accurate solutions to be determined for both smooth and discontinuous flows by making use of upwind techniques for the advective parts of the fluxes.

Figure 1 shows the plume developing with the adaptive mesh clustered around the developed portion of the solution. The visualisation was realised by running the parallel code in conjunction with the SCIRUN system at the University of Utah, see [23]. The main area of mesh refinement is along the plume edges close to the chimney. Using the adaptive mesh, we can clearly see the plume edges and can easily identify areas of high concentrations. The effects of the plume on the ozone concentration also provides some interesting results. Close to the plume the concentration of $O_3$ is much lower than that in the background. Due to the high NOx concentrations the inorganic chemistry is dominant in this region and ozone is consumed. As the plume travels downwind and the NOx levels decrease, the plume gradually picks up emissions of VOC's and leads to the production of $NO_2$ which in turn causes levels of ozone to rise above the background levels at quite large distances downwind from the source of NOx.

The issue of whether the mesh is appropriate for this application is somewhat more complex than for a simple linear problem. Strong local variations in solution component values make it difficult to assess the quality of the mesh for each component without somehow incorporating solution behaviour.

The importance of mesh refinement is shown by the 2D cross section in Figure 2 of the plume close to its source, see Tomlin et al. [31, 32]. In this case the interplay between mesh refinement and nonlinear chemistry produces a solution on the level 3 refined mesh with peak values that differ by a factor of two from the coarse (level 0) mesh solution. This example thus shows that when solving multi-component problems there is much to be gained from making use of an appropriate mesh for the strongly directional model flow.

In the rest of this paper the issue of what a good mesh is for a flow problem is considered in the following way. In Section 2 a number of moving mesh methods are used to demonstrate the many improvements possible when non-uniform meshes are used. Section 3 will describe similar developments in three space dimensions. Although there is much convincing computational evidence for the usefulness of stretched meshes error indicators that explicitly combine solution information and mesh shape information are still rare. Examples of such indicators are considered in Section 4 where a case is made for using local error information. Finally in Section 5 a return is made to the motivating problem and it is shown how local error estimates may be combined with advanced visualization techniques to provide the user with a good understanding of the quality of a complex mesh. Much of the material in these notes is based on the work described in [1, 9, 10, 11, 16, 22].

# 2 Mesh Movement in Two Space Dimensions

The idea that it is important for the shape of the elements to reflect local solution behaviour, particularly for highly directional flow problems, is well-known [5, 25]. There are many approaches to generating such meshes. One of the earliest algorithms is that of Gnoffo [19] which has been widely used since then. The algorithm, which is sometimes referred to as Laplacian smoothing, is cheap to implement and can be modified, as described here, to work with an arbitrary error indicator, $w_e$ say, on an element. This method seeks to position the nodes so that the estimated error is evenly distributed across the mesh. However, rather than compute the new mesh globally it moves nodes locally in an iterative fashion. The formula used to compute the updated position $\mathbf{x}_{\mathrm{new}}$ of a node is

$$\mathbf{x}_{\mathrm{new}} = \frac{1}{\sum_{e=1}^{m} w_e} \sum_{e=1}^{m} w_e \mathbf{x}_e \qquad (2)$$

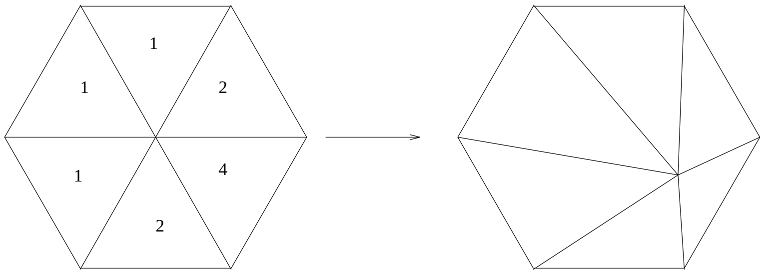



Figure 3: Local node movement, showing error estimates for each element.

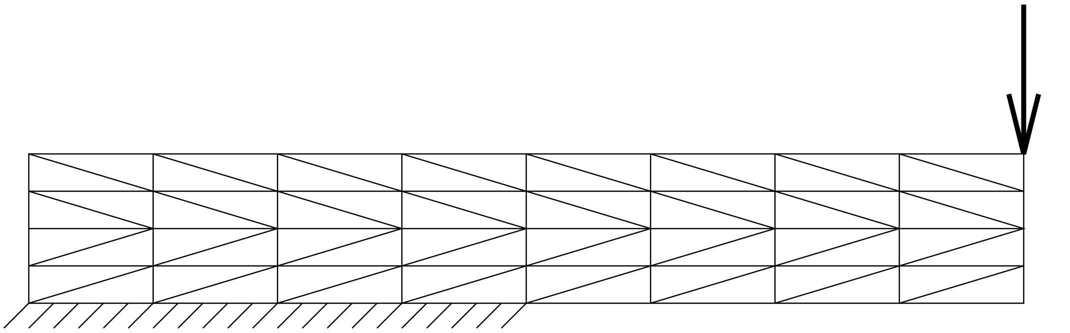


Figure 4: The initial mesh for an overhanging cantilever beam with a vertical concentrated load at the end of the cantilever.

where $m$ is the number of surrounding elements, $\mathbf{x}_1, .., \mathbf{x}_m$ are the centroids of these elements and $w_1, .., w_m$ are the error indicators on the surrounding elements. The result of carrying out this procedure for each node is to place the node at the weighted average of the centroids of the surrounding triangles as shown in Figure 3. In practice, all the new node positions are computed prior to being updated (a Jacobi-type process), rather than in a Gauss-Seidel fashion which is dependent upon the order in which the nodes are updated. In order to avoid mesh tangling only a fraction of the movement indicated by this algorithm is employed in practice, see [11] and Section 3.1 below. Examples of meshes moved by using this algorithm will be shown in Sections 2.5 and 3.1. Although this algorithm gives a simple way of moving the mesh it is natural to ask if it is not possible to combine the calculation of the mesh and the solution in a single unified algorithm.

## 2.1 Moving Finite Element Method

The importance and attraction of the Moving Finite Element method is that it provides such a single unified algorithm for calculating both the mesh and the solution simulataneously. The method was developed by Keith Miller and is described in detail in the monograph of Baines [2]. The method has been used with great success on a number of parabolic, elliptic and hyperbolic equations resulting in some cases in meshes (see [2]) that are highly distorted. One example of such a mesh is given by Jimack, [22], who applies the MFE method to the case of an overhanging cantilever beam with a vertical concentrated load at the end of the cantilever. The initial mesh is uniformly spaced as is shown in Figure 4. The effect of local refinement coupled to the MFE method is shown in Figure 5. The advantage of using the MFE method with coupled h refinement is that not only does the mesh move to minimise the energy but also that h refinement can then further improve the situation. Jimack points out that the stretched mesh gives a signifi-

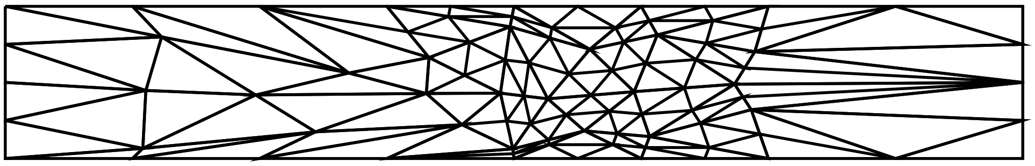


Figure 5: The effect of local $h$-refinement on the steady MFE mesh for the overhanging cantilever beam.

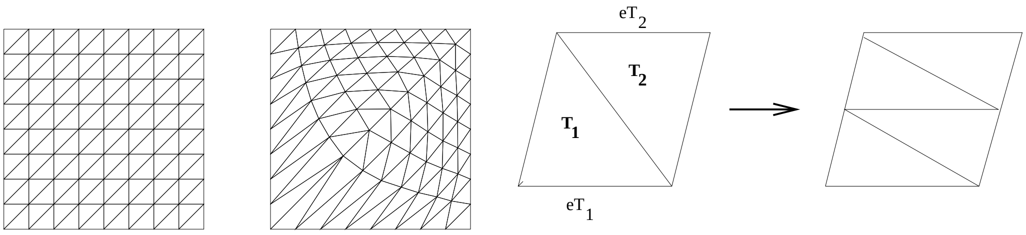



Figure 6: (a) Energy minimisation (b)Blue Refinement

cant improvement in the energy, but that this is achieved at the cost of much extra work over a simple fixed mesh solution. The success of the moving finite element method has, to some extent, been offset by the computational costs incurred and the implementation difficulties that need to be overcome to stop meshes tangling. As a result there have been many attempts to decouple the solution and mesh calculations while still obtaining the stretched meshes characteristic of the method. One such example in Section 2.5 will use the same combination of movement and h refinement.

