INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 23, 527–544 (1996)

A 3D AGGLOMERATION MULTIGRID SOLVER FOR THE REYNOLDS-AVERAGED NAVIER–STOKES EQUATIONS ON UNSTRUCTURED MESHES

D. J. MAVRIPLIS AND V. VENKATAKRISHNAN

Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23665, U.S.A.

SUMMARY

An agglomeration multigrid strategy is developed and implemented for the solution of three-dimensional steady viscous flows. The method enables convergence acceleration with minimal additional memory overhead and is completely automated in that it can deal with grids of arbitrary construction. The multigrid technique is validated by comparing the delivered convergence rates with those obtained by a previously developed overset-mesh multigrid approach and by demonstrating grid-independent convergence rates for aerodynamic problems on very large grids. Prospects for further increases in multigrid efficiency for high-Reynolds-number viscous flows on highly stretched meshes are discussed.

KEY WORDS: multigrid; unstructured; Navier–Stokes

1. INTRODUCTION

With unstructured mesh techniques having proved their usefulness for two- and three-dimensional steady inviscid flow solutions, the push is now on for realistic and practical three-dimensional unstructured mesh viscous flow solvers. While useful inviscid calculations can be performed in three-dimensions using several hundred thousand grid points, the situation is quite different for viscous flow cases. Judging from current viscous block-structured and overset-structured grid calculations, the accurate aerodynamic simulation of isolated aircraft components can be expected to require the use of several million grid points,[1] whereas accurate computations for entire vehicles can be expected to require tens of millions of grid points.[2] Thus explicit time stepping is clearly not feasible and the development and implementation of efficient, robust and automated algorithms are essential if unstructured mesh techniques are to be employed for such cases. The task of developing an efficient solver is further complicated by the stiffness associated with the extreme grid stretching which is generally required for resolving high-Reynolds-number flows.

Implicit solution techniques have been demonstrated for accelerating convergence to steady state of unstructured grid solvers for both two- and three-dimensional problems.[3–6] While these methods result in substantial reductions in CPU time for a given solution, they most often greatly increase the amount of required memory. In fact, simply storing the Jacobian of the discrete equations requires two to three times more memory than an explicit scheme. Matrix-free implicit methods provide a low-storage alternative,[7] but the effectiveness of such schemes is limited by the lack of good matrix-free preconditioners.

CCC 0271–2091/96/060527–18

Received 10 March 1995
Revised 14 November 1995

Multigrid techniques provide an alternative for increasing solution efficiency in terms of CPU time while incurring minimal additional memory overhead. For unstructured meshes, various multigrid strategies have been demonstrated successfully. The fully nested multigrid approach begins with a coarse unstructured mesh which is repeatedly subdivided (possibly adaptively) in order to obtain the new finer levels.[8,9] Since the coarse and fine grid cells are nested, the intergrid transfer operators are simple to evaluate. However, the fine mesh construction is constrained by the structure of the coarse mesh, which may have adverse implications on the fine grid quality and thus solution accuracy. In another approach, coarse and fine mesh levels are generated independently from one another and the interpolation patterns between these non-nested meshes are predetermined in a preprocessing operation.[10–13] A more automated variant of this approach involves generating coarse mesh levels from a given fine mesh level by removing a subset of the fine grid points and retriangulating the remaining points.[14]

Although the above methods have all been demonstrated for practical problems, there are certain drawbacks associated with each of them. The fully nested approach requires a strong coupling between the mesh generation and multigrid strategies and does not permit the use of a predetermined fine grid. The overset-grid method places an unnecessary burden on the user by requiring the generation of multiple meshes for a single solution. The automation of this procedure does not fully resolve the issue, since generation of coarse mesh levels (manual or automated) can become extremely difficult for complex geometries which exhibit features finer than the desired grid resolution.

Agglomeration multigrid methods[15–17] and algebraic multigrid methods[18] can overcome such problems. In agglomeration multigrid, coarse mesh levels are formed by fusing together or agglomerating neighbouring cells in a given fine grid to obtain a smaller number of larger and more complex polyhedral cells. The degree of these agglomerated polyhedra increases on each coarser mesh level, but they always conform exactly to the original fine grid boundaries. Algebraic multigrid methods obviate the need for considering coarse mesh levels altogether. Instead, they operate directly at the discrete equation level, employing multiple smaller sets of discrete equations to aid in the solution of the original discrete equations.

