INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:445–455 (DOI: 10.1002/fld.313)

Anisotropic adaptation and multigrid for hybrid grids

Jens-Dominik Müller

School of Aeronautical Engineering, Queen's University Belfast, U.K.

SUMMARY

An anisotropic refinement method for 2D and 3D hybrid grids is presented and applied to viscous flow problems. The algorithm is unique in that it is not limited to a particular grid structure, e.g. hexahedral elements, but allows the anisotropic division of hexahedra and prisms in 3D, quadrilaterals in 2D and the isotropic division of the other element types. At the core of the method is a novel surface tessellation of an element with hanging nodes which guarantees mesh consistency also in the case of arbitrary directional refinement over an arbitrary number of levels. The efficiency of the anisotropic refinement algorithm is evaluated on viscous flow testcases. The resulting grid sequence is compared to an element-collapse sequence for its suitability for directionally coarsened multigrid. Copyright © 2002 John Wiley & Sons, Ltd.

1. INTRODUCTION

Mesh adaptation has a long history in CFD [1]. However, it has not yet become a technique that is used routinely. One of the reasons is that the currently available algorithms are isotropic, much simpler to implement but much less efficient for convectively dominated flows. Of prime interest here are viscous flows with strong shear layers. Commercial packages for unstructured mesh generation for viscous flows generally allow to specify only one global thickness of the layer and a fixed number of elements across the layer. Isotropic adaptation can be used to refine the grid in the normal direction but most of the refinement is wasted. An efficient adaptive method for shear layers has to be able to refine directionally to adapt the mesh to the anisotropy of the flow.

The hierarchy of adapted meshes lends itself straightforwardly for multigrid applications [2–4]. In the case of the Navier–Stokes equations an efficient multigrid method requires directionally coarsened grids to apply smoothing to the strongly coupled modes in the viscous layer [5, 6]. Conversely, when using a sequence of adaptively refined grids for multigrid, the grids have to be refined anisotropically. The combined technique then promises very good convergence rates and robustness, since the coarsening is not based on the geometry but on the flow. The results shown here are preliminary but where comparisons were possible, the inverse-adaptive algorithm compared well against an unstructured semi-coarsening algorithm.

*Correspondence to: J.-D. Müller, School of Aeronautical Engineering, Queen's University Belfast, David Keir Building, Stranmillis Road, Belfast BT9 5AG, U.K.

Received May 2001
Revised November 2001

They have been presented to emphasize the principal strengths and weaknesses of such an approach and to make a case for further development of anisotropic hierarchic adaptation methods. However, the main contribution of the paper is the anisotropic adaptation algorithm.

Anisotropic adaptation has seen very little work. One of the earliest works by Castro-Díaz *et al.* is based on triangular grids [7]. Their technique uses point deletion or insertion on an edge complemented by an edge-swapping and point movement technique. Robichaud *et al.* have extended this to 3D and claim an extension of this algorithm to hybrid grids but fail to present details of their method [8]. For tetrahedral grids their method is robust and we would advocate the use of a similar approach based on a combination of h-, r-refinement and face swapping in the tetrahedral areas of the grid. Lahur and Nakamura have presented a method for Cartesian grids suitable for the solution of the Euler equations [9]. Biswas and Strawn have presented anisotropic refinement procedures for hexahedral grids that remove hanging nodes by buffering with tetrahedra, prisms and pyramids [10]. Mavriplis has presented a 'moderately' anisotropic adaptation method for hybrid grids that allows the cells of the finest level to be refined directionally. If these cells require further refinement, the directional refinement is replaced by an isotropic one [3]. None of the methods above attempts to produce a general algorithm that works for hybrid grids and allows any number of anisotropic refinement levels.

A much more general method for anisotropic refinement of hybrid grids has been developed by the author and co-workers in Reference [11]. In that paper 2D results were presented together with a cell-vertex discretization for elements with hanging nodes. The method bears some similarities to Reference [10], but is based on a separation of the processes involved in adaptation: the marking of edges, the hierarchical refinement of elements according to a selected number of patterns, and the post-processing of interfaces with hanging nodes. One of the contributions of Müller *et al.* [11] is to make these processes independent of each other. This minimizes propagation of the adaption and makes the adaptation independent of the path. Propagation is problematic in that it can dramatically increase the mesh size and in that it leads to additional communication in a parallel implementation. Path independence is important in that previous directional refinements need not be undone in order to refine a split element further or in a different direction.

