Published paper
Mahmood, R. and Jimack, P.K. (2004) *Locally optimal unstructured finite element meshes in 3 dimensions.* Computers & Structures, 82 (23-26). pp. 2105-2116.

Locally Optimal Unstructured Finite Element Meshes in 3 Dimensions

Rashid Mahmood and Peter K. Jimack

School of Computing, University of Leeds, Leeds LS2 9JT, UK.

Abstract

This paper investigates the adaptive finite element solution of a general class of variational problems in three dimensions using a combination of node movement, edge swapping, face swapping and node insertion. The adaptive strategy proposed is a generalization of previous work in two dimensions and is based upon the construction of a hierarchy of locally optimal meshes. Results presented, both for a single equation and a system of coupled equations, suggest that this approach is able to produce better meshes of tetrahedra than those obtained by more conventional adaptive strategies and in a relatively efficient manner.

Keywords: finite elements, variational problems, mesh optimization, tetrahedral elements, node movement, edge swapping, node insertion.

1 Introduction

In this paper we present an extension of our previous work on mesh optimization, presented in [7, 8], from two space dimensions to three. The approach that we follow is to consider the adaptive finite element solution of a general class of variational problems using a combination of node movement, edge swapping, face swapping and node insertion. The particular adaptive scheme that is used is based upon the construction of a hierarchy of locally optimal tetrahedral

meshes starting with a coarse grid for which the location and connectivity of the nodes is optimized. This grid is then locally refined and the new mesh is optimized in the same manner.

The class of problem that we consider in this work may be posed in the following form (or similar, according to the precise nature of the boundary conditions):

$$\min_{u:\Omega(\subset\mathcal{R}^m)\to\mathcal{R}^n} \int_\Omega F(\underline{x}, u, \underline{\nabla} u) d\underline{x} \tag{1}$$

for some energy density function $F : \mathcal{R}^m \times \mathcal{R}^n \times \mathcal{R}^{m\times n} \to \mathcal{R}$. Here m is the dimension of the problem and n is the dimension of the dependent variable u. Physically this variational form may be used to model problems in linear and nonlinear elasticity, heat and electrical conduction, motion by mean curvature and many more. Throughout this paper we restrict our attention to the three-dimensional case where $m = 3$.

For variational problems of the form (1), the fact that the exact solution minimizes the energy functional provides a natural optimality criterion for the design of computational grids using r-refinement (defined here to include both node relocation and mesh reconnection). Indeed, the idea of locally minimising the energy with respect to the location of the vertices of a mesh of fixed topology has been considered by a number of authors (e.g. [2],[16]), as has the approach of locally minimising the energy with respect to the connectivity of a mesh with fixed vertices (e.g. [14]). All of this work has been undertaken in only two space dimensions however and, to our knowledge, this is the first work in which mesh optimization with respect to the solution energy has been attempted for unstructured tetrahedral meshes in three space dimensions.

The algorithm that we use consists of a number of sweeps through each of the nodes in turn until convergence is achieved. At the beginning of each sweep the gradient, with respect to the position of each node, of the energy functional

$$E = \int_\Omega F(\underline{x}, u^h, \underline{\nabla} u^h) d\underline{x} \tag{2}$$

is found (where u^h is the latest piecewise linear finite element solution). When each node is visited the direction of steepest decent is used in order to determine along which line the node

should be moved. The distance that the node is moved along this line is computed using a one-dimensional constrained minimization of (2), and once this new position for the node has been found the value of the solution at that node is updated by solving a local problem. Once this update is complete the same process is undertaken for the next node and when each node has been visited the sweep is complete. Provided convergence has not been achieved the next sweep may then begin.

Once convergence with respect to the position of each node has been achieved a further reduction in the energy of the solution is sought by the use of edge and face swapping. In three dimensions there are a large number of different ways in which the local connectivity of the nodes may be altered, see for example [3, 5, 9, 10]. In this work we use the same edge and face swapping stencils as [3, 4], whose work is restricted to improving the geometric quality of the mesh rather than minimizing energy as we do here.

Of course the positions of the nodes are likely to be no longer locally optimal at this point due to the edge/face swapping. Hence it is necessary to alternate between the node movement and the swapping algorithms until the whole process has converged (at least approximately). At this stage we allow the application of local mesh refinement to obtain a new mesh at the next level which must itself now be optimized. The process is complete when either a desired accuracy has been obtained or a maximum number of nodes or elements has been reached. Figure 1 illustrates the overall algorithm proposed.

2 Node Movement

A necessary condition for the position of each node of the tetrahedral mesh to be optimal is that the derivative of the energy functional with respect to each nodal position is zero. Like the approaches of [7, 16] our algorithm seeks to reduce the energy functional monotonically by moving each node in turn until the derivative with respect to the position of each node is zero. Whilst this does not guarantee with absolute certainty that a local minimum (as opposed to a saddle point or a local maximum) is reached, the presence of rounding errors combined

```
Stop = false
repeat
    repeat
        undertake node optimization
        undertake connectivity optimization
    until converged
    if (accuracy satisfactory) or (maximum mesh size reached) then
        Stop = true
    else
        refine mesh
        solve discrete problem on new mesh
    end if
until Stop
```

Figure 1: Overview of proposed mesh optimization algorithm for the finite element solution of (1).

with the downhill nature of the technique ensures that in practise any other outcome is almost impossible.

As indicated above the node optimization phase of the overall algorithm in Figure 1 consists of a number of sweeps through each of the nodes in turn until convergence is achieved. At the beginning of each sweep the gradient, with respect to the position of each node, of the energy functional (2) is found. This is done using the same approach as described in [7], based upon [6]. In [6] it is proved that if $\underline{s_i}$ is the position vector of node i then

$$\frac{\partial E}{\partial s_{id}} = \int_\Omega \left\{ \left[F\delta_{dj} - \frac{\partial u_k^h}{\partial x_d} F_{,3kj} \right] \frac{\partial \alpha_i}{\partial x_j} + F_{,1d}\alpha_i \right\} dx \ , \tag{3}$$

where α_i is the usual local piecewise linear basis function at node i, s_{id} is the d^{th} component of $\underline{s_i}$ ($d = 1$ to m), $F_{,p}$ represents the derivative of F with respect to its p^{th} argument, other suffices represent components of tensors, δ_{dj} is the Kronecker delta and repeated suffices imply summation ($j = 1$ to m and $k = 1$ to n). Note that using (3) the gradients with respect to all of the vertices in the mesh may be assembled in a single pass of the elements. These gradients are then sorted by magnitude and the nodes visited one at a time, starting with the largest gradient.