## 2.2 Energy Minimising Methods

One example of a decoupled mesh and solution algorithm is described by Tourigny and Baines [33], who investigate the construction of locally optimal piecewise polynomial fits to data and produce meshes which vary from smooth to skewed, depending on the solution. The idea dates back as far as Delfour et al. [12], and is further extended by Tourigny and Hulsemann [34], who minimise an energy functional using a Gauss-Siedel method locally to get similarly skewed meshes. Figure 6 shows the original and final meshes for the example used in Section 6.2 of Delfour et al. in which the p.d.e. is given by

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + 56(1 - x - y)^6 = 0 \tag{3}$$

with zero Dirichlet boundary conditions. Rippa and Schiff, [27], present algorithms for constructing minimum energy triangulations by using local operations and also present convincing results to show that these improve the quality of the solution. Jimack [22] shows that the moving finite element method also minimises the error in the energy norm for the class of problems he considered, thus underlining the connection between the methods.

### 2.3 Blue Refinement

Beinert and Kroner [6] move edges so that they are aligned with shock waves and also define a blue directional refinement approach. For example in Figure 6(b) if the edges $eT_1, eT_2$ are parallel and aligned with the flow direction then the pair of triangles is replaced by four anisotropic triangles. Although the indicator used to guide refinement is the gradient of the Mach number rather than an explicit error estimator, the results are nevertheless impressive. A number of convincing demonstrations of the effectiveness of this approach are given by Skalicky and Roos [29] who also show that dramatic improvements in the rate of convergence may be made by using anisotropic mesh refinements.

### 2.4 Roe's Mesh Optimisation Scheme with Edge Swopping

Roe presented a scheme for improving the solution of a two-dimensional convection equation on an unstructured triangular mesh, [28]. The scheme minimises a functional by incrementing solution values and nodal coordinates based on a linear approximation of the solution over the mesh. Consider a scalar variable $u(\mathbf{x})$ on a mesh $K$ of triangular elements. Let $\Phi_k$ be the element residual and $Q_k$ be some positive weighting quantity. A functional is defined by

$$F = \sum_{k \in K} F_k \quad \text{where} \quad F_k = \frac{1}{2}\Phi_k Q_k \Phi_k. \tag{4}$$

An optimisation procedure is then developed based on minimising the functional $F$ by computing the element functional $F_k$ and making adjustments to the unknowns $z_j$ on that element to reduce the functional. Here an increment $\delta z_j$ is defined by following a steepest descent approach, $\delta z_j = -\frac{\partial F_k}{\partial z_j}$ . Roe shows that this process is conservative. These local increments to the nodes of the element can be summed at the nodes to define an update of the mesh and solution in a relaxed Jacobi fashion.

Roe [28] mentions that edge swapping could be used to better align the mesh with the characteristic direction, and so improve the performance of the optimisation scheme. A badly aligned mesh will slow the convergence of the scheme and may even cause the optimisation to fail to converge. Noting from equation (4) that the functional may be decomposed into contributions from each element an alternative edge swapping scheme considers the two possible orientations, shown in Figure 7 at each internal edge of the mesh. Whichever choice leads to the minimum $F^*$ of the maximum of the two functional

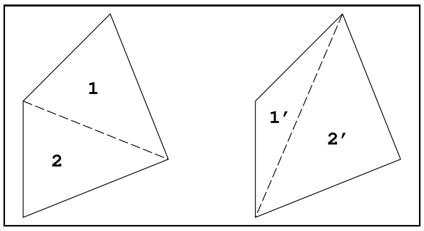



Figure 7: The two possible configurations for edge swapping

values $F$ and $F'$ is chosen, where $F = max(F_1, F_2)$ and $F' = max(F_{1'}, F_{2'})$.

As a numerical example the linear convection equation is solved with constant convection velocity $\mathbf{a} = (2, 1)$ on the domain $(x, y) \in [-1, 1] \times [0, 1]$. At the inflow boundary the continuous solution is prescribed as $u(x) = 0$ for $x \leq -0.8$ or $x \geq 0.2$ and $u(x) = 1$

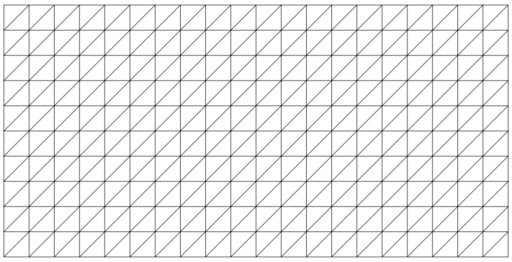
for $-0.4 \leq x \leq -0.2$ with $u$ varying linearly between these values. Figures 8(a)-(b) show the initial mesh, a uniform discretisation of the domain with node spacing 0.1, and the solution obtained if the mesh is kept fixed. Figure 9 shows the graph of the maximum element functional on the mesh and it is clear that this is not converging when only the solution values are updated on this mesh. Figures 8(c)-(d) show the mesh and solution obtained if the previous solution is used as an initial guess but the nodes are also updated using Roe's algorithm. In this case the mesh is becoming aligned with the characteristic direction but the distortion in the mesh prevents an exact representation of the solution. The graph of the maximum element functional shows that the convergence stalls and further improvement of the mesh is not possible. Figures 8(e)-(f) show the mesh and solution obtained if the initial mesh and solution shown in Figures 8(a)-(b) are followed by the edge swapping procedure described above, and then a further stage of Roe's mesh and solution optimisation algorithm. The swapping is implemented by repeatedly sweeping over the whole mesh until no swaps are performed, which in this case requires 5 passes over the mesh. It is seen that this procedure almost perfectly aligns the mesh with the characteristic direction, and that the solution obtained is exact. The convergence of the functional shown in Figure 9 shows that the initial value has been significantly reduced and that the performance of the scheme is greatly enhanced by the swapping. Although the method works well with linear convection problems it has not so far proved straightforward to obtain the same success for nonlinear problems.

### 2.5 Combined h-r Refinement

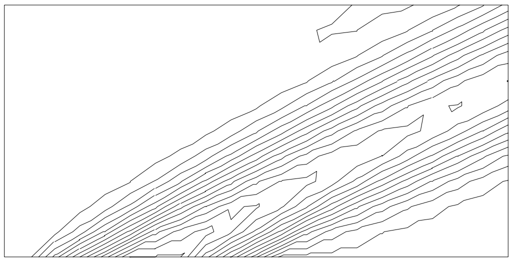
Capon and Jimack [11] employ a Galerkin least-squares finite element method and use a residual based error indicator to guide mesh refinement. Their approach combines Laplacian smoothing mesh movement as described by equation (2) with uniform h refinement of triangles into four. The problems considered by Capon and Jimack are compressible flow calculations over airfoils. The meshes shown in Figures 10 and 11 are taken from an airfoil calculation with a Reynolds number of 106, a ten degree angle of attack and a free stream Mach number of 2 with a Naca0012 airfoil. Figure 10 shows the meshes obtained by solely using h refinement based on (a) density gradient and (b) on the residual as an error indicator. These meshes have 8360 and 8819 elements respectively. Figure 11 shows the final meshes obtained by using coupled h-r refinement with the (a) residual and (b) the density gradient as the error indicator. In this case the meshes have 4660 and 4801 elements respectively. The figures show that using h-r refinement based on the residual provides a smoother mesh that varies more uniformly and has about half the elements compared to the case in which h refinement alone is used. The approach thus seems highly satisfactory in practice.