Here the extension of the method is presented which is necessary to make it work in three dimensions. The major problem that needs to be addressed is the possibility of 'cross-refinement', a quadrilateral face shared by two elements which have been refined directionally but in different directions. The adaptation algorithm must ensure that a consistent flux through that face can be calculated. This problem has been dealt with in Reference [3] by limiting the depth of anisotropic refinement to one level, which severely curtails the efficiency of the adaptive algorithm and precludes it from being used in 'reverse-mode' for directionally coarsened multigrid. Alternatively, as in Reference [10], one can 'buffer' the cross-refined face by tesselating one of the neighbours with the insertion of a central node. This in turn limits the flexibility and efficiency of the method and destroys the regularity of the grid.

A new approach to treating cross-refinement is presented here that is simple and does not impose constraints on the choice of refinement nor the depth of levels. It is based rigorously on the definition of four properties that the adaptation algorithm has to satisfy and the consistency of the mesh is shown, i.e. the surfaces at the refinement interface are shown to match. The algorithm allows arbitrary choice of adaptation direction for an unlimited number of levels, a key requisite for using the adapted sequence in multigrid.

There are still pieces missing to make this algorithm generally applicable. For one, this work does not attempt to deal with the problem of surface reconstruction and the possible mesh repair that has to be done if a surface point has been moved so much as to make the mesh fold. Grid convergence studies with linear surface interpolation will converge to an answer that depends on the coarse initial mesh which is different from the correct smooth surface. Similarly it is not attempted here to perform quantitative comparisons of the accuracy of the results between isotropic and anisotropic adaptation. It is known from the literature that unweighted sensors based on first differences often perform poorly. These sensors are used here, however, because of their simplicity. Future work will investigate the use of adjoint sensors for anisotropic adaptation, preliminary work on this in the isotropic case has been presented in Reference [12].

2. BASIC ADAPTATION ALGORITHM

Four paradigms underly our anisotropic adaptation algorithm (AAA).

1. There can be at most a difference of one level in refinement between neighbouring elements. This limitation is adopted by nearly all adaptive methods and simplifies the construction of conservative fluxes across these interfaces.
2. All edges selected for refinement by the user's sensor must be refined in all elements. The algorithm can add other edges to be refined in some elements only. This distinction between required and non-required edges adds a large degree of flexibility and minimizes propagation. It is a unique feature of the AAA.
3. The directional refinements must be orthogonal in the sense that they combine to an isotropic one. This requirement makes the final grid independent of the way in which it has been refined. It also ensures that the element quality does not degrade in recursive refinements.
4. The surface tessellation of an element with hanging nodes is uniquely prescribed by the distribution of hanging nodes around the faces. It recovers the edges of the elements that may exist on either side of a face. This guarantees conservation by construction. Note that this implies that the edges of elements touching on a face have to be recovered by the node pattern on it. As shown below, this can be achieved in a simple way for quadrilateral faces but cannot be guaranteed for directionally refined triangular ones.

In our implementation a scalar error indicator is calculated for each element, e.g. the maximum of the first-order velocity differences along the edges. Either a fixed fraction of the elements is derefined or refined or the elements below and above one mean deviation from the average. With this initial list we can loop over the elements and select for each element a refinement pattern from a predefined list that contains all required edges (cf. Figure 1).

These directional refinement patterns satisfy property 3, they sum to the isotropic one. The refinement pattern that is applied to the element is chosen such as to refine all the edges that are 'required', either because they are flagged by the sensor or they are needed due to propagation, as explained below. The non-required edges which are needed to complete the pattern are added to the list of refined edges. Note the difference between the required edges which need to be refined in all elements and the completion edges which may be hanging in a neighbouring element.

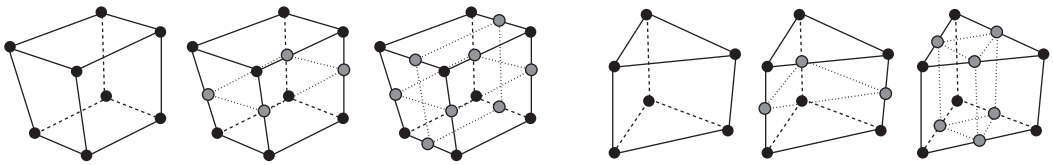

Figure 1. The directional refinement patterns for 3D elements, appropriate rotations apply. The isotropic patterns are uniquely defined (except for the choice for interior diagonal in a tetrahedron) and are not shown.

