INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:593–603 (DOI: 10.1002/fld.316)

Agglomerated multigrid on hybrid unstructured meshes for compressible flow

K. A. Sørensen, O. Hassan*, K. Morgan and N. P. Weatherill

Department of Civil Engineering, University of Wales, Swansea, SA2 8PP, U.K.

SUMMARY

A review of the applicability of the agglomerated multigrid solution acceleration approach for the simulation of compressible flows is presented. The flow descriptions are wide ranging, spanning from inviscid to turbulent and from steady state to transient. The spatial discretization is performed by a node-centred finite volume scheme, with explicit addition of fourth-order artificial dissipation. The improved performance resulting from the introduction of unstructured hybrid meshes is illustrated. For time-dependent flows involving moving geometries, a second order geometrically conservative scheme is applied, with multigrid accelerated implicit timestepping. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: compressible flow; agglomerated multigrid; unstructured mesh; finite volume; time accurate

1. INTRODUCTION

In industry, steady inviscid compressible flow calculations on full aircraft configurations are now becoming routine. There, is however, a need to improve on the methods at use in present for more complicated calculations involving viscous effects and time dependency. Viscous problems often involve meshes of tens of millions of elements which are highly stretched close to wall surfaces while explicit transient calculations often require large number of timesteps due to the stability limit of the numerical scheme. These factors result in slow convergence and limits the practical use of such computations.

The current approach addresses these problems by applying a multigrid acceleration procedure, known for its excellent convergence behaviour on inviscid problems at a minimal increase in memory usage. Geometric multigrid does however require the construction of several meshes of an *a priori* specified decreasing complexity, which often proves to be a difficult task for complex problems. This issue has prompted research into automatically creating the set of meshes from one original, created in an unstructured mesh generator [1, 2].

*Correspondence to: K. A. Sørensen, Department of Civil Engineering, University of Wales, Swansea, Singleton Park, Swansea SA2 8PP, U.K.

Contract/grant sponsor: The Research Council of Norway; contract/grant number: 125676/410

Received May 2001
Revised October 2001

The current approach uses the finest mesh as a starting point, creating the coarser meshes by an agglomeration process [3]. This method was selected for its robustness and the fact that it generates nested meshes. The solver used is a vertex-centred finite volume method with explicit addition of fourth-order dissipation and is fully parallelized.

For additional reduction in computation time on viscous flows, the use of hybrid meshes is investigated. Quadrilaterals are introduced in two-dimensional meshes, and pyramids, prisms and hexahedra are generated in three dimensions. The creation of these elements is performed by merging elements created in a simplex unstructured mesh generator in several layers close to wall boundaries where the original mesh is created by an advancing layer scheme [4]. For a viscous three-dimensional mesh, the majority of the elements are situated close to wall boundaries and this approach therefore yields a significant decrease in the number of mesh edges in the problem.

For transient problems, implicit timestepping is used. This results in a reduction of the number of timesteps required for an accurate solution, and the reduction in solution time is often of more than an order of magnitude. The scheme also allows for geometry movement where an ALE approach is used [5]. For problems involving relatively small movement a mesh movement technique is applied. If there are regions of large mesh movement, the mesh movement technique may produce meshes of poor quality, such regions are detected and local remeshing is performed.

2. NUMERICAL SCHEME

The time-dependent, Favre-averaged, compressible Navier–Stokes equations in integral form on a three-dimensional Cartesian domain $\Omega(t) \subset \mathbb{R}^3$, with surface $\partial\Omega$, can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega} U_i \, \mathrm{d}\mathbf{x} + \int_{\partial\Omega} (F_{ij} - U_i v_j) n_j \, \mathrm{d}\mathbf{x} = \int_{\partial\Omega} G_{ij} n_j \, \mathrm{d}\mathbf{x} \tag{1}$$

where n_j is the outward unit normal vector to $\partial\Omega$. The domain is allowed to be time dependent and the velocity of the domain boundary is denoted by v_j. The unknown vector of the conservative variables is given by U_i while the inviscid and viscous flux tensors are denoted by F_{ij} and G_{ij}, respectively. To close this system for turbulent flow calculations, the Spalart–Allmaras turbulence model [6] is applied.