When each node is visited the direction of steepest descent,

$$\underline{s} = -\frac{\partial E}{\partial \underline{s_i}} \ , \tag{4}$$

is used in order to determine along which line the node should be moved. The distance that the node is moved along this line is computed using a one-dimensional minimization of the energy subject to the constraint that the node should not move more than a proportion w ($0 < w < 1$) of the distance from its initial position to its nearest neighbour. Once a new position for the node has been found the value of the solution, u_i say, at that node must be updated by solving the local problem

$$\min_{u_i \in \mathcal{R}^n} \int_{\Omega_i} F(\underline{x}, u^h, \underline{\nabla} u^h) d\underline{x} \ . \tag{5}$$

Here Ω_i is the union of all elements which have node i as a vertex and Dirichlet conditions are imposed on $\partial\Omega_i$ using the latest values for u^h. All nodes in the sorted list (based upon the magnitude of the gradient in (4)) are updated in this way in turn in order to complete a

single sweep of the node optimization step. A number of sweeps are generally taken in order to converge, at least approximately, to an optimal solution. It should be noted that in general this simple steepest decent approach will not yield the best possible speed of convergence for the node movement phase, however our purpose here is to demonstrate the effectiveness of the overall algorithm rather than focus on obtaining the most efficient possible implementation. Modifications for more sophisticated gradient-based optimization schemes could easily be made without altering the underlying technique.

Using the above approach the interior nodes may move in any direction however a slight modification is required for nodes on the boundary of Ω. These nodes may only be moved tangentially along the boundary and even then this is subject to the constraint that the domain remains unaltered. Where this constraint is not violated the downhill direction of motion along the boundary is easily computed by projecting $\underline{s}$ from (4) onto the local tangent of the boundary. The one-dimensional minimization in this direction is then completed as for any other node. On Dirichlet boundaries the updated value of u is of course prescribed however on any other type of boundary it must be computed by solving a local problem of the same form as (5). In the implementation described here only planar boundaries have been considered. The extension to curved boundaries could most easily be achieved by treating the boundary as being locally flat (using the tangent plane for the boundary node being optimized for example) and then projecting the updated position in the plane onto the true boundary.

3 Optimizing Connectivity

In three dimensions tetrahedral mesh connectivities may be altered either by undertaking so-called *edge swaps* or *face swaps*. In this work we make use of both of these techniques by exploiting their implementation within the GRUMMP software package, described in [3, 4]. This software seeks to optimize three-dimensional mesh connectivity based upon geometric criteria such as angle conditions and similar qualitative mesh quality measures. Since the source code is publicly available it is possible to modify this in order to undertake optimization

of the mesh connectivity based upon our own criteria: specifically minimization of the energy functional (2) on the patches of elements surrounding an edge or a face respectively. The two algorithms used for edge and face swapping are now briefly described.

3.1 Edge Swapping

Edge swapping in three dimensions is not really a swap but a removal of an edge followed by its replacement by one, two or many edges depending upon how many elements surround that edge (see Figure 2 for example). Edge swapping reconfigures the E tetrahedra incident on an edge of the mesh by removing that edge and replacing these E tetrahedra by $2E-4$ new tetrahedra. As an example, consider an initial configuration with five tetrahedra incident to an edge. The left side of Figure 2 shows five tetrahedra incident to an edge OP and the right side shows one possible reconfiguration of this sub-mesh into six tetrahedra. This new configuration is specified by defining three "equatorial triangles", i.e. which are not incident on either of vertices O and P. In Figure 2 these triangles are $\triangle 124$, $\triangle 234$ and $\triangle 145$. There are four other possible configurations for this case (each corresponding to a different set of equatorial triangles), which can be obtained by rotating the interior triangle in Figure 2. As edge swapping replaces E original tetrahedra into $2E-4$ tetrahedra, when $E > 4$ more elements are created than are removed. For all of the figures in this section solid lines are used to show the front view of the diagram, lines with dashes show the back of the diagram and dotted lines are used in the interior of the convex hull of the points.

In addition, the number of possible ways that elements can be reconnected after deleting an edge increases with E and is given by

$$C_r = \frac{(2E-4)!}{(E-1)!(E-2)!} \tag{6}$$

(see [5]). When $E = 5$ this gives the five possibilities noted in the previous paragraph. However, as E grows the number of possible configurations grows very rapidly and so, following [3, 4], only edges with $E < 8$ are considered as candidates for edge swapping. The possible configurations for $4 \leq E \leq 7$ are shown diagrammatically in Figure 3, where equatorial

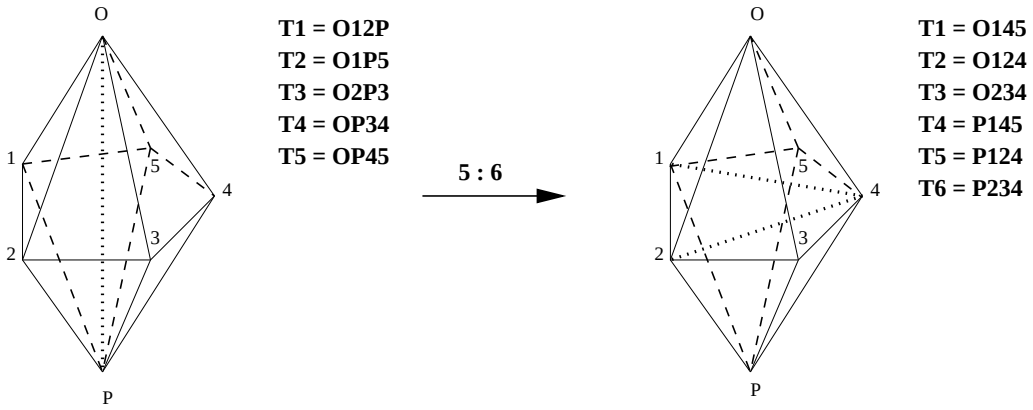

Figure 2: Edge swapping for 5 tetrahedra to 6, where edge OP is surrounded by 5 tetrahedra.

triangles are shown along with the number of unique rotations for each configuration. An optimization method therefore has to search through a large number of connectivity permutations for large E in order to determine which reconfiguration of the original E tetrahedra has the lowest energy. For this it is necessary to compute the energy for each tetrahedron in each configuration. Fortunately, when E is large, the number of unique tetrahedra is much smaller than the number of configurations times the number of tetrahedra since many tetrahedra appear in more than one configuration. This is shown in Table 1 (taken from [3]) and means that the cost of performing a local mesh optimization is not quite as high as (6) initially suggests.