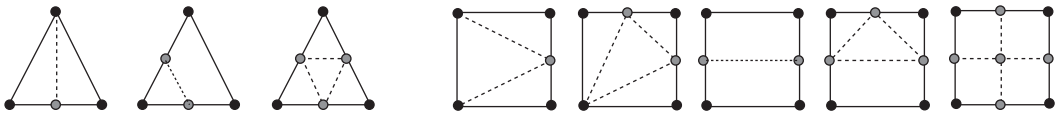

Figure 2. Unique surface tessellations for faces with hanging nodes.

If a refinement edge is requested between the mid-node and one of the end-nodes of a hanging edge there is a difference of two levels. In this case the parent edge needs to be added to the list of required edges, ensuring that it will be refined in all elements and the difference in levels is reduced to one. A loop over all elements to upgrade to the next containing refinement pattern is executed until no more edges have been added.

At this stage each element is marked with a refinement pattern that is consistent with the first three properties. Satisfying property 4 then leads by construction to a grid suitable for a conservative discretisation: for each face that is shared by two elements we can identify the unique surface tessellation by looking up the hanging edges in our list.

2.1. Tessellating the faces

The existence of a unique surface tessellation for each face depending on the distribution of hanging nodes around it, property 4, is crucial for the algorithm. In two dimensions this is trivial: an edge with a hanging node is split in two. In three dimensions we have to distinguish between triangular and quadrilateral faces. Figure 2 shows all possible combinations, appropriate rotations apply.

A strictly limited propagation of refinement due to level differences in refinement has to be accepted even in isotropic refinement. This propagation is needed to satisfy property 1, at most one level difference in refinement between neighbours. Consider, e.g. two neighbours with one having been refined. If this one is to be refined again, the algorithm has to refine the neighbour as well, in order to limit the level difference.

Biswas and Strawn show an example of propagation due to different anisotropic refinement directions [10]. Resolving this difference by reverting to isotropy would result in a propagation of the isotropic refinement through the domain until a boundary is hit. In Reference [10] it is resorted to buffer one of the elements by insertion of a central node in the gravity centre of one of the elements and by connecting the faces to this centroid.

Here this contention is resolved by making the tessellation of the shared face uniquely dependent of the distribution of hanging nodes around it, irrespective of the directional refinements that produced these nodes. In Figure 2 the reader might have noted the absence

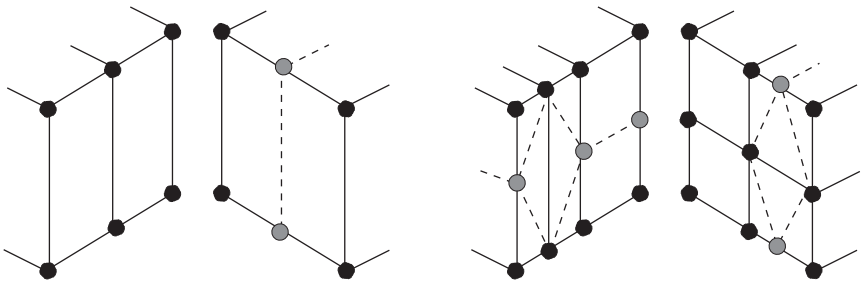

Figure 3. Cross-refinement of a quadrilateral face over two levels. First refinement on the left, second refinement on the right. Hanging nodes shaded, surface tessellation dashed.

of the case with four hanging nodes around the perimeter and no central node on the face. In this case a unique surface tessellation that recovers the elemental edges cannot be defined (Figure 3). The AAA introduces a node at the centre of a quadrilateral face that has hanging nodes on all four edges. The resulting surface tessellation then recovers the edges of the children elements. This node can be a hanging one in all the cells it appears in.

The example in Figure 3 shows that 'face' has to be interpreted in a more general way for the addition of the central node. We see in the left half of the figure a face shared between two elements of which the left one has been refined directionally. At this stage there is no problem and the configuration of hanging nodes around the face results in the unique surface tessellation. For the next refinement stage the left child of the left element has been refined directionally again, while the right element has been refined isotropically. The surface tessellations differ when viewed from the left or from the right.

This is, of course, cured by the insertion of a central node. However, the face it is inserted on does not form part of an element. It is split on both sides in different directions. To our advantage this case can only occur on 'half-faces' of a parent element. Quarter faces are necessarily the elemental face of a child. Thus, we have to include in our test for adding the central node each quadrilateral face and all its four possible half-faces if the element is a parent.

It is much more difficult to achieve a unique tessellation for directionally refined triangular faces. However, the directional refinement of a triangular face also violates the third property, orthogonality of directional refinements. E.g. three directional splits of a triangle through each side do not result in an isotropically divided one. Consequently, the algorithm is limited to allow directional divisions of quadrilateral faces only, that is hexahedra and prisms in 3D. All other elements will be subdivided isotropically. This, in practice, in not a severe restriction since most mesh generation algorithms place layers of either prisms or hexahedra in boundary layers to achieve elements with high aspect-ratios and good orthogonality.