2.1. Discretization procedure

When an edge based representation of the mesh is employed, a typical integral over the boundary Γ_I of the control volume surrounding node I is computed using the approximation

$$\int_{\Gamma_I} \mathbf{H}^j n_j \, \mathrm{d}S \approx \sum_J \frac{1}{2} C_{IJ}^j (\mathbf{H}_I^j + \mathbf{H}_J^j) + \mathbf{B}_I \tag{2}$$

for a general $\mathbf{H}^j$. Here, the summation extends over all nodes J that are connected to node I, $\mathbf{H}_I^j$ is the value of $\mathbf{H}^j$ at node I and the edge coefficients are defined as

$$C_{IJ}^j = \sum_k A_k n_k^j \tag{3}$$

The summation now extends over all control volume interfaces k that intersect the edge between nodes I and J, A_k is the length in 2D and the area in 3D of interface k and n_k^j is the unit outward normal to this interface. The boundary term, $\mathbf{B}_I$, is only non-zero when node I is located on the boundary of the computational domain. In this case,

$$\mathbf{B}_I = \sum_J \frac{1}{4} D_{IJ}^j (3\mathbf{H}_I^j + \mathbf{H}_J^j) \tag{4}$$

where the summation extends over all boundary nodes, J, connected to node I by an edge on the boundary and

$$D_{IJ}^j = A_e n_e^j \tag{5}$$

Here A_e is the area of the control volume face associated with the edge e between I and J, which simplifies to one-half of the length of the edge in two dimensions, and n_e^j is the unit outward normal of the control volume. Volume integrals are performed under the assumption of a constant integrand within the control volume. For a simplex mesh, these procedures produce a set of discrete equations that are equivalent to a mass-lumped linear finite element Galerkin scheme at interior nodes.

On hybrid meshes, the dual mesh construction is performed using a modified median-dual approach. The merging of tetrahedra creates quadrilateral faces that are in general not planar. This means that the usual definition of face centroid as the co-ordinate average is invalid. Instead the face is defined by the two triangular faces originating from the simplex mesh and the midpoint of the interface edge between the two faces is used as face centroid. In addition, the element centroids on the prisms, pyramids and hexahedra are chosen in a way to guarantee that they are contained within the element and that the dual mesh definition coincides with the median-dual for a regular element [7]. For simplex meshes the normal median-dual approach is used.

Stabilization and discontinuity capturing are achieved by replacing the physical convective flux function over each edge by a consistent numerical flux function of the JST type [8, 9]. For stabilisation, the fourth-order diffusion operator is constructed in a form that preserves a linear field [2], while discontinuity capturing is realized by the addition of a pressure-switched second-order diffusion.

2.2. *Agglomerated multigrid*

To accelerate the convergence of the method, the FAS multigrid scheme of Brandt [10] is applied with local timestepping and explicit three-stage Runge–Kutta relaxation. In the current implementation, a linear restriction mapping is used, while point injection is applied for the prolongation mapping.

The edge-based formulation provides a method for merging control volumes at a given grid level, thus automatically creating a nested coarser mesh for use in the multigrid procedure [3]. This approach has proven to be very robust and efficient, in addition to producing nested meshes which simplify the prolongation and restriction operations. The procedure itself is also very fast, typically generating the mesh cascade in the time it takes to perform 1–3 multigrid cycles. The agglomeration scheme can be summarized

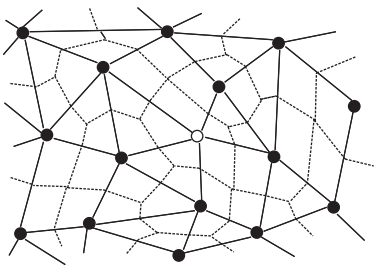

Figure 1. The agglomeration procedure begins with selecting a seed node. The selection of this node is either performed randomly or by connectivity criteria. The mesh is shown in solid lines while the dashed lines denote the dual mesh.