## 3 Mesh Redistribution and Movement in 3D

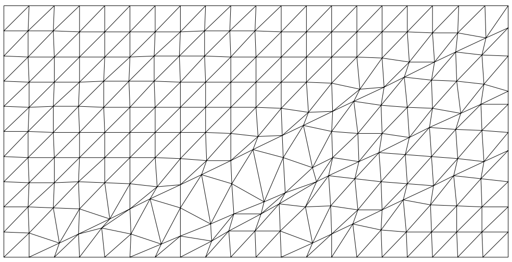
Although there is much work on anisotropic meshes in two space dimensions work on mesh redistribution in 3D is less common. Freitag and Ollivier-Gooch [17] and Iliescu [21] investigate interesting algorithms for splitting tetrahedra. In Iliescu's approach pairs of tetrahedra satisfying convexity and angle conditions related to the flow direction are split into three tetrahedra so as to be aligned with the flow direction, see Figure (12). Freitag and Ollivier-Gooch [17] also provide convincing evidence that mesh smoothing can have beneficial consequences for the rate of convergence of the iterative solver. Tsukerman



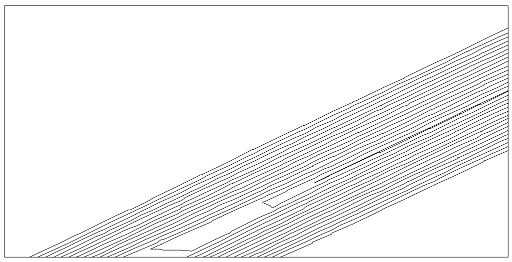
(a) Fixed mesh



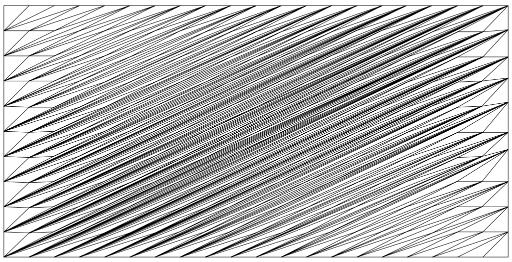
(b) Solution contours



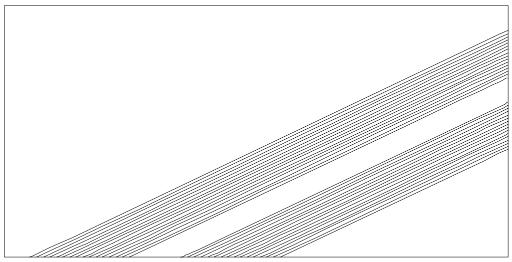
(c) Mesh after node movement



(d) Solution contours



(e) Mesh after edge swapping



(f) Solution contours

Figure 8: Comparison of results with and without edge swapping

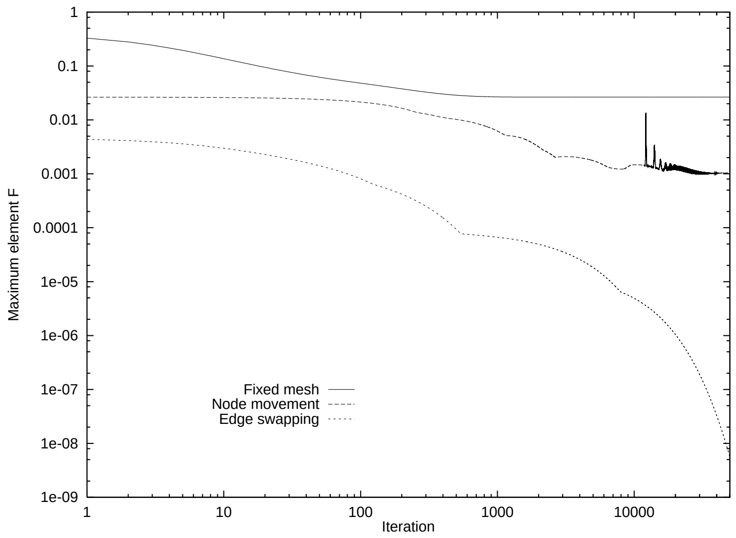



Figure 9: Convergence of the maximum element functional value in each case

[35, 36] derives a condition which shows that it is the maximum eigenvalue of the element stiffness matrix that characterises the impact of the shape of the element on the energy norm of the error of the finite element approximation. Freitag and Knupp [18] use similar information based on the condition number of the linear transformation between a physical tetrahedron and an idealised model tetrahedron. Studies by Freitag and Knupp show that Laplacian smoothing can be detrimental to the quality of tetrahedral meshes on complex geometries. They also show that a smoother based on their new quality measure gives better results than Laplacian smoothing.

## 3.1 A Case Study of 3D Anisotropic Refinement

In this section a simple example will demonstrate the practical use of anisotropic adaptation using an $hr$-refinement scheme in which the nodes are moved according to an edge-based error indicator and compared against standard $h$-refinement results. The test problem considered is the steady 3D hyperbolic equation $\mathbf{a} \cdot \nabla u = 0$, and a standard SUPG finite element method is used, based on an unstructured tetrahedral mesh with linear basis functions $\phi_i(\mathbf{x})$ and test functions $\psi_i(\mathbf{x}) = \phi_i + \tau \mathbf{a} \cdot \nabla \phi_i$ defined at each of the $N_P$ nodes $\mathbf{x}_i$. The parameter $\tau$ is defined as an element quantity, $\tau^K = \alpha h^K / |\mathbf{a}|$ for some measure of the element length $h^K$, taken here to be the minimum element height.

In order to compare the two adaptive algorithms being considered, standard finite element h-refinement, as shown in Figure 13, is driven by solving a problem with a known solution and using the $L^1$ norm of the *exact* error $e(\mathbf{x})$ on each element. The total $L^1$ error, $e$, may then be split into its contributions $e^K$ from each of the $N_E$ elements. To achieve a final $L^1$ error of $e^*$ the error in each cell is reduced to below $e^*/N_E$ using nested isotropic $h$-refinement of an initial unstructured tetrahedral base mesh.

As well as using local mesh refinement, the h-refinement algorithm makes use of a simple node movement scheme designed to steer nodes towards regions of sharp variation in the solution. This is motivated by recent work of Berzins, [8, 9], in which the interpolation error is estimated by assuming that the exact solution can be approximated in a locally quadratic form on each tetrahedral element and then considering the difference

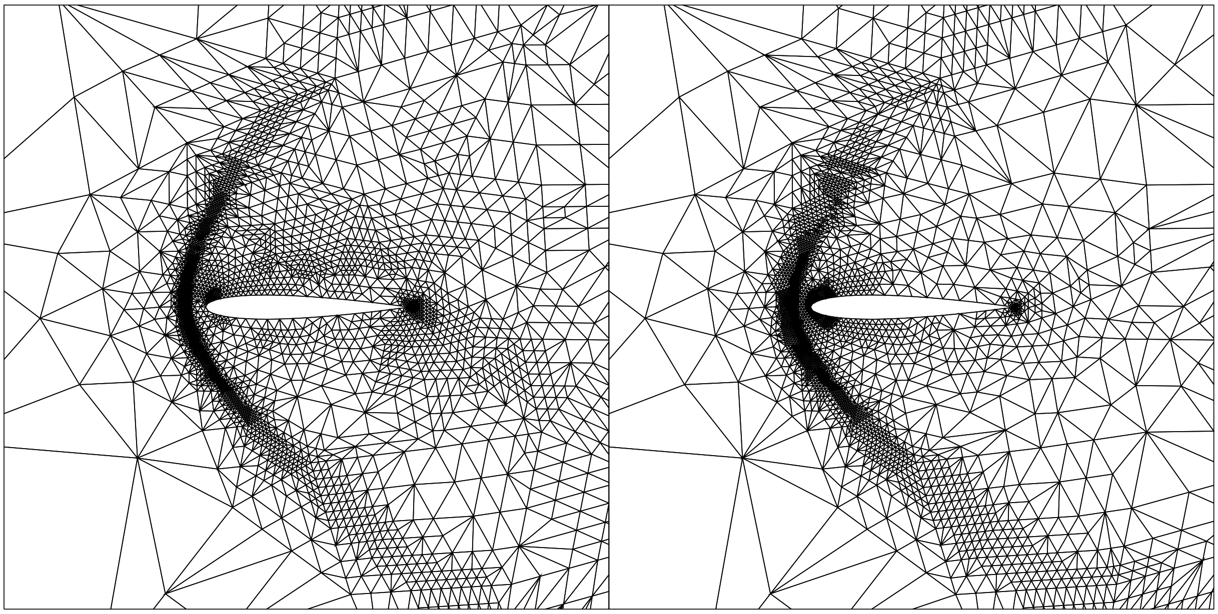


Figure 10: Meshes using h refinement based on (a)density gradients and (b) residuals.