3. RESULTS

3.1. Viscous NACA 0012 starting with an Euler mesh

To demonstrate the power of anisotropic adaptation the viscous flow over a NACA 0012 profile is calculated starting from an isotropic grid for Euler calculations. We do not advocate such an approach, of course, since it is not an efficient procedure. However, anisotropic adaptation is very suitable to adjust the mesh stretching in an arbitrary fashion.

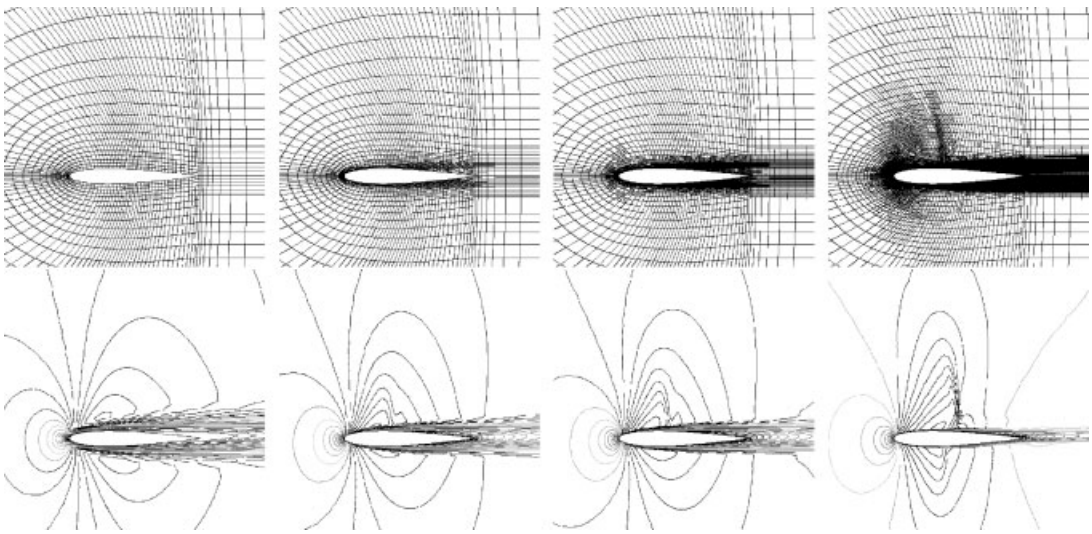

Figure 4. Anisotropic adaptation for viscous NACA 0012 airfoil, levels 0, 2, 4, 6. View of the airfoil above, close-up near the shock below.

The freestream Mach number is 0.8, the angle of attack 1.25° and the Reynolds number 10^5. The calculations have been done with the edge-based finite volume multigrid solver of [6]. A first-order velocity difference was used as a refinement sensor in all of the cases. The adaptation is based on first differences of velocity, with the refinement fraction rising from 15% to 60% for levels 1 through 6. The grids and solutions are shown in Figure 4, the final solution is shown in Figure 5.

Figure 6 shows a comparison between the coarser levels used for the multigrid algorithm. The regular grids on the left are the refined grids of level 3 and 4, the coarser ones are the second and third level coarsened by applying element-collapsing [13] to the level 6 adapted mesh. The convergence history with the two different sequences is compared in Figure 5.

The inverted sequence of refined meshes is able to coarsen the boundary layer more rapidly. The coarsening algorithm on the other hand has no knowledge of the grid hierarchy, but has to satisfy positive element volumes after each local collapse. Moreover, the element-collapsing sees the buffered mesh while refinement works on the cleaner mesh with hanging nodes.

The situation is reversed in the farfield. The adaptive algorithm starts with a reasonably fine Euler mesh and cannot coarsen beyond that. The element-collapsing coarsens the far-field starting on the first coarser grid, applying coarsening everywhere in the grid and not just in the refined regions. It is this global coarsening that is responsible for the superior convergence of the element-collapse sequence in Figure 5.

The combination of the two algorithms in a zonal way promises to be very effective. An initial solution can be obtained using a moderately stretched grid and a collapsed multigrid sequence. The refinement generates patches of directionally refined grid which can be solved for in zones [3]. The final solution will then be a composite of the unrefined zones on the initial mesh and the finest level of refinement elsewhere.