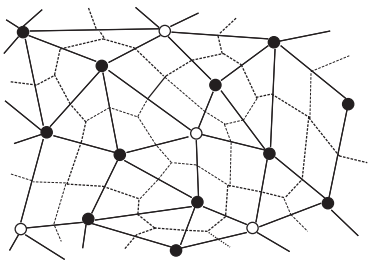

Figure 2. For homogenous agglomeration, every node connected to the seed that have not yet been agglomerated, are associated with the seed and the control volumes are merged. The resulting dual is shown in bold dashed lines.

as follows:

1. A seed node is selected. Initially, the seed node is taken from the list of boundary nodes. When no boundary nodes remain, the seed node is selected in a random manner from the internal nodes in the mesh.
2. The nodes connected to the seed node by edges that have not already been merged are grouped together creating a supernode.
3. When the above procedure is completed for all nodes, internal edges in the supernodes are deleted and edges bordering the same two supernodes are merged into superedges. This is done by adding the coefficients defined in (3) of the merged edges. In a similar way, the boundary coefficients defined by (5) are added.

Directional agglomeration is used on highly stretched meshes by restricting merging to directions with large edge coefficients. This is done with the objective of enhancing the stability and convergence rate of the scheme [1]. The agglomeration scheme is illustrated in Figures 1–4.

2.3. Time-accurate calculations

When time accuracy is desired, a second order time discretization is applied to the time term appearing in the governing equations. The time term is treated as a source for the steady-state governing equations, and this system is solved in a similar fashion to steady-state calculations.

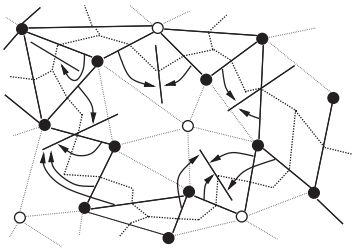

Figure 3. Edges internal to an agglomerated domain are deleted and edges spanning between the same two agglomerated domains are combined into one edge by summing the edge coefficients.

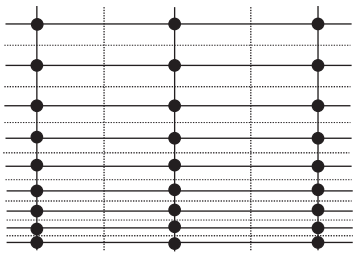

Figure 4. For anisotropic meshes, directional agglomeration is performed where merging of nodes is only carried out in the direction of large edge coefficients.

In this way, the timestep can be selected from accuracy considerations only, without taking into account stability of the timestepping. It also allows the use of convergence acceleration and local timestepping for the subiterations of each timestep.

The formulation presented allows for time-dependent control volume definitions which may be necessary if the geometry in consideration is moving in time. The surface velocities of the control volumes are selected in a way to ensure the satisfaction of the geometric conservation law [5]. The movement of the mesh is performed by the spring analogy method [11] where the outer mesh boundary is held fixed and the edges in the mesh are behaving as springs when the geometry is altered. This approach retains the connectivity of the mesh and produces meshes of high quality for significant mesh movement. In some cases however, the mesh movement procedure can produce regions of low mesh quality. This may, for example, occur if the geometry consists of several objects that are moving in relation with each other, such as for store separation. When this occurs, the program removes the collection of low-quality elements in the mesh and performs a local remeshing.