Tets before	Tets after	Configurations	Tets $\times$ configs	Unique tets
4	4	2	8	8
5	6	5	30	20
6	8	14	112	40
7	10	42	420	70

Table 1: Number of unique tetrahedra and possible configurations for edge swapping (taken from [3]).

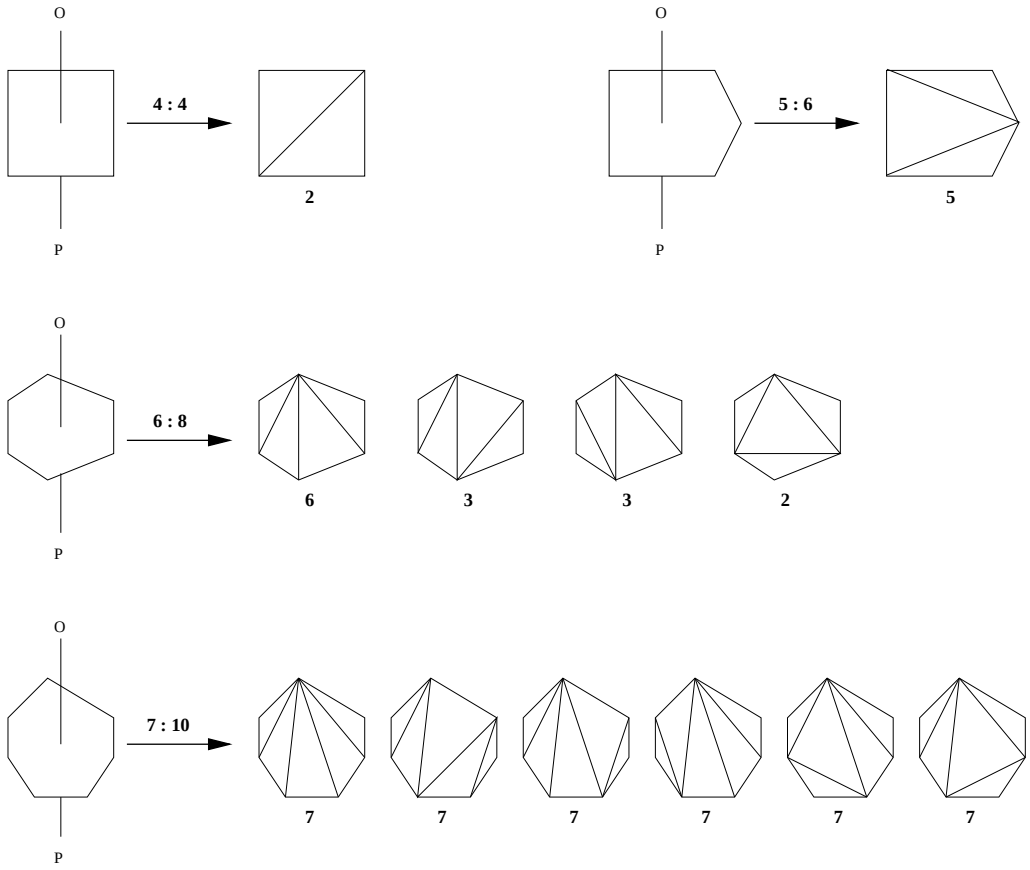

Figure 3: Equatorial triangles after swapping edge OP, surrounded by 4,5,6 and 7 tetrahedra, including the number of unique rotations for each configuration shown.

3.2 Face Swapping

Face swapping is cheaper to execute, although possibly more complicated to implement, than edge swapping in three dimensions. It is based upon the possible configurations of sets of five distinct non-coplanar points [9, 11] (since each interior face in a tetrahedral mesh separates two tetrahedra, which contain a total of five points between them). Five such configurations may arise, as described below and illustrated in Figures 4 and 5.

1. No four of the five points are coplanar and none of the points is in the interior of the convex hull of the other four. In this case the five points may be connected as either two tetrahedra (denoted as configuration 1A) or three tetrahedra (denoted as configuration 1B). This is the most common configuration to arise and both types of connectivity are

illustrated in Figure 4.

2. No four of the five points are coplanar however one of the points lies in the interior of the convex hull of the other four. In this case the five points may be connected uniquely into four tetrahedra, which each have the interior point as a vertex. This is illustrated in Figure 4 where point B in configuration 2 is the interior vertex.

3. Four of the five points are coplanar and none of these four points lies inside the convex hull of the other three. In this case the five points may be connected as two tetrahedra in two different ways (denoted here as configurations 3A and 3B respectively). These two possible connectivities are shown in Figure 5.

4. Three of the five points are colinear. In this case the five points may be connected uniquely as two tetrahedra, as shown in Figure 5 (configuration 4).

5. Four of the five points are coplanar and one of these four points lies inside the convex hull of the other three. In this case the five points may be connected uniquely as three tetrahedra. This is illustrated in Figure 5 where point B in configuration 5 is in the plane formed by ACD and in the interior of their convex hull.

It should be noted that face swapping is only possible for those sets of five points which are in configurations 1 or 3.