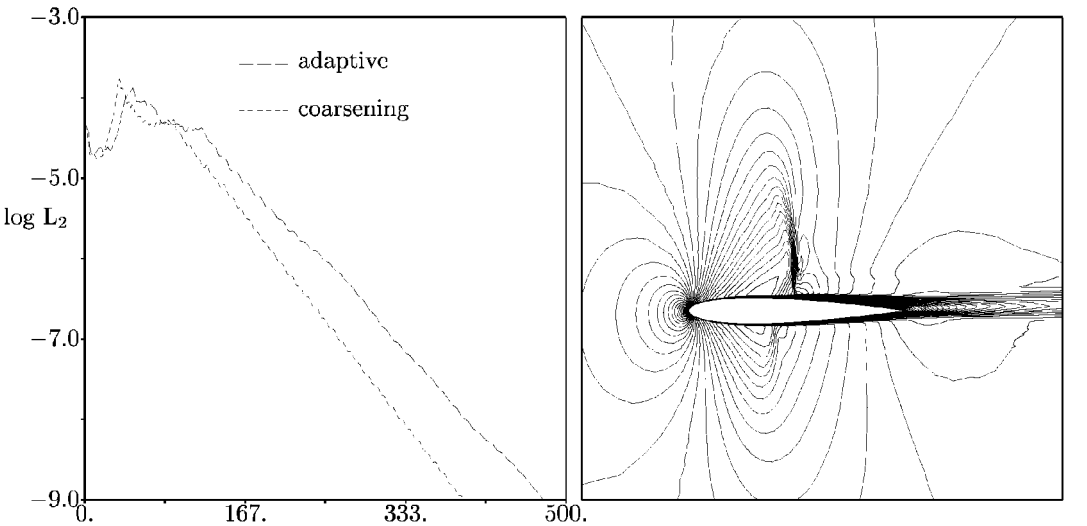

Figure 5. Mach contours and convergence histories for the viscous NACA 0012.

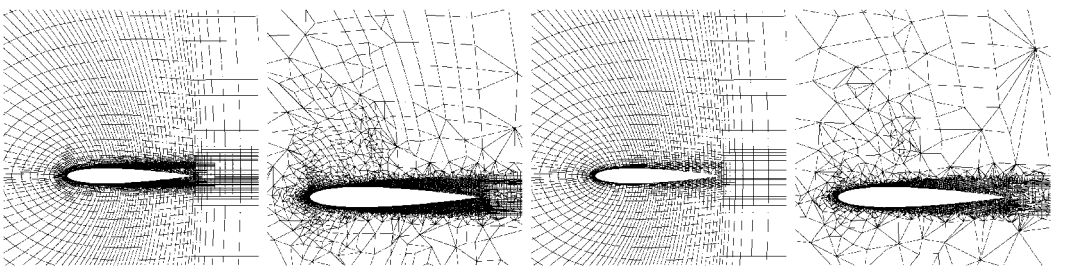

Figure 6. Multigrid levels 2 and 3 for the grid of level refinement 6 for the NACA 0012 from left. Adaptive hierarchy on the left, element-collapsing on the right.

3.2. Viscous RAE 2822

This testcase looks at the efficiency of the directional adaptation in 2D and compares the accuracy with isotropic refinement. The initial mesh for this testcase is a hybrid grid with the quadrilateral portion derived from a fine structured grid with 353×65 nodes that has been coarsened to 177×33. The freestream Mach number is 0.725, the angle of attack 2.4° and the Reynolds number 6.5×10^6.

The boundary surface is not recovered for the examples in this section but interpolated as a piecewise linear. The initial mesh is already highly stretched and it is very likely that a boundary node when moved onto the actual surface will create a folded element with negative volume. Linear interpolation limits the levels of adaptation to one or two; otherwise the refinement process would quite correctly resolve expansion fans at the kinks in the geometry.

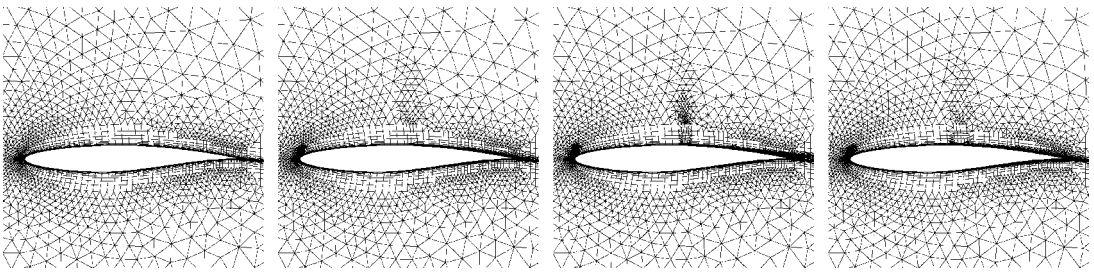

Figure 7. RAE 2822 airfoil, Ma 0.725, α 2.4°, Re 6.5×10^6. Initial grid, first anisotropic, second anisotropic and first isotropic refinement from the left.