3. NUMERICAL EXAMPLES

A set of numerical examples are included to illustrate the approach. The computations include an inviscid calculation on a complicated aeroplane geometry, viscous calculations on

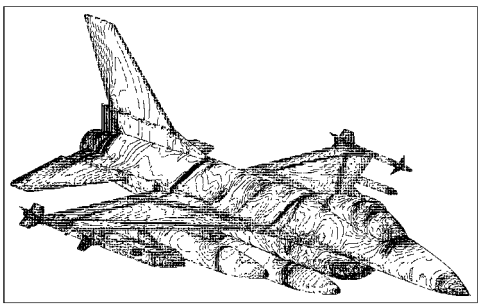

Figure 5. Pressure contours on surface for inviscid F16 calculation.

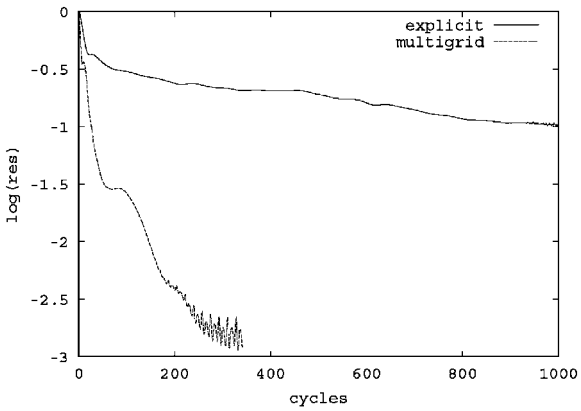

Figure 6. The convergence for the multigrid solver on the inviscid F16 calculation. The solver could not produce a true steady-state solution for this problem due to shedding behind the stores of the aircraft.

an aerofoil and a wing, and a time-dependent deforming mesh calculation on an aeroplane configuration.

3.1. Inviscid flow over F16 with full stores

As an example of a typical industrial application that can be efficiently solved with the computer resources available today, an inviscid calculation on an F16 geometry is included. The Mach number used is 0.9 and the angle of attack is 5.5. The calculation was performed in parallel using eight partitions. The wall-clock time required for the calculation was around 2 h, a speedup of over 20 compared to a single-grid parallel solver with the same number of partitions. A plot of the pressure calculated on the surface is shown in Figure 5. The convergence curves of the multigrid accelerated procedure and for a single-grid solver are shown in Figure 6. The surface definition of the F16 aircraft was generously provided by EADS.

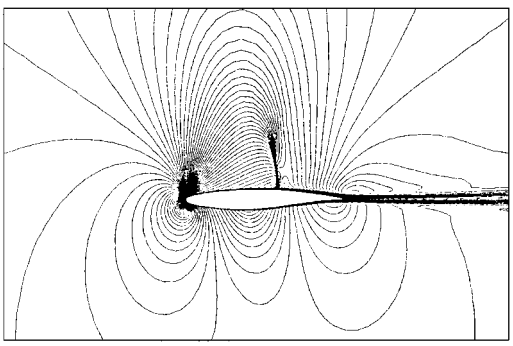

Figure 7. Mach field for RAE2822 example.

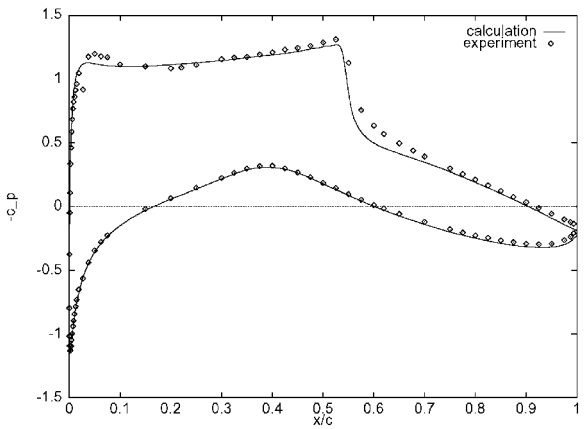

Figure 8. Pressure coefficient on surface for RAE2822 example.