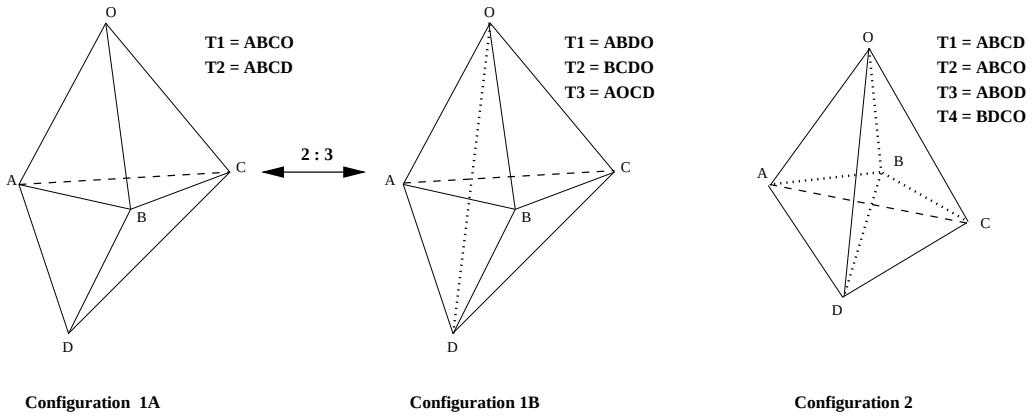

Figure 4: Possible configurations of five points where no four of the five points are coplanar.

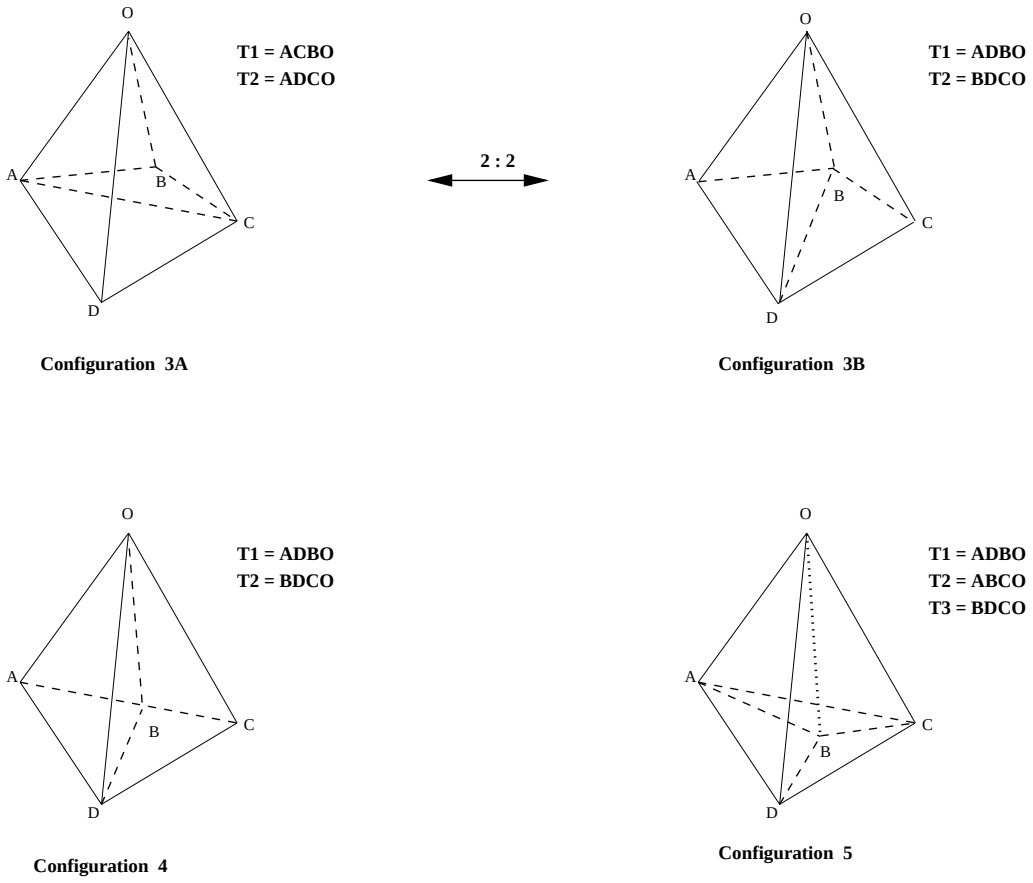

Figure 5: Possible configurations of five points where four of the five points are coplanar.

Unlike with edge swapping, where many possible reconfigurations are possible, if a face swap is possible (configurations 1 and 3 in Figures 4 and 5 respectively) then only two possible choices need to be compared. This allows a simple and quick comparison to find the one with the lower energy. Details of the way in which the face swapping can be implemented in practise can be found in [10, 11]. In [3, 4] face swapping is the primary algorithm for reconnecting the mesh and edge swapping is used as a supplement to it. The edge swapping routines are also used as part of a separate procedure specifically designed to remove poor quality tetrahedra but we do not make use of this procedure in this work since we are motivated only by energy reduction.

4 Node Insertion

The main difficulty with the node movement and edge/face swapping strategies above is that it is impossible to know *a priori* how many nodes or elements will be required in order to get a sufficiently accurate finite element solution to any given variational problem. Even an optimal mesh with a given number of nodes may not be adequate for obtaining a solution of a desired accuracy. For this reason some form of mesh refinement is essential.

In this work we use the regular refinement algorithm implemented in [15]. This divides each tetrahedral element that is to be refined into eight children by introducing nodes at the mid-points of each edge. Each new node is then connected to the other two new nodes lying on each face as illustrated in Figure 6. The three new edges on each face may be seen to cut off four child elements at the corners of the parent tetrahedron, leaving an octahedron at the centre. This may be divided into four more child tetrahedra by adding a further edge (LJ in Figure 6) connecting two opposite vertices. The choice of which internal diagonal to insert is important: the approach used in [15] is to choose the longest one but other approaches are possible (see, for example, [13]). It should be noted that this refinement technique produces child tetrahedra that are of different shapes to their parent, which may be an issue for some mesh generators. This is not an issue for this algorithm however since we are not concerned with geometric mesh quality and since both node movement and edge/face swapping are also used anyway.