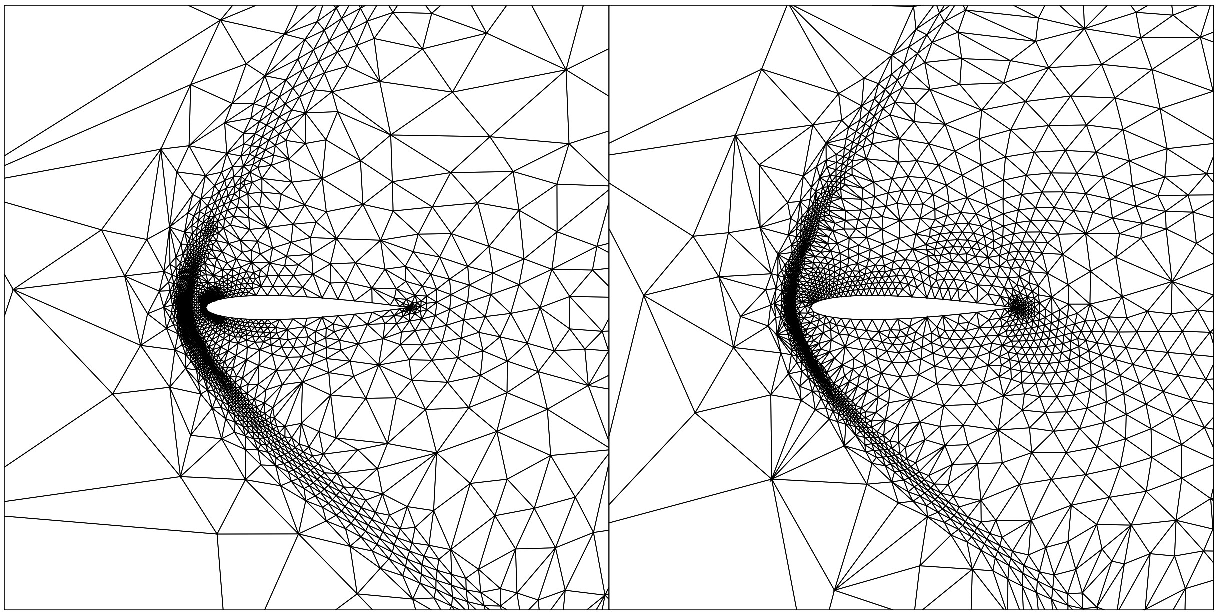


Figure 11: Meshes using h-r refinement based on (a) residuals and (b) density gradients.

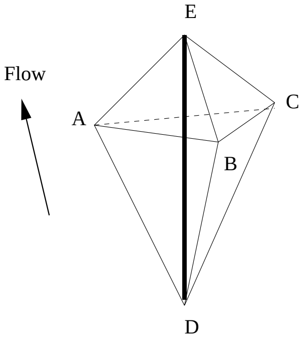



Figure 12: Iliescu's Directional Refinement Procedure.

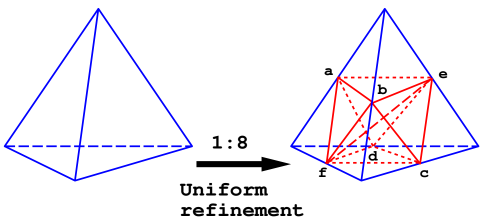



Figure 13: Tetrahedral h refinement

between this quadratic function and the linear finite element interpolant:

$$\int_K e_{lin}^2 \, d\Omega \; = \; \frac{6}{4} V^K \frac{2}{7!} \left( \left( \sum_{s=1}^{6} d_s \right)^2 - d_1 d_4 - d_2 d_5 - d_3 d_6 + \sum_{s=1}^{6} d_s^2 \right) \qquad (5)$$

where $V^K$ is the volume of element $K$. Here, $d_s$ denotes the directed edge second derivative for an edge $s = s(\mathbf{x}_i, \mathbf{x}_j)$ that connects the nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. It is given by

$$d_s := \Big(\nabla u(\mathbf{x}_i) - \nabla u(\mathbf{x}_j)\Big) \cdot (\mathbf{x}_i - \mathbf{x}_j) \cdot L_{ij}^{-2} \qquad \text{with} \qquad L_{ij} := \|\mathbf{x}_i - \mathbf{x}_j\| \, . \qquad (6)$$

The Laplacian smoothing node movement scheme of Section 2 is then used with the edge second derivatives (6) as the weights, e.g.

$$\mathbf{x}_i^{av} \; = \; \frac{\sum_j L_{ij} |d_s| \mathbf{x}_j}{\sum_j L_{ij} |d_s|} \qquad \text{with} \qquad s = s(\mathbf{x}_i, \mathbf{x}_j) \, . \qquad (7)$$

The nodal position is updated by $\mathbf{x}_i \;\rightarrow\; (1-\gamma_i)\mathbf{x}_i + \gamma_i \mathbf{x}_i^{av}$ where $\gamma_i$ is a safety factor at each node $\mathbf{x}_i$ that prevents the mesh from becoming tangled. Several such iterations are performed at each $r$-refinement stage (which is undertaken prior to $h$-refinement).

Further details of the test problem are shown in Figure 14 (a 2D form of the problem is also included to aid comprehension). At the inflow boundary, where $\mathbf{a}\cdot\mathbf{n} < 0$, the imposed solution is defined to have a thin vertical layer across which $u$ varies linearly from 1 to 0. The solution within the domain is therefore defined by the convection of this layer in the direction $\mathbf{a}$, as illustrated. Here, $\mathbf{a} = (2, 1, 1)$ and the layer has thickness 0.025 on a unit cube domain. The initial mesh is a uniform $11^3$ discretisation of the unit cube, with each

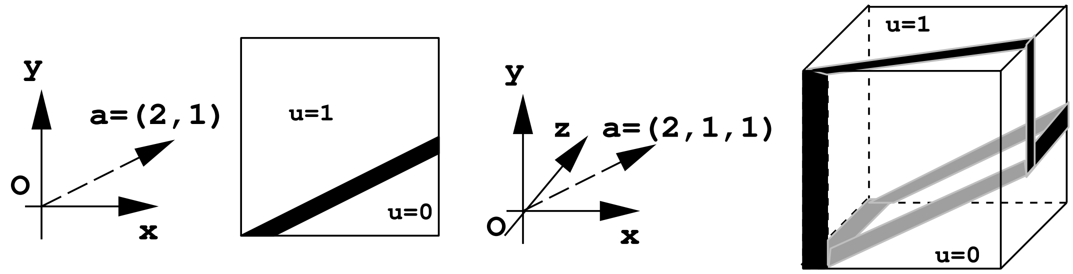



Figure 14: Model steady solution of the 3D convection equation

sub-cube divided into six similar tetrahedra. Figure 15 compares the total error for the $h$- and $hr$-refinement schemes. The $hr$-refinement scheme significantly reduces the error on a given grid, leading a similar level of error with approximately 20000 nodes to that requiring over 100000 nodes using isotropic $h$-refinement. Figure 16 shows two tetrahedra from the final $hr$-refined mesh in the region $(x, y, z) \in [0.3, 0.4] \times [0.7, 0.8] \times [0.1, 0.2]$. It is seen that the upper tetrahedron is aligned with the layer, although large internal angles have been produced. In contrast, the lower tetrahedron is less well-aligned.