3.2. *Turbulent flow over RAE2822 aerofoil*

Turbulent transonic flow around the RAE2822 aerofoil is considered. The Mach number is 0.73, $Re = 6.5 \times 10^6$ and $\alpha = 2.79$. The mesh used is refined in the shock region and contains 71 319 triangular elements and the first mesh layer from the aerofoil surface is placed at 2.5×10^{-6} times the cord distance from the wall. A plot of the Mach field is given in Figure 7 and a comparison between computed and experimental surface pressure coefficients is shown in Figure 8. A comparison between the convergence for the multigrid scheme and a single-grid solver is shown in Figure 9. The multigrid speedup to engineering accuracy compared to a single-grid solver in this case is over 12.

3.3. *Laminar flow over ONERA M6 wing*

This example considers laminar flow over an ONERA M6 wing at a mid-chord-based Reynolds number of 1.0×10^6. The angle of attack is 3.0 and the Mach number is 0.5. The testcase was run on three different meshes; two different hybrid configurations and a regular unstructured

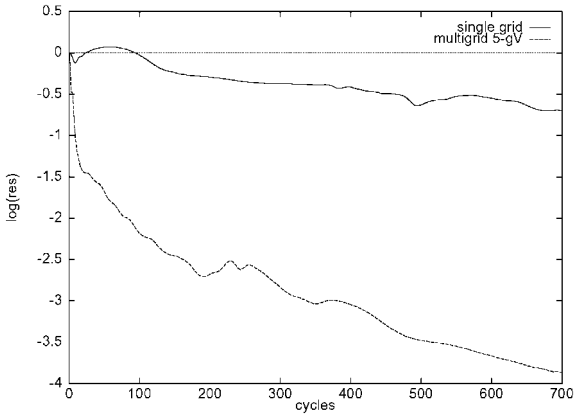

Figure 9. Convergence curves for RAE2822 example.

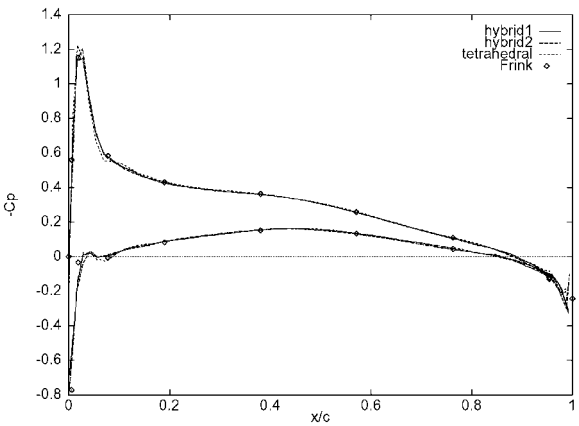

Figure 10. Comparison of pressure coefficient at $\eta = 0.44$ on the ONERA M6 wing testcase for various mesh types.

tetrahedral mesh. The tetrahedral mesh consists of 594 022 elements and 705 883 edges. The first hybrid mesh consists of 193 910 tetrahedra, 250 pyramids, 133 204 prisms and 505 506 edges. The second hybrid mesh consists of 191 822 tetrahedra, 3541 pyramids, 70662 prisms and 34049 hexahedra which adds up to 477 404 edges. The nodal structure in the three meshes are identical and all meshes have 103 168 nodes. Figure 10 shows a comparison of the surface pressure at wing station $\eta = 0.44$ for the meshes used and with the results of Frink [12]. In Figure 11 the convergence curves of the meshes used are shown using a five grid multigrid cycle scheme, and for the tetrahedral mesh using a one-grid procedure. Figure 12 shows the convergence history for the wing lift coefficient. There was little difference between the two hybrid meshes for this testcase, the solution process on both required about 60% of that required for the tetrahedral mesh, with a total speedup of around 10 compared to a single grid solver.