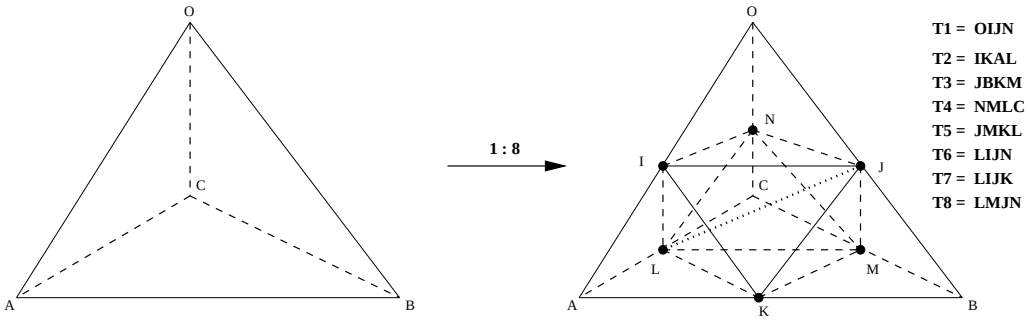

Figure 6: Regular refinement of a tetrahedron into 8 child tetrahedra, by bisecting all of the edges.

For the results that are presented in the following section both global and local refinement examples are included. In the former case the regular refinement algorithm alone is sufficient however, when local mesh refinement is used, an additional refinement scheme is required to deal with the *hanging nodes* that are left on an unrefined element which has one or more neighbour that has been refined. In [15] these cases are dealt with through the use of a number of so-called *green refinement* stencils which deal with elements that have one or more hanging node.

5 Numerical Results

In this section we consider two example problems of the form (1). The first of these is a single equation (i.e. $n = 1$), and the second of these is a system for which $n = m = 3$.

5.1 Problem One

For an initial test problem we consider the following equation:

$$-\Delta u + \frac{1}{\varepsilon^2} u = 0 \,, \qquad \underline{x} \in \Omega = (0,1) \times (0,1) \times (0,1) \,, \tag{7}$$

subject to the Dirichlet boundary conditions

$$u = e^{-x_1/\varepsilon} \tag{8}$$

throughout $\partial\Omega$. This is chosen so that (8) is the exact solution of (7) throughout Ω. Hence, for any given value of ε the analytic solution, and therefore the true energy minimum, are both known (in this case $\varepsilon = 0.01$ is chosen and the optimal value for the energy is $E = 50.0000$).

Following the approach used in [7] for testing the two-dimensional algorithm, we begin by assessing the performance of three-dimensional multilevel mesh optimization when combined with global h-refinement. Initially the test problem is solved on a regular coarse grid of 384 tetrahedral elements. This mesh is then optimized locally using node movement and edge/face swapping and the total energy of the solution reduces from 378.62763 to 62.113265. However

the number of elements increases from 384 to 407 due to the application of edge/face swapping. Three levels of uniform refinement, each followed by optimization, then yield solutions with energies of 51.223148, 50.200687 and 50.048211 on meshes of 3330, 27346 and 220769 elements respectively. For each of these three levels the number of elements increased by slightly more than a factor of eight due to the edge/face swapping.

To see that this final mesh is superior to one obtained without multilevel optimization the problem is then solved on a three level uniform refinement of the initial mesh, (with 196608 elements therefore), to yield a solution with energy 67.278957. When this mesh is optimized however the energy only decreases to a value of 52.338504, with an increase in the number of elements to 197070 due to edge/face swapping.

We now demonstrate the potential advantages of using local refinement with the multilevel optimization. Starting with the locally optimal 384 element grid, a sequence of three further meshes is obtained through local h-refinement (by refining those elements whose local energy exceeded 60% of the maximum local energy on any element) followed by local optimization. These meshes contain 2931, 18741 and 110170 tetrahedral elements and the corresponding solutions have energies of 51.226773, 50.200292 and 50.043149 respectively.

Finally, we demonstrate the superiority of this final mesh over one obtained using only local h-refinement followed by local optimization at the end. This comes from the observation that a grid of 232140 elements obtained using only local h-refinement yields a solution energy of 54.813215 and, when this is optimized, the solution energy only reduces to 51.443760. A summary of all of these computational results is provided in Table 2 and an illustration of the meshes obtained using multilevel optimization with local h-refinement is given in Figure 7.

5.2 Problem Two

The second problem that we consider involves the calculation of the displacement field for a three dimensional linear elastic model of an overhanging cantilever beam with domain

$$\Omega = \{(x, y, z) : 0 < x < 4, 0 < y < 1, 0 < z < 1\}.$$

Elements	Energy	Description
384	378.62763	Initial mesh.
407	62.113265	
3330	51.223148	Multilevel optimization and
27346	50.200687	global h-refinement.
220769	50.048211	
196608	67.278957	Global h-refinement followed
197070	52.338504	by optimization.
407	62.113263	
2931	51.226773	Multilevel optimization and
18741	50.200292	local h-refinement.
110170	50.043149	
232140	54.813215	Local h-refinement followed
233506	51.443760	by optimization.

Table 2: Summary of the results obtained for the first test problem (the global energy minimum is 50.0000).

The bottom half of the beam is fixed as illustrated by the shaded region in Figure 8 and the energy functional is given by,

$$E = \frac{1}{2}\int_\Omega \frac{\partial u_i}{\partial x_j} C_{ijk\ell} \frac{\partial u_k}{\partial x_\ell}\, d\underline{x} - \int_\Omega \rho b_i u_i \, d\underline{x} \,. \tag{9}$$

Here, all repeated suffices are summed from 1 to 3, $\mathbf{C}$ is the usual fourth order elasticity tensor, chosen to correspond to an isotropic material with a non-dimensionalized Young's modulus $E = 100$ and a Poisson ratio $\nu = 0.001$, $\rho\underline{b}$ provides the external body forces due to gravity. The small value of Poisson's ratio is chosen to ensure that the beam deforms significantly under its own weight. This makes the problem suitable for mesh adaptivity.