The same approach can be further extended in a very simple way with standard 1:8 h-refinement of a tetrahedron shown in Figure 13. In this refinement there is some freedom in that division of each face of a tetrahedron into four leaves a central octahedron volume, with nodes a,b,c,d,e,f as shown in Figure 13, consisting of two pyramids to be decomposed into four tetrahedra. There are three ways of doing this as is shown in Figure 17. It is natural to ask if one of these is to be preferred over the others. The obvious choice is to

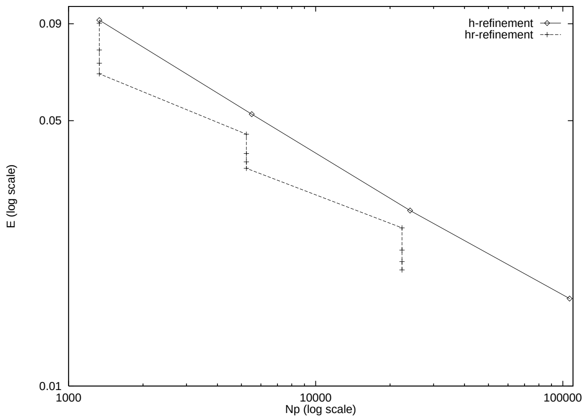



Figure 15: Errors on the adapted grid sequences for $h$- and $hr$-refinement

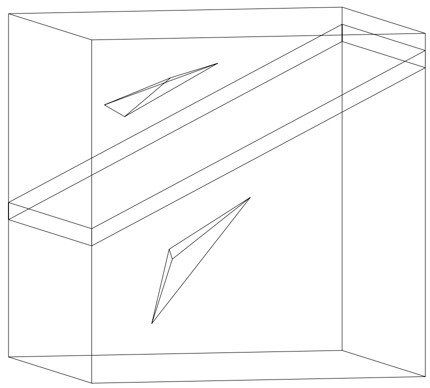


Figure 16: Zoom of the $hr$-refined tetrahedra near the layer

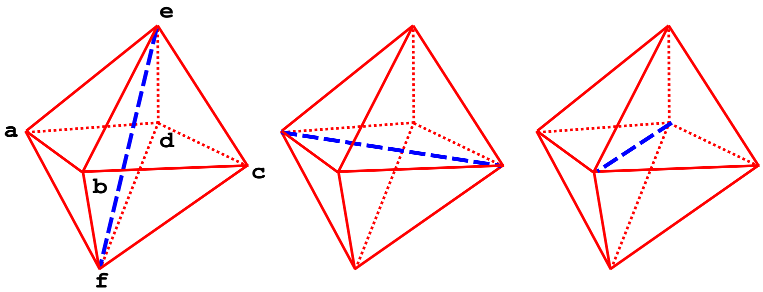



Figure 17: Refinement options for obtaining four tetrahedra from a single octahedron

choose the interior diagonal so that the resulting tetrahedra are aligned with the flow. The alternative is to choose this value geometrically. The graph in Figure 18 shows the effect of both these strategies. As expected aligning the interior edge with the flow direction has immediate benefits.

# 4 Error Estimators with Explicit Geometry Dependence

A common feature of all the methods listed in the previous section is that although the mesh is improved in some sense, the criterion used is only indirectly related to the error. Recent work in error estimation is starting to reveal the explicit dependence of the error on both solution derivatives and on the mesh. This work shows great promise but is still in its early stages. Apel, Kunert and Dobrowolski [1, 13] have developed promising estimates for Poisson-type problems which explicitly indicate mesh and solution dependence.

Bank and Smith [3], in error analysis for the method used in the PLTMG code, show how the error can be written using the second derivatives of the solution along edges, $d_i$, and Bank's mesh quality estimate $q_b$ (see [8] ) as a quotient of solution and geometry information:

$$\int_t | \bigtriangledown e_{lin}(x,y)| dx \approx \frac{d_1^2 + d_2^2 + d_3^2}{q_b}. \tag{8}$$

This somewhat simpler form than the expressions in [9] comes about because Bank and

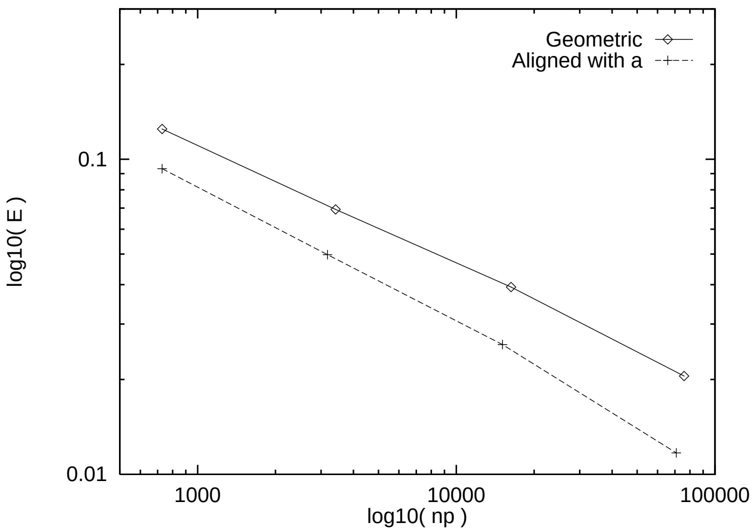



Figure 18: Error vs Number of Points for h-refinement options.

Smith consider only the diagonal terms in a matrix to arrive at their approximation. While this error estimator only applies to steady problems, Lang [25] considers transient problems and explicitly includes both solution derivative and geometry information in the error estimates he derives. For 2D reaction-diffusion p.d.e.s modelling highly-directional phenomena such as flame propagation, Lang proves the error estimate:

$$||e_{lin}(x,y)||^2_{H^1} \quad \leq \quad \tilde{c}\left(\sum_{T\epsilon T_k}\eta_T^2\right)^{1/2} \tag{9}$$

where the local error estimator $\eta_T^2 = C^2(\tau,\lambda,T)\ D_T^2U$ and $D_T^2U$ is a computed approximation to the seminorm $|u|_2$ of the second derivatives of the solution. The constant $C(\tau,\lambda,T)$ is defined by

$$\begin{aligned} C(\tau,\lambda,T) = (1+|\lambda|+\lambda^2)^2h^2[0.2587(1+\frac{1}{\tau})h^2 \\ +\frac{1}{\pi^2}(1+|\lambda|+\lambda^2)] \end{aligned} \tag{10}$$

where $\lambda = tan(\pi/2-\theta)$, $\theta$ is one of the interior angles adjacent to the longest edge of the triangle, $h$ is the longest edge and $\tau$ is the timestep. This estimate thus precisely describes the effect of both the geometry and the solution on the error and enables decisions regarding directional refinement to be taken. The same approach of explicitly defining the relationship between the geometry and the error is also investigated by Barnette [4] in the context of finite difference/volume schemes for simple models of the Navier-Stokes equations.

## 4.1 Finite Volume Methods for Hyperbolic Equations.

The transport part of the atmospheric dispersion problem outlined in Section 1 may be modelled by the simple 3D advection equation

$$U_t + aU_x + bU_y + cU_z = 0. \tag{11}$$

Suppose that the numerical method employed is a first order accurate, conservative cell-centred finite volume scheme. The numerical solution in element $i$ at time $t_n$ is denoted by $u_i^n$, and is an approximation to the exact element averaged volume integral of the solution, [30], over $V_i$ the volume of element $i$, and is usually regarded as being positioned at the element centroid for cell centred schemes. The numerical solution at the next time level $t^{n+1}$ may be written as:

$$u_i^{n+1} = u_i^n - \delta t F_i(t_n, \underline{u}) \quad \text{where} \quad F_i(t_n, \underline{U}) = \frac{1}{V_i} \sum_{k=0}^{3} A_k \mathbf{F}_k . \mathbf{n}_k \tag{12}$$

and where the sum is over the $k$ faces of the element $i$. The $\mathbf{n}_k$ are the outward face unit normal vectors and $A_k$ are the face areas. The fluxes $\mathbf{F}_k$ represent the numerical flux function for each element face, termed the element face fluxes, and are determined by the scheme. In the case of the Godunov scheme these element face numerical fluxes are constructed from the solution of the local element Riemann Problem (RP) at each element face, see [30].

A standard method for choosing the timestep in the numerical solution of p.d.e.s is to use a CFL condition. Although such a condition may ensure stability it may be imprecise as an accuracy control, particularly when complex chemistry source terms are present in the p.d.e. problem. It is important to use an error control which reflects the spatial and temporal contributions to the error incurred. These are the local time integration error, the underlying mesh approximation error, the error in constructing face values using (limited) interpolation and the quadrature errors due to approximating the face integrals and source term contributions. In particular, face orientation in flow problems is critical as error may not be convected through faces aligned with the flow.