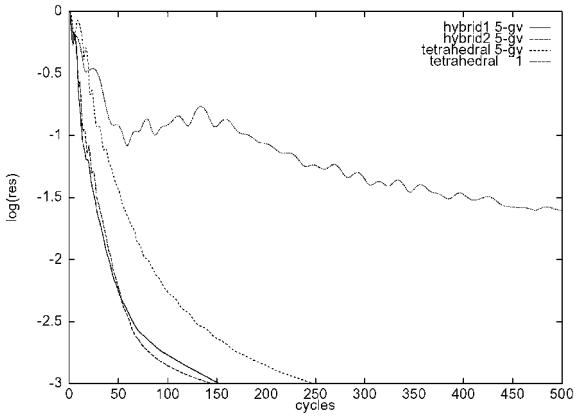

Figure 11. Convergence curves for laminar ONERA M6 wing testcase.

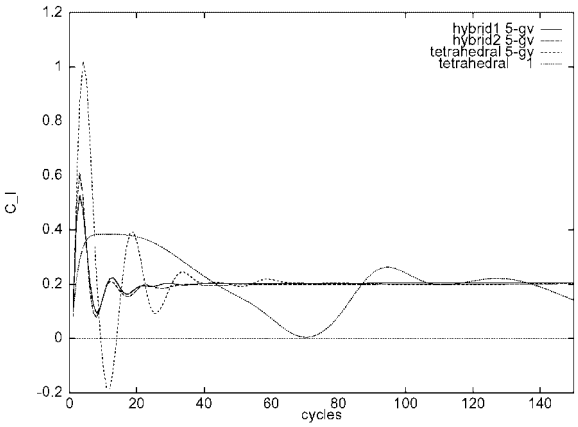

Figure 12. Plot of lift coefficient vs cycles for various mesh configurations for the ONERA M6 wing testcase.

3.4. Transient flow over oscillating B60 configuration

The concluding example is included to illustrate the time-accurate implicit solution procedure. A cyclic movement is prescribed on the wings and engines on a B60 configuration with freestream Mach number of 0.809 and an angle of attack of 2.738. The wing movement is piecewise linear, with a pitch amplitude of one degree and heave amplitude of 2% of the wing semispan at the wing mid-point, and a pitch amplitude of 5° and heave of 6.5% at the wingtip. The reduced frequency of the movement is 0.0025 and 32 timesteps were used on each cycle. Two plots of the pressure distribution of the surface at different times is given in Figure 13. The mesh used contained 775 877 tetrahedral elements, Figure 14. No remeshing was required for this example. A plot of the lift polar calculated is shown in Figure 15.

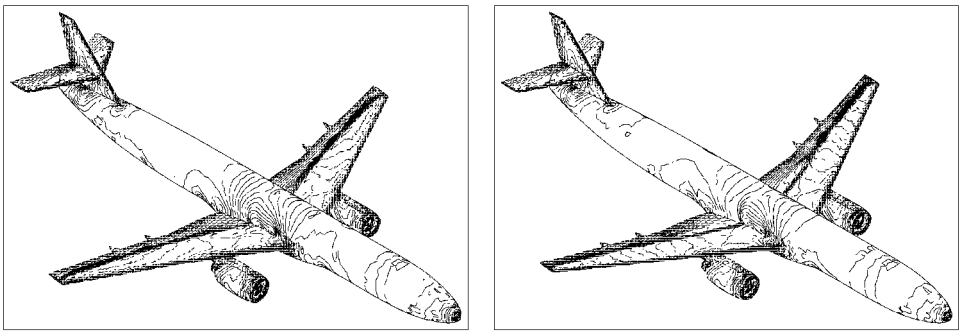

Figure 13. Snapshots of the surface pressure distributions of the transient B60 calculation.

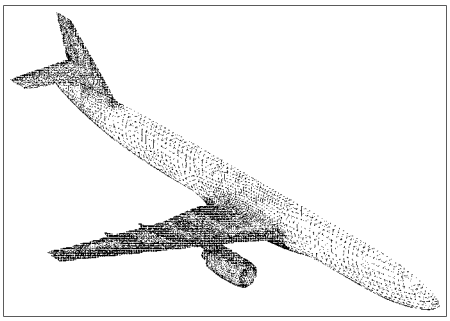

Figure 14. Surface mesh used in transient B60 calculation.