As before we begin by solving the problem on a uniform coarse mesh, this time containing 192 elements. This mesh is then optimized using the node movement and edge/face swap-

ping algorithms to reduce the total energy from -0.168295 to -0.208546. For this particular mesh the edge/face swapping keeps the number of elements same. Three levels of uniform refinement, each followed by mesh optimization, are undertaken. This produces meshes with 1548, 12415 and 99349 elements and solutions with energies of -0.262773, -0.280849 and -0.285704 respectively.

We consider two further meshes of 98304 and 98370 elements. The first of these is obtained by global refinement of the initial uniform mesh and the second by optimizing this mesh directly. The energies of the solutions on these meshes are -0.272196 and -0.283207 respectively and so we again observe the superiority of the hierarchical approach when r-refinement is combined with global h-refinement.

As with the previous example, our goal is to assess the hybrid algorithm that combines r-refinement with local h-refinement hence we now consider a sequence of meshes obtained in this manner. The first mesh is the same optimized mesh, containing 192 elements, used as the basis for the global refinement results. The energy of the solution on this mesh is -0.208546. Four further locally optimal meshes are then obtained, each time via the use of local refinement (of those elements whose local energy exceeds 60% of the maximum local energy on any element) followed by mesh optimization. These meshes contain 958, 4529, 15315 and 48403 elements and yield solutions with energies of -0.252279, -0.267699, -0.281052 and -0.286102 respectively.

We again conclude our example by illustrating the advantage of applying the hybrid approach hierarchically by contrasting it with the use of local h-refinement alone, possibly followed by a single application of r-refinement. We refine locally the initial mesh of 192 elements in five levels to achieve a mesh of 132698 elements (again using a threshold of 60% for the local refinement). The total energy of the solution on this mesh is -0.278015. The mesh is then optimized to reduce the total stored energy to -0.284321, with an increased number of elements, 132958, due to edge/face swapping. As before it is clear that the quality of the locally optimal meshes obtained in this manner is inferior to that of meshes obtained using

the hierarchical approach. A summary of all of the computations made for this test problem is provided in Table 3 and an illustration of the meshes obtained using multilevel optimization with local h-refinement is given in Figure 9.

Elements	Energy	Description
192	-0.168295	Initial mesh
192	-0.208546	
1548	-0.26773	Multilevel optimization and
12415	-0.280849	global h-refinement.
99349	-0.285704	
98304	-0.272196	Global h-refinement followed
98370	-0.283207	by optimization.
192	-0.208546	
958	-0.252279	Multilevel optimization and
4529	-0.267699	local h-refinement.
15315	-0.281052	
48403	-0.286102	
132698	-0.278015	Local h-refinement followed
132958	-0.284321	by optimization.

Table 3: Summary of the results obtained for Problem Two (the global energy minimum is unknown).

It is interesting to note that for the optimal meshes shown in both this example (Figure 9) and the previous (Figure 7) there are a large number of elements that would be rejected if the usual geometric quality criteria (e.g. [3, 10]) had been employed. Using the energy criterion however these elements are perfectly acceptable.

6 Discussion

The two examples of the previous section have clearly illustrated that the quality of the final mesh produced when using the proposed algorithm is better, in the sense that the finite element solution has a lower energy, than that obtained by either h-refinement or r-refinement alone. Furthermore it is demonstrated that combining the mesh optimization with local h-refinement is superior to combining it with global h-refinement. Finally, the advantage of using the hierarchical approach, whereby intermediate level mesh are optimized, is also apparent: an excellent combination of small mesh sizes and low energies for the corresponding finite element solutions being achieved.

It should be noted that, although quite complex to implement in 3-d, the edge/face swapping component of the hybrid algorithm is crucial. This may be demonstrated, for example, by contrasting the results of Table 2 with those obtained for the same test problem but without the connectivity optimization step included in Figure 1 (see [8] for further details). Such modified results are presented in Table 4 and clearly demonstrate the limitations of the adaptive algorithm when edge/face swapping is neglected. The difference in the solution quality between an energy of 50.75 and 50.04 (where the true optimal value is 50.00) is really quite substantial. In fact much greater accuracy (energy = 50.20) is obtained on a coarser mesh when edge and face swapping are used. The difference in accuracy between the 50.04 solution and the 50.20 solution is less pronounced but this additional level of local refinement does provide a significant improvement nevertheless.

It should also be noted that cpu times have not been included in this paper since our goal has been to investigate mesh optimality rather than to study the fastest way of obtaining a solution of a give accuracy. However some sample solution times are provided in [12, chapter 4] for a variety of different parameters that occur in the algorithm of Figure 1. It is clear that in general it will not pay to spend an excessive amount of time obtaining the very best possible mesh compared to obtaining a good, but slightly sub-optimal, mesh (i.e. requiring more elements to achieve the same accuracy) at a significantly reduced cost. Furthermore, on those optimal

Elements	Energy	Description
384	378.62763	Initial mesh.
384	104.85725	
3072	59.907732	Multilevel optimization and
24576	52.398871	global h-refinement.
196608	50.755212	
196608	67.279033	Global h-refinement followed
196608	52.434265	by optimization.
384	104.85704	
2655	59.902412	Multilevel optimization and
16933	52.381223	local h-refinement.
100866	50.746025	
573834	54.885230	Local h-refinement followed
573834	51.332477	by optimization.

Table 4: Summary of the results obtained for the first test problem *without* edge/face swapping (the global energy minimum is 50.0000).

meshes that have highly distorted elements the condition number of the corresponding discrete equations can be very large and so it will generally require more computational work to solve these equations than those obtained from an inferior, but less distorted, mesh. These twin considerations of time spent obtaining the discrete equations and time spent solving them mean that the problem of obtaining the fastest possible solution of a given accuracy is a lot more complex than the optimal mesh problem considered here. This highly challenging problem is clearly deserving of significant continued research.

To conclude this paper we observe that only two numerical examples have been included here and that further work is likely to be required to ensure the robustness of the proposed algorithm for a wide variety of application problems. In particular, it is likely that the mesh

refinement technique used here will be sub-optimal for problems with highly anisotropic solutions, which may well benefit from a more anisotropic 3-d refinement algorithm, such as [1] for example. It is also possible that different criteria could be used for deciding which elements should be locally refined (e.g. based upon energy gradients rather than energy values) in order to enhance the technique further. Nevertheless, the provisional implementation and results presented here suggest that this approach has significant potential and that further research is indeed likely to be fruitful.