While reliable error estimators for finite volume unstructured mesh solvers exist for simple problems, e.g. Kroner and Ohlberger [24], there are no such estimators for problems with complex source terms, such as the atmospheric dispersion problem in Section 1. Consequently for such problems we are forced to rely instead on local error indicators. One such indicator is described by Berzins, [7], who estimates the local-in-time growth of the spatial error by the difference between high and low order computed solutions at the end of a timestep. For problems without source terms the estimate of Kroner and Ohlberger may be adapted to estimate this local-in-time space error. Let $\hat{e}(t)$ be the local-in-time spatial error computed on a timestep then, combining the estimates of Corollary (2.14) of [24] and the ideas of Berzins [7], gives

$$\int\int\int_V \hat{e}(t_{n+1}) d\tau = a \ \delta t \ h^2 \ Q + 2\sqrt{b \ c \ \delta t \ h^2 \ Q} \tag{13}$$

where $a, b, c$ are constants, see [24]. For an evenly spaced mesh with spacing $h$ and timestep $\delta t$, the value of $Q$ is given by

$$Q = \sum_{j \epsilon NT} h|u_j^{n+1} - u_j^n| + L \sum_{E \epsilon NT} (\delta t + h)|u_j^n - u_l^n|, \tag{14}$$

where $L$ is a constant, $u_j^n$ is the solution value associated with the $jth$ tetrahedron out of a mesh of $NT$ tetrahedra with edges $E \epsilon NT$ at time $t_n$. The important feature of this error estimator is that, apart from the constants, the only solution information used consists of solution jumps across faces i.e. $(u_j^n - u_l^n)$ and solution changes in time $(u_j^{n+1} - u_j^n)$ on a particular tetrahedron. It is also easy to see the effect of mesh refinement on the

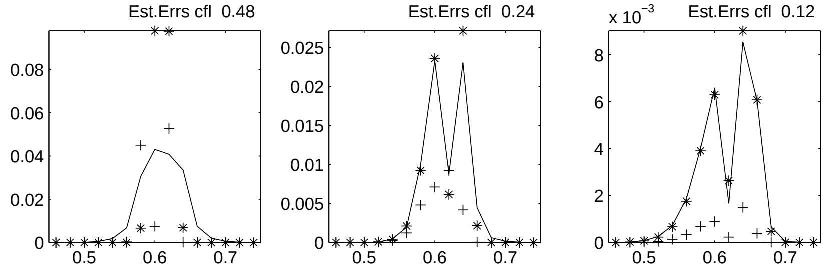



Figure 19: Graphs of local space and time errors for 1D advection.

error estimate. Refinement of two tetrahedra sharing a common face will not change the solution jump between them but will change the multiplying constant.

Although the forward Euler method is being used for timestepping, it is not immediately clear how the error estimate involving the jump in the error over a timestep relates to the standard forward Euler local error estimate, see [7]. The explanation is that the jump in the solution over a timestep provides a bound on the local error over the timestep. Suppose that we have values $u_j^n$ and $u_j^{n+1}$ at times $t_n$ and $t_{n+1}$ . The standard time local error estimate of the forward Euler estimate is $\frac{1}{2}k^2\,\frac{d^2u_j}{dt^2}$. Suppose that we use the standard finite difference approximation, but with an unknown value $u_j^{n+\frac{1}{2}}$ and assume that this value is bounded by the values $u_j^n$ and $u_j^{n+1}$. In other words

$$u_j^{n+\frac{1}{2}} = \alpha\ u_j^n + (1-\alpha)\ u_j^{n+1}, \quad 0 \le \alpha \le 1. \tag{15}$$

Hence

$$u_j^n - 2\ u_j^{n+\frac{1}{2}} +\ u_j^{n+1} \ =\ (1-2\alpha)\ (u_j^n\ -\ u_j^{n+1}) \tag{16}$$

and so in this case

$$|u_j^n - 2\ u_j^{n+\frac{1}{2}} +\ u_j^{n+1}|\ \le\ |(u_j^n\ -\ u_j^{n+1})| \tag{17}$$

and so the change in the solution over the timestep, when multiplied by a constant, provides an upper bound on the estimate of the time local error.

### 4.2 A simple 1D advection equation example

It is of interest to evaluate the error estimation approaches on a similar simple 1D version of Problem 4 (linear advection) in [7]. The local-in-time error is measured about halfway across the domain. Figure 19 shows the spatial distribution of the error $\hat{e}(t)$ with the solid line being the true error and the values * showing the error estimate defined by Berzins [7] and the values + showing the time local error. The peaks in the error graph occur where the scheme smears the top and bottom of the discontinuity. The figure shows that the error estimator does a good job of estimating the structure and the magnitude of the local-in-time spatial error, particularly as the cfl number is reduced, [7]. Table 1 shows the values of the error indicators for different values of the CFL number. The results show that both error estimators do a good job of estimating the L1 norm of the error.

## 5 Visual Mesh Quality Analysis

Although all the above methods make it possible to move the mesh and/or estimate the error there still remains the challenge, especially in three space dimensions of under-

Table 1: Comparison of L1 error norm for error indicators

| CFL Number | 0.96 | 0.48 | 0.24 | 0.12 | 0.06 | 0.03 |
|---|---|---|---|---|---|---|
| True $\hat{e}$ | 1.17e-2 | 3.35e-2 | 1.46e-3 | 6.12e-4 | 2.81e-4 | 1.33e-4 |
| Berzins eqn(6) | 4.53e-2 | 4.18e-2 | 1.42e-3 | 6.23e-4 | 2.73e-4 | 1.26e-4 |
| Kroner eqn(7) | 1.15e-1 | 8.13e-2 | 2.55e-3 | 8.42e-4 | 2.85e-4 | 9.90e-5 |

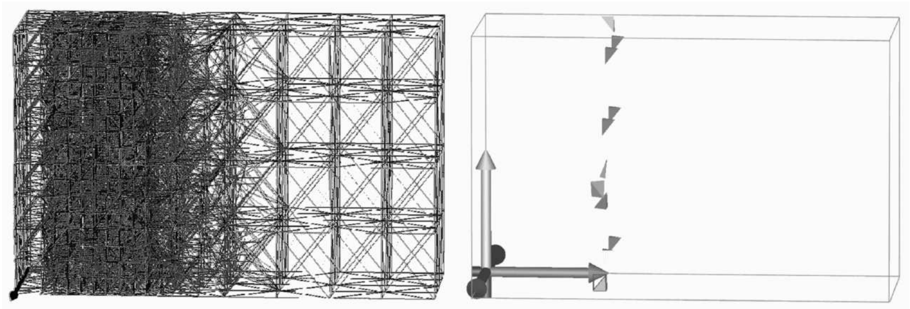


Figure 20: (a) Wire frame mesh and (b) visualization of poor elements

standing precisely how the numerical scheme and mesh quality conspire together with the solution to produce large numerical errors. It is in order to investigate this relationship between the error and the mesh that the user interface developed by Durbeck [14] was used. Durbeck's interface serves as a visual debugger for unstructured tetrahedral mesh solvers. Consider the advection of a simple one-dimensional discontinuity moving from left to right in a 3D channel, as defined by equation (12) with $a = 1, b = c = 0$. A typical example of a 3D unstructured mesh with 16,000 elements is shown in Figure 20a.

The mesh is shown in wire frame, with all the nodes and their attachments shown, and has been refined about the discontinuity. For a mesh such as this Durbeck's system presents analytic information about the mesh geometry and error indicators so that the user can deduce the potential causes for poor quality elements. A view of the mesh, reduced via the debugger to its poorest elements, is shown in Figure 20b. The elements are displayed as solids, with lighting and shading effects. The colour assigned to each tetrahedron corresponds with its relative error indicator value. The poorest elements are roughly aligned and occur near the leading edge of the area refined to represent the discontinuity.