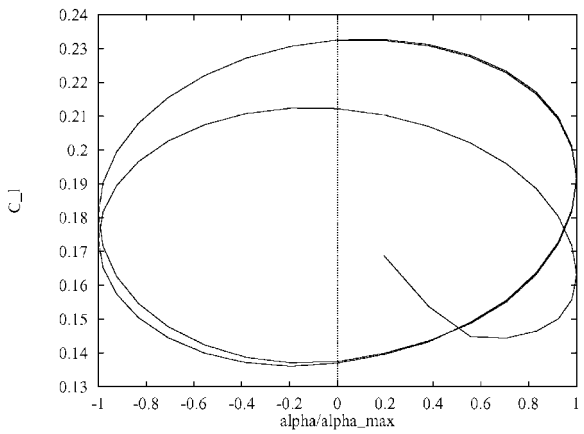

Figure 15. Lift vs normalized deflection angle for transient B60 calculation.

For this reduced frequency, the calculation time required for an explicit timestepping scheme would be a factor of 12 times greater than for the current approach.

4. CONCLUSIONS

An overview of the application of the agglomerated multigrid method to compressible flow simulations has been presented. Grid agglomeration has simplified the application of multigrid significantly, allowing the routine use of this very efficient solution acceleration scheme. The computational time is typically reduced by around one order of magnitude compared to explicit solvers, with only about a 30% increase in the memory usage. Some of the advantages related to convergence speedup with hybrid methods have been touched upon. Future publications will elaborate more on this issue, as well as focusing on the significant improvements in solution quality that result from the application of such meshes.

ACKNOWLEDGEMENTS

The project was partly funded by The Research Council of Norway, project number 125676/410.

REFERENCES

1. Mavriplis DJ. On Convergence acceleration techniques for unstructured meshes. *ICASE Report No. 98-44*, 1998.
2. Crumpton PI, Moinier P, Giles MB. An unstructured algorithm for high Reynolds number flows on highly-stretched grids. *Proceedings of the 10th International Conference on Numerical Methods for Laminar and Turbulent Flow*, 1997.
3. Lallemand MH, Steve H, Dervieux A. Unstructured multigridding by volume agglomeration: current status. *Computers & Fluids* 1992; **21**:397–433.
4. Weatherill NP, Hassan O. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering* 1994; **37**:2005–2039.
5. Nkonga B, Guillard H. Godunov type method on non-structured meshes for three-dimensional moving boundary problems. *Computational Methods in Applied Mechanical Engineering* 1994; **113**:183–204.
6. Spalart PR, Allmaras SR. A one-equation turbulence model for aerodynamic flows. *AIAA paper 92-0439*, 1992.
7. Sørensen KA, Hassan O, Morgan K, Weatherill NP. An agglomerated multigrid method on unstructured hybrid grids for turbulent compressible flow. *Proceedings of ECCOMAS 2000, Barcelona*, 2000.
8. Morgan K, Peraire J. Unstructured grid finite element methods for fluid mechanics. *Reports on Progress in Physics* 1998; **61**:569–638.
9. Jameson A, Schmidt W, Turkel E. Numerical simulation of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes. *AIAA Paper 81-1259*, 1981.
10. Brandt A. Multi-level adaptive solutions to boundary value problems. *Mathematics of Computation* 1977; **21**:333–390.
11. Probert EJ, Hassan O, Peraire J, Morgan K. An adaptive finite element method for transient compressible flows with moving boundaries. *International Journal for Numerical Methods in Engineering* 1991; **32**:1145–1159.
12. Frink NT. Recent progress toward a three-dimensional unstructured Navier–Stokes flow solver. *AIAA paper 94-0061*, 1994.