Table I. Refinement of the RAE airfoil.†

Level	Refined	Children	Added	Total	Buffered
0				7240	7240
Aniso 1	987	2176	1189	8429	8700
Aniso 2			2847	11276	12865
Iso 1	987	3948	2203	10201	11102

†Total grid size is the size with hanging nodes, buffered is the size when hanging nodes have been removed.

Figure 7 shows the grids. The refinement threshold was set to one mean deviation above the average velocity difference along an edge, resulting in a refinement of 14% of the cells for the first and 34% on the second level. As can be seen from Table I, the first anisotropic refinement adds only 40% of the elements of the isotropic refinement, the second level grid with anisotropic refinement is comparable in size with the first level isotropic one. 77% of the elements were refined anisotropically for the level 1 grid.

Figure 8 compares the c_p contours for the different refined grids with a fine grid one based on the 333×65 hybrid grid which recovers the surface in all boundary nodes. It can be seen that apart from the resolution of the shock the results are very similar, the initial grid with 7240 elements is already very good. The c_p curves in the shock of the first level refinements with isotropic and anisotropic refinement are nearly identical, at 30% extra cost, the second level anisotropic is slightly more accurate. Since the surface is not recovered, we cannot expect to converge towards the fine grid solution.

3.3. Generic low pressure turbine

This example is a very coarse hexahedral grid around a generic low pressure turbine. A cross-section at mid-span is shown in Figure 9. Close-ups of the trailing edge are shown in Figure 10. The flow is steady on the initial grid but becomes unsteady due to vortex shedding around the blunt trailing edge in the adapted meshes that have a smaller artificial viscosity due to the finer mesh. This precludes a quantative comparison of the results and of the convergence history. The testcase is presented to show that the adaptation algorithm does work in three dimensions.

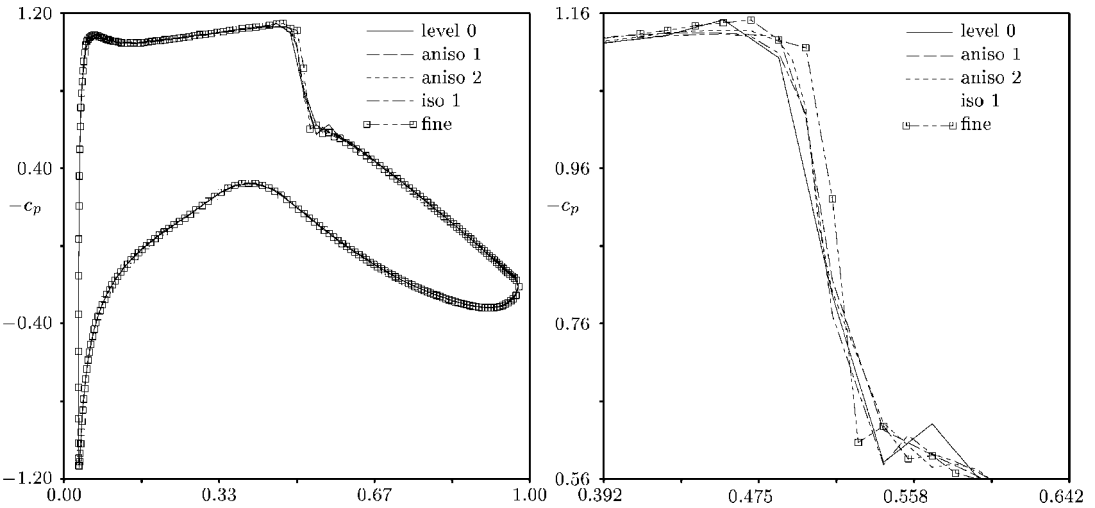

Figure 8. RAE 2822 airfoil, Ma 0.725, α 2.4°, Re 6.5×10^6.
C_p distribution for adapted grids and a fine grid.