The visual debugger also provides closeups used for analysis of a specific error indicator. The worst element depicted in Figure 20b is shown in closeup view in Figure 21, along with all neighbouring elements which may contribute to its error value. The information presented in this view is intended to correspond closely with the error indicator: in our case, an element's poor quality can be a combination of its shape, orientation and precise vertex locations within the mesh. The same inquiry continues outward to its neighbours and, to a lesser extent, the next level outward as well, as they contribute to the element in question. The worst element and its direct neighbours are displayed as shaded solids and the (less important) next level outward in wire frame. Graphical representations of

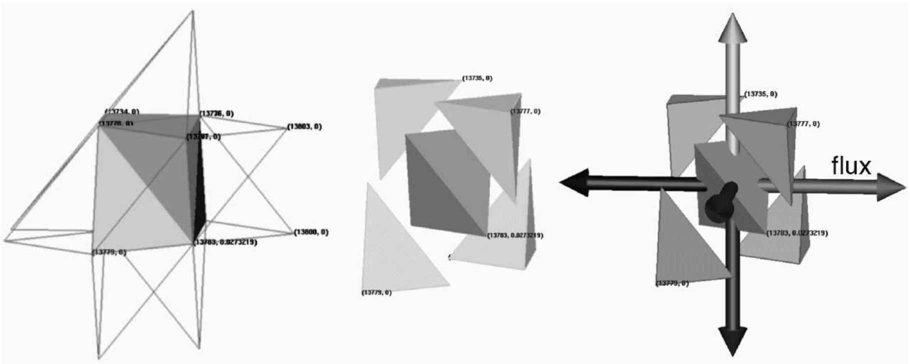



Figure 21: (a) In place (b) exploded (c) rotated closeup
views of worst element and its neighbours

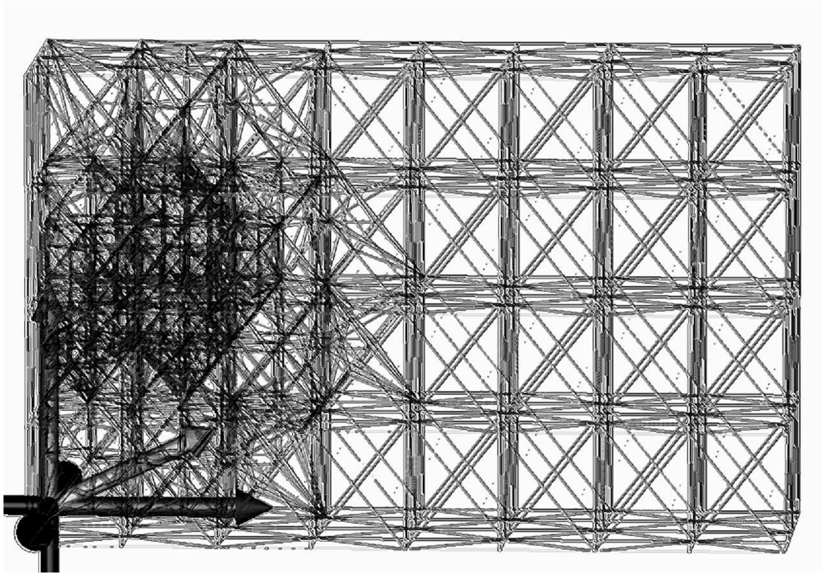


Figure 22: Mesh for the atmospheric model example, displayed as a wire frame model. No error information is displayed. The area of heavy refinement reflects the top of the power station chimney, a point source of several of the chemicals in the model.

each element are annotated with the element number, error indicator value, and solution value. Colour also provides relative error indicator values. As shown in Figure 21b and 21c, the closeup can be rotated about, and exploded outward from the central element in order to better view all the tetrahedra. As seen in Figure 21, the two main contributors to the central element's high error value appear to be its orientation, which causes two faces to be close to perpendicular to the flux, and the wedge shape of the element, which causes these two faces to be relatively wide. Thus the user has been able to easily identify the cause of poor mesh quality in a complex mesh.

Figure 22 shows a SCIRun visualization [23] of the plume developing with the adaptive mesh clustered around the developed portion of the solution. The main area of mesh refinement is along the plume edges close to the chimney. Using the adaptive mesh, we can clearly see the plume edges and can easily identify areas of high concentrations. For this atmospheric diffusion model, the meshes and means of obtaining them are more fully described in [23]. The error indicators were again derived from a simple first-order calculation based on gradients. The same formula (14) of Kroner and Ohlberger [24] was

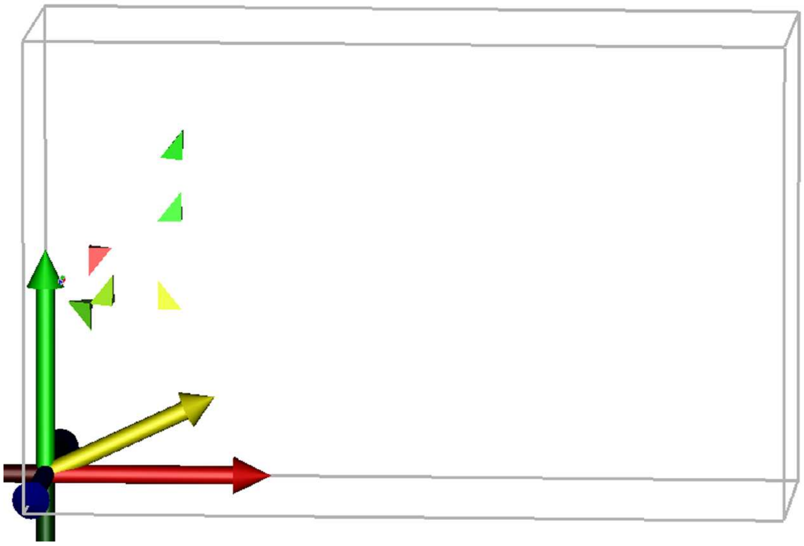


Figure 23: Global view showing the worst elements from the atmospheric model example, as flagged by a specific indicator.

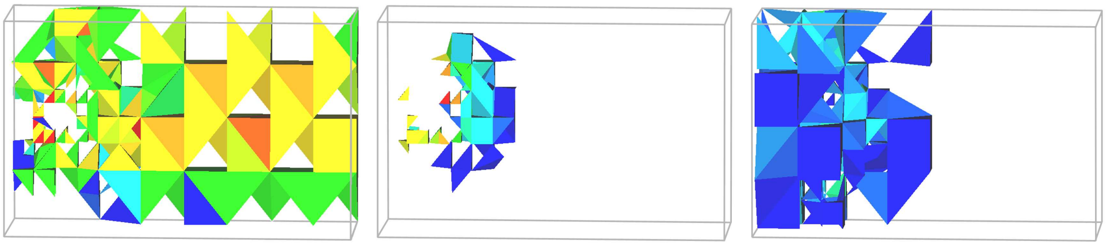


Figure 24: Comparison of the elements flagged by 3 different indicators (for NO, SNGN and a tracer) used for the atmospheric model. In each figure, all elements within 98% of the maximum error value are drawn. The same shading map is used for all ranges. The spatial distribution of error is visibly different, as is the numeric distribution of error values.

applied to obtain separate error indicators for each separate chemical species as well as for a sum of the main NOx species.

The issue of whether the mesh is appropriate for this application is somewhat complex as strong local variations in solution component values make it difficult to assess the quality of the mesh for each component without somehow incorporating solution behaviour. Results are presented in Figure 24 from interactively investigating the differences among the error indicators and from analyzing one particularly bad mesh element.

## 5.1 Analysis of the atmospheric example

Not surprisingly, the set of "bad" elements depends on which species' error indicator is used. The superlative "worst" is conferred onto different elements by the different indicators, as shown in Fig. 23 and 24. Figure 23 shows the "worst" few elements as defined by the union of the "worst" elements flagged by each of the indicators. At a broader level, the distribution of error also varies depending on the error indicator, as shown in Fig 24, which contrasts the sets of imperfect elements from three different indicators, those for NO, SNGN and a simple nonreacting tracer showing the wind direction. Generating these images from the user interface was straightforward, and the results give the user immediate evidence of the effect of choice of indicator. Which indicator to pay attention to is unclear, but the potential impact of ignoring certain chemicals can be hypothesised and tested by going back to the mesh adaptation software and recalculating solution and error information, then visualizing the new results. For instance, it may be beneficial to remesh regions where any of the indicators are large.

One element near the chimney was flagged by several indicators. In this case, this poor quality element is surrounded by good quality elements, as was also true in the simple advection example. This implies that its error is not a simple function of a single face or vertex; otherwise, one neighbour would also have a nonzero error value. Comparison of solution values across the mesh, also visible in the text displayed in the local view, reveals that there is a large jump in the solution value across one face. Abrupt changes in solution component values are expected with this atmospheric model. In that sense, the location of this element puts it at risk. Its volume is above average for the subregion of the mesh around the chimney and in particular is larger than the volume of surrounding elements. The error value associated with this element would obviously be reduced by further refinement. The orientation, shape, and size of the offending face appear to be factors in the solution jump, and therefore factors in this element's poor quality.

# 6 Conclusions

There are a number of conclusions that can be drawn from the work we have described above.

- Complex multicomponent transient p.d.e.s on unstructured meshes pose formidable problems as regards the issue of mesh quality. One such problem area is turbulent combustion which may involve the interaction between many chemical species and complex fluid flows.