References

[1] E. Bänsch, *"An adaptive finite element strategy for the 3-dimensional time-dependent Navier-Stokes equations"*, Journal of Computational and Applied Mathematics, 36, 3–28, 1991.

[2] M. Delfour, G. Payre and J.-P. Zolésio, *"An optimal triangulation for second-order elliptic problems"*, Computer Methods in Applied Mechanics and Engineering, 50, 231–261, 1985.

[3] L.A. Freitag and C. Ollivier Gooch, *"Tetrahedral mesh improvement using swapping and smoothing"*, International Journal for Numerical Methods in Engineering, 40, 3979–4002, 1997.

[4] C. Ollivier Gooch, *GRUMMP users' guide 7.0"*, University of British Columbia, 1999.

[5] E.B. de l'Isle and P.-L. George, *"Optimization of tetrahedral meshes"*, in Modeling, Mesh Generation and Adaptive Numerical Methods for Partial Differential Equations (Springer, Berlin), eds. I.Babuska *et al.*, 97–127, 1995.

[6] P.K. Jimack, *"An optimal finite element mesh for elastostatic structural analysis problems"*, Computers and Structures, 64, 197–208, 1997.

[7] P.K. Jimack and R. Mahmood, *"Multilevel Approach for Obtaining Locally Optimal Finite Element Meshes"*, in Developments in Engineering Computational Technology, ed. B.H.V. Topping (Civil-Comp Press), 191–197, 2000.

[8] P.K. Jimack, R. Mahmood, M.A. Walkley and M. Berzins, *"A Multilevel Approach for Obtaining Locally Optimal Finite Element Meshes"*, Advances in Engineering Software, 33, 403–415, 2002.

[9] B. Joe, *"Three-dimensional triangulations from local transformations"*, SIAM Journal of Scientific and Statistical Computing, 10, 718–741, 1989.

[10] B. Joe, *"Construction of three-dimensional improved quality triangulations using local transformations"*, SIAM Journal of Scientific Computing, 16, 1292–1307, 1995.

[11] C.L. Lawson, *"Properties of n-dimensional triangulations"*, Computer Aided Geometric Design, 3, 231–246, 1986.

[12] R. Mahmood, *"Multilevel Mesh Adaptivity for Elliptic Boundary Value Problems in Two and Three Space Dimensions"*, Ph.D. thesis, University of Leeds, UK, 2002.

[13] M.E. Ong, "Uniform refinement of tetrahedron", Siam Journal of Scientific Computing, 15, 1994.

[14] S. Rippa and B. Schiff, *"Minimum energy triangulations for elliptic problems"*, Computer Methods in Applied Mechanics and Engineering, 84, 257–274, 1990.

[15] W. Speares and M. Berzins, *"A 3-d unstructured mesh adaptation algorithm for time-dependent shock dominated problems"*, International Journal for Numerical Methods in Fluids, 25, 81–104, 1997.

[16] Y. Tourigny and F. Hulsemann, *"A new moving mesh algorithm for the finite element solution of variational problems"*, SIAM Journal on Numerical Analysis, 35, 1416–1438, 1998.

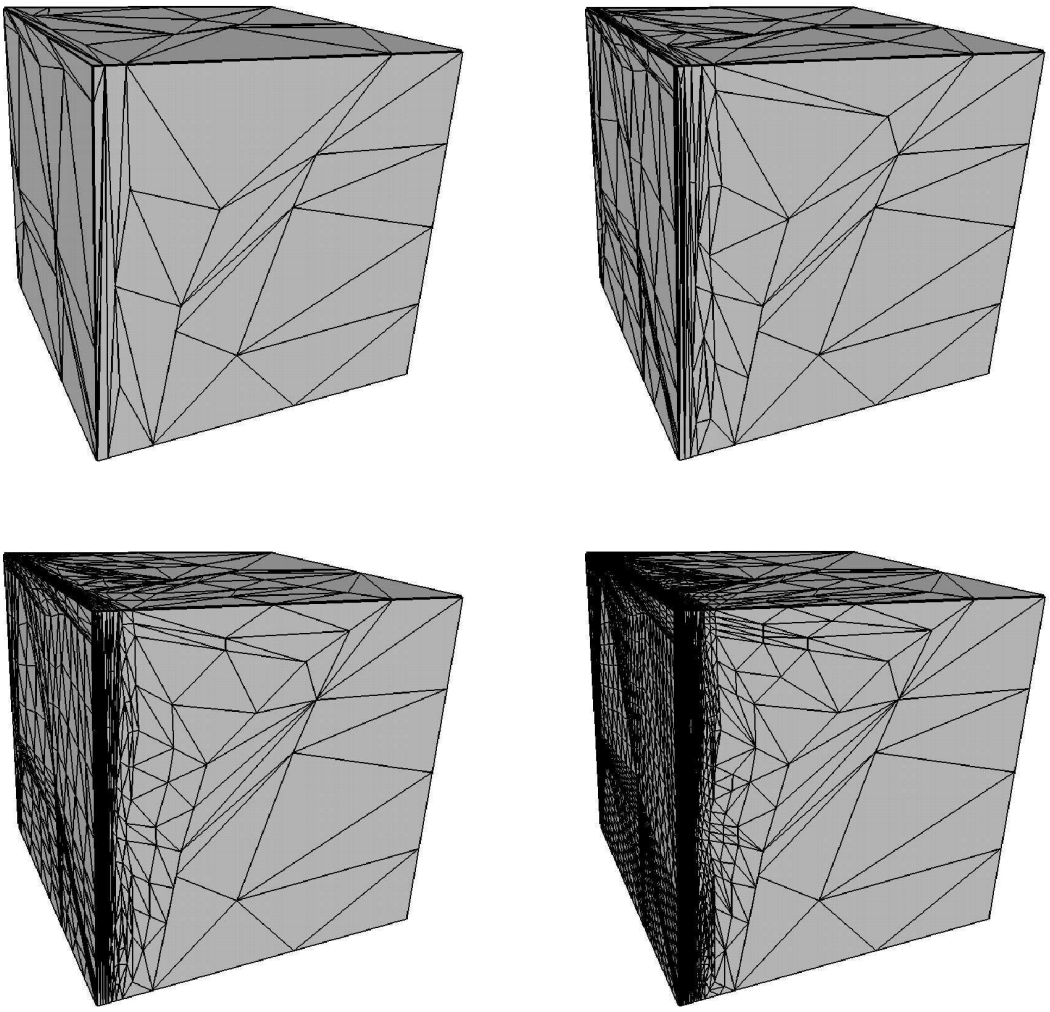

Figure 7: An initial locally optimised mesh (top left) followed by a sequence of meshes obtained by combinations of local h-refinement with r-refinement for the first test problem.