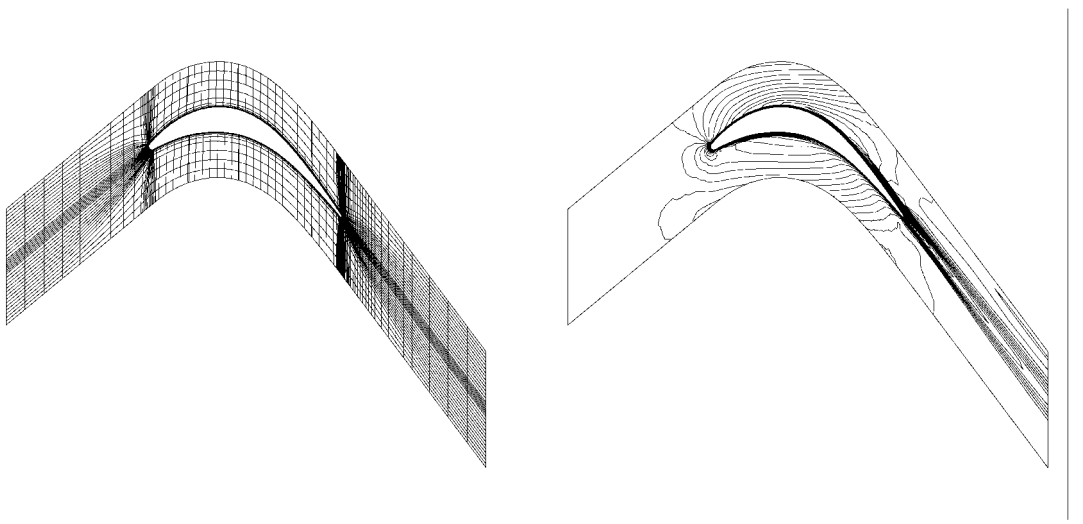

Figure 9. Generic low pressure turbine. Grid cross section and Mach contours at half-span.

Thirty per cent of the elements have been adapted based on velocity differences. Table II shows the sizes of the adapted grids. The anisotropic refinement added only a fifth of the elements compared to the isotropic refinement, after buffering the grid size was less than half. The second level grid with anisotropic refinement is smaller than the isotropic first level one.

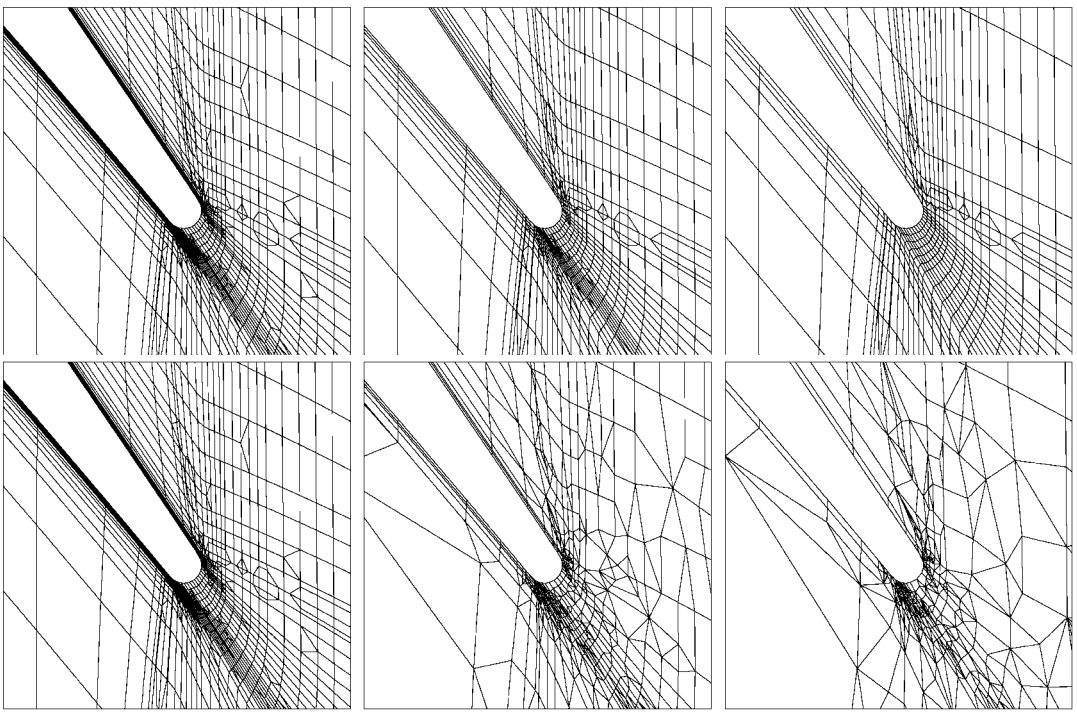

Figure 10. Two levels of anisotropic refinement compared to two levels of coarsening from the refined grid. The coarsening approach results in a better multigrid sequence in the farfield.