- The many moving mesh algorithms and mesh adaptation procedures in two space dimensions provide a wealth of experimental evidence for the effectiveness of stretched meshes.

- Although there is less work in three space dimensions and as yet very few computable error estimates for anisotropic meshes, it is clear that there will be more work in the future. It is also the case that the availability of such error estimates will perhaps always lag behind the problems being solved by practitioners.

- All the evidence also suggests that for problems with anisotropic solutions appropriately stretched meshes will represent the solution better and may also have an effect on the rate of convergence in the iterative solvers employed. At present there are examples of both positive and negative effects and this is clearly an area for further research.

- Good visualization systems are essential in aiding the user's understanding of why a mesh may be poor.

The overall conclusion is that the only really satisfactory approach would seem to be to use an error estimator based on both solution and geometry information to modify the mesh. This would appear to be true for strongly directional fluid flows for which highly distorted meshes appear to be very effective.

## Acknowledgements

Two of the authors (M.B. and L.J.K.D.) would like to thank the SCI group at the University of Utah for their help in many areas as part of the collaboration that has led to [10, 16, 23].

## References

[1] Th Apel, M Berzins, P K Jimack, A Plaks, I Tsukerman, M Walkley, Mesh Shape and Anisotropic Elements: Theory and Practice. To appear in Proceedings of MAFELAP 1999, ed. J.R. Whiteman, 2000.

[2] M.J. Baines, Moving Finite Elements. Oxford Science Publications, 1994.

[3] R.E. Bank and R.K. Smith, Mesh Smoothing Using A Posteriori Error Estimates. SIAM J. Numerical Analysis, 34, 979-997, 1997.

[4] D.W. Barnette, A Mathematical Basis for Automated Structured Grid Generation with Close Coupling to the Flow Solver. Sandia Report SAND97-2382, 1997.

[5] T.J. Barth, Aspects of Unstructured Grids and Finite Volume Solvers for the Euler and Navier Stokes Equations. Lecture Notes Presented at the VKI Lecture Series 1994-95, 1994.

[6] R. Beinert, D. Kroner, Finite Volume Methods with Local Mesh Refinement in 2D, 39-53 in Adaptive Methods-Algorithms Theory and Applications Numerical Methods for Fluid Dynamics Volume 46, Proc of 9th Gamm Seminar Kiel January 22-24. Vieweg, Wiesbaden, 1993.

[7] M. Berzins. Temporal Error Control in the Method of Lines for Convection Dominated Equations, SIAM J. on Sci. Comput., 16, pp.558-580, 1995.

[8] M. Berzins. An Introduction to Mesh Quality. Lecture Notes for Von Karman Institute. March 2000.

[9] M. Berzins. Mesh Quality: A Function of Geometry, Error Estimates or Both? Engineering with Computers (1999) 15:236-247.

[10] M.Berzins, and L.J.K. Durbeck. Mesh Quality for Unstructured Mesh Riemann Solver Based Methods Applied to Hyperbolic PDEs with Source Terms. Proceedings of Godunov Methods: Theory and Applications, Oxford, 18-21 Oct. 1999.

[11] P.C. Capon and P K Jimack, On the Adaptive Finite Element Solution of Partial Differential Equations Using h-r-Refinement. School of Computer Studies, Unviersity of Leeds, Report number 96.13, 1996.

[12] M. Delfour, G. Payre, and J-P Zolesio, An Optimal Triangulation for Second Order Elliptic Problems. Computer Meths. in Applied Mech. and Engnrg, 50, 3, 231-261, 1985.

[13] M.Dobrowolski, S.Graf and C.Pflaum, On a Posteriori Error Estimators in the Finite Element Method on Anisotropic Meshes. Electronic Transactions on Numerical Analysis, Vol. 8, 36-45, 1999.

[14] L.J.K. Durbeck. Contrast Displays: A Haptic and Visual Interface Designed Specifically for Mesh Quality Analysis. M.Sc. Thesis Univ. of Utah, 1999.

[15] L.J.K. Durbeck, Evaporation: a Technique for Visualizing Mesh Quality. Proceedings of the 8th International Meshing Roundtable, South Lake Tahoe, 11-13 October 1999. 259-265.

[16] L.J.K. Durbeck and M.Berzins, Unstructured Mesh Quality Analysis Using an Interactive Haptic and Visual Interface. Submitted to Computing and Visualization in Science, 2000.

[17] L.A. Freitag and C. Ollivier-Gooch, Tetrahedral Mesh Improvement Using Swapping and Smoothing. Int. Jour. fior Numerical Methods in Engineering, 40, 3979-4002, 1997.

[18] L.A. Freitag and P.M. Knupp Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number Proceedings of the 8th International Meshing Roundtable, South Lake Tahoe, 11-13 October 1999. Argonne National Laboratory Report AN/MCS-P767-0699

[19] P.A.Gnoffo, AIAA Journal, 21, No. 9 1249 ,1983.

[20] T. Iliescu, A Flow-Aligning Algorithm for Convection-Dominated Problems, Int. J. Numer. Meths. in Engng, 46, 993-1000, 1999.

[21] T. Iliescu, A 3D Flow-Aligning Algorithm for Convection-Dominated Problems, Accepted by Applied Math. Letters, 1999.

[22] P.K. Jimack, An Optimal Finite Element Mesh for Elastostatic Structural Analysis Problems, Computers and Structures, vol.64, pp.192-208, 1997.

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

[24] D. Kroner and M. Ohlberger A Posteriori Error Estimates for Upwind Finite Volume Schemes for Nonlinear Conservation Laws in Multi-Dimensions. Mathematics of Computation, [to appear]

[25] J. Lang, Two-Dimensional Fully Adaptive Solutions of Reaction-Diffusion Equations. Appl. Numer. Math. 18, 223-240, 1995.

[26] J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz, Adaptive Remeshing for Compressible Flow Calculations, J. of Comp. Physics, 22, 131-149, 1986.

[27] S. Rippa and B. Schiff, Minimum Energy triangulations for elliptic problems. Comput Meths. in Appl. Mech and Engnrg 84, 257-274, 1990.

[28] P. Roe, Compounded of Many Simples. Reflections on the Role of Model Problems in CFD, Barriers and Challenges in Computational Fluid Dynamics, Kluwer Academic, 241-258, 1998

[29] T.Skalicky and H-G. Roos, Anisotropic mesh refinement for problems with internal and boundary layers. Int. Jour. for Numer. Meths in Eng. 46, 1933-1953, 1999.

[30] W.Speares and M. Berzins A 3D Unstructured Mesh Adaptation Algorithm for Time-Dependent Shock-dominated Problems. International Journal for Numerical Methods in Fluids 25 (1997): 81-104.

[31] A.S. Tomlin, M. Berzins, J.M. Ware, J. Smith and M Pilling, On the Use of Adaptive Gridding Methods for Modelling Chemical Transport from multi-scale sources. Atmospheric Env. Vol. 31 (18) 2945-2959, 1998.

[32] A.S.Tomlin, S. Ghorai, G. Hart, and M. Berzins. "The Use of 3-D Adaptive Unstructured Meshes in Air Pollution Modelling." Proceedings of Nato Workshop on Air Pollution Modelling, Sofia,1998. Eds. Z.Zlatev et al. Large Scale Computations in Pollution Modelling, Kluwer Academic Publishers, 1998. 339-348.

[33] Y.Tourigny, and M.J. Baines, Analysis of an Algorithm for Generating Locally Optimal Meshes for L2 Approximation by Discontinuous Piecewise Polynomials. Math. Comp. 66, 218, 623-650, 1997.

[34] Y.Tourigny and F. Hulsemann, A New Moving Mesh Algorithm for the Finite Element Solution of Variational Problems, SIAM J. Numer. Anal, 35, 4, 1416-1438, 1997.

[35] I. Tsukerman, A General Accuracy Criterion for Finite Element Approximation. IEEE Transactions on Magnetics, 34, 5, September 1998.

[36] I. Tsukerman, Comparison of Accuracy Criteria for Approximation of Conservative Fields on Tetrahedra. IEEE Transactions on Magnetics, 34, 5, September 1998.

[37] N.P. Weatherill, M.J. Marchant, O. Hassan, Unstructured Grid Generation and Adaptation for a transport Aircraft Configuration, Paper presented at 1993 European Forum on Recent Developments and Applications in Aeronautical Computational Fluid Dynamics. Held at Bristol UK 1-3 September 1993.