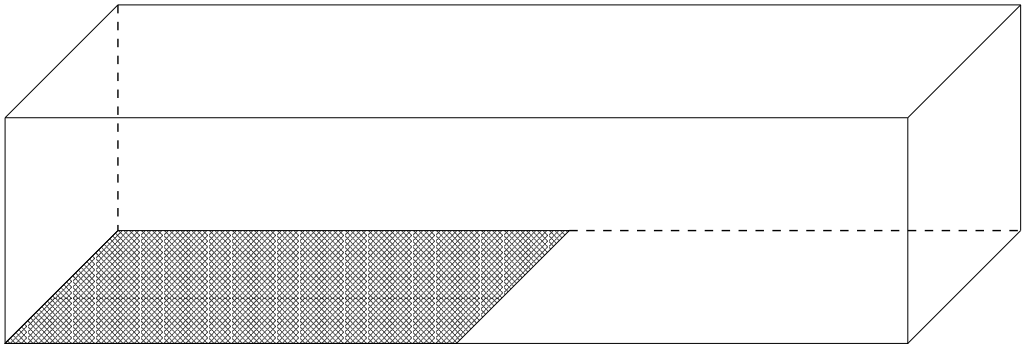

Figure 8: An illustration of the overhanging cantilever beam

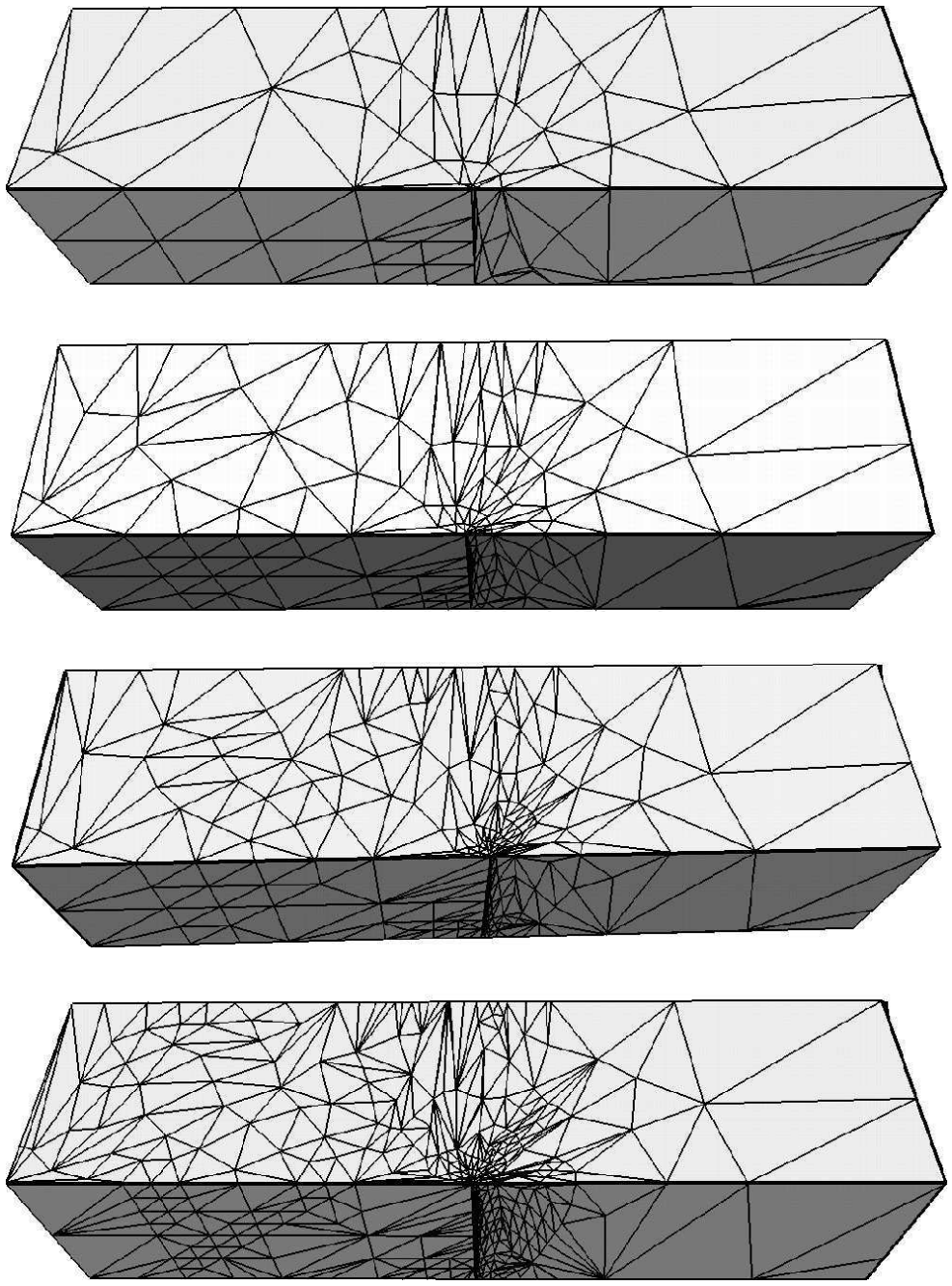

Figure 9: A sequence of meshes obtained by combinations of local h-refinement with r-refinement for the second test problem.