Table II. Generic low pressure turbine.[†]

Level	Added hex	Tet	Pyr	Pri	Hex	Total
0					95760	95760
Iso 1	201089	95545	118131	6122	279473	499271
Aniso 1	38201	57678	54796	0	122937	235411
Aniso 2		147967	150491	46	159664	458168

[†]The leftmost column lists the number of added elements before buffering, the ones to the right the element counts of the grids after buffering.

Figure 10 shows the initial grid and two levels of refinement. All of the refinement occurred in the boundary layer, leaving large parts of the far-field unchanged, favouring a combined zonal approach for the multigrid.

4. CONCLUSIONS

An adaptation algorithm has been presented that allows anisotropic refinement of hexahedra and prisms without restrictions of level or order. The crucial ingredient of the algorithm is

the definition of a unique surface tessellation that depends only on the hanging nodes of the element. The insertion of a hanging node on faces that are fully refined on the perimeter ensures a unique surface tessellation also in the case of refinement in different directions on either side of the face. Using this local test it can be guaranteed that the mesh is globally consistent.

Comparisons of the anisotropic and isotropic refinements for various viscous flow cases show the efficiency of anisotropic adaptation. The refinement of boundary layers results in near 80% of all elements being refined anisotropically. For a 3D viscous case with a very coarse boundary layer discretisation the anisotropic strategy adds only $\frac{1}{5}$ of the elements as compared to the isotropic one. In both 2D and 3D the second level anisotropically refined grid was comparable in size to the first level isotropic one.

It was also investigated how the resulting hierarchy of 'semi-refined' grids could be used for 'semi-coarsened' multigrid. The refinement sequence was compared to a sequence resulting from applying a directional element-collapsing algorithm to the adaptively refined grids. It has been shown that the two approaches have very complementary advantages and disadvantages. The element-collapsing coarsens also in the far-field, but the coarsening stalls after 3–4 levels. The inverted refinement sequence by construction has a perfect coarsening of the boundary layer, retaining the quality of the original grid throughout. It never coarsens beyond the initial grid, however, leading to too fine a grid in the far field. Future work will investigate the combination of the two approaches in a zonal method.

REFERENCES

1. Thompson JF, Soni BK, Weatherill NP. *Handbook of Grid Generation*. CRC Press: Boca Raton, FL, 1999.
2. McCormick S, Thomas J. The fast adaptive composite grid (fac) method for elliptic equations. *Mathematics of Computation* 1986; **46**:439–456.
3. Mavriplis DJ. Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes. *AIAA Paper 97-0857-CP*, Reno, 1997.
4. De Zeeuw D. A quadtree-based adaptively-refined Cartesian-grid Algorithm for Solution of the Euler Equations. *Ph.D. Thesis*, The University of Michigan, 1993.
5. Allmaras S. Analysis of a local matrix preconditioner for the 2-d Navier–Stokes equations. *AIAA Paper 93-3330-CP*, 1993.
6. Moinier P, Müller J-D, Giles MB. Edge-based multigrid schemes and preconditioning for hybrid grids. In *AIAA-CP-99-3339*, *14th AIAA CFD Conference*, Norfolk, VA, June 1999.
7. Castro-Díaz MJ, Hecht F, Mohammadi B, Pironneau O. Anisotropic unstructured mesh adaptation for flow simulations. *International Journal for Numerical Methods in Fluids* 1997; **25**:475–491.
8. Robichaud M, Ait-Ali-Yahia D, Baruzzi G, Kozel V, Peeters M, Habashi W. 3-d anisotropic adaptation for external and turbomachinery flows on hybrid unstructured grids. *AIAA paper 2000-2248*, *Fluids 2000 Conference*, Denver, CO, 2000.
9. Lahur PR, Nakamura Y. Anisotropic cartesian grid adaptation. *AIAA Paper 2000-2243*, Fluids 2000, Denver, CO, 2000.
10. Biswas R, Strawn RC. A dynamic mesh adaption procedure for unstructured hexahedral grids. *AIAA-CP 96-0027*, 1996.
11. Müller J-D, Schönfeld T, Rudgyard M. A comparison of the treatment of hanging nodes for hybrid grid refinement. *AIAA-CP-97-1859*, 1997.
12. Müller J-D, Giles MB. Solution adaptive mesh refinement using adjoint error analysis. *AIAA-CP-2001-2550*, 2001.
13. Müller J-D. Coarsening 3-d hybrid meshes for multigrid methods. In *9th Copper Mountain Multigrid Conference*, April 1999.