The agglomeration multigrid strategy is somewhat dual in nature in that it can be interpreted either as a geometric approach or as an algebraic approach. In the geometric interpretation the coarse level discrete equations are obtained by discretizing the original governing equations on the coarse polyhedral cells. For the Euler equations this is easily accomplished using a finite volume analysis with the control volume taken as the polyhedral element.[15,16] For the viscous terms a least-squares technique can be used to construct the required velocity gradients. In fact, this implementation of agglomeration multigrid has an exact analogue in the automated non-nested mesh approach.[14] If we consider each agglomerated cell to be identified by its seed point (i.e. the starting fine grid vertex used to initiate the agglomeration of the cells), then these seed points may be though of as common to both the fine and coarse levels, while all other fine grid vertices may be thought of as deleted by the agglomeration procedure in the construction of the next coarser level. There then exists a particular triangulation of these coarse grid points which recovers the discretization employed in the agglomeration algorithm. The essential problem with this analogy is that, in general, the triangulation may not be valid, i.e. it may contain negative area cells and may not respect the boundary of the domain.

In the algebraic interpretation of agglomeration the coarse grid equations are constructed algebraically rather than by geometric rediscretization, using techniques similar to those developed for algebraic multigrid strategies. This is the approach pursued in this work. The basic premise is that convergence acceleration techniques should not be bound by geometry-based constraints and the removal of the influence of geometry from the multigrid process should lead to a more robust strategy. Under certain conditions, for the inviscid terms, this approach can be shown to be equivalent to the rediscretization technique. However, this is not the case for the viscous terms. Furthermore, the

algebraic philosophy carries with it implications for other details of implementation, such as boundary conditions. In fact, under certain conditions the agglomeration multigrid approach and the algebraic approach can be shown to be identical.

Agglomeration multigrid was originally devised by Lallemand *et al.*[15] for vertex schemes and by Smith[16] for cell-centred schemes. Agglomeration for diffusion problems has been investigated by Koobus *et al.*[19] as well as by the present authors in a previous paper.[20] The algebraic interpretation of agglomeration multigrid has been described in Reference 17, where the technique was applied to large three-dimensional inviscid problems. In Reference 20, further comparisons between algebraic and agglomeration multigrid methods were performed and the results were used to construct a two-dimensional Navier–Stokes solver. This paper represents the extension of the results from Reference 20 to large three-dimensional problems and the comparison of this scheme with a previously developed overset-grid multigrid technique.[13]

2. SINGLE-GRID SOLVER

The fine grid discretization is identical with that described for the non-nested multigrid approach in Reference 13. The fine grid equations are obtained by discretizing the Reynolds-averaged Navier–Stokes equations using a Galerkin finite element approach. Artificial dissipation is added in the form of an undivided Laplacian and biharmonic operator in order to damp out strong oscillations in regions near shocks and to maintain stability in smooth regions of flow respectively. In order to reduce memory overheads, both the inviscid and viscous terms are assembled using an edge-based data structure. For the inviscid terms this requires the storage of the x-, y- and z-projected areas of the dual face associated with the edge (three coefficients), while for the viscous terms it requires the storage of six edge coefficients (formally nine coefficients, which are reduced to six through symmetry properties.[10,13,21] While the use of a single edge-based data structure is instrumental in lowering memory overheads, it also enables a straightforward implementation of the agglomeration multigrid algorithm using the same data structure on all grid levels. The discrete fine grid equations are integrated in time to obtain the steady state solution, using a multistage time-stepping scheme designed to rapidly damp out high-frequency errors for multigrid effectiveness. Convergence is accelerated using local time stepping, residual averaging and the agglomeration multigrid strategy. On the coarse grid levels, only a Laplacian dissipation operator is employed, enabling the use of nearest-neighbour stencils.

Turbulence closure of the Reynolds-averaged Navier–Stokes equations is achieved using the single-field-equation turbulence model of Spalart and Allmaras.[22] This equation is discretized using first-order upwinding for the convective terms and second-order Galerkin finite elements for the diffusion and source terms. The residuals are assembled using the edge-based data structure and advanced in time using a Jacobi iteration. Convergence is accelerated using the agglomeration multigrid algorithm. The turbulence equation is thus solved simultaneously with but decoupled from the flow equations.

3. MULTIGRID STRATEGY

The central idea of agglomeration multigrid is to form coarse level meshes by grouping together neighbouring fine level control volumes. For vertex-based schemes the fine grid control volumes are defined by the centroid dual mesh, as shown in Figure 1 for the two-dimensional case. The coarser agglomerated mesh levels consist of a smaller number of larger and more complex polyhedral control volumes. Originally the coarse grid equations for agglomeration multigrid were obtained by rediscretizing the governing equations on the coarse grid levels using a finite volume approach.[15–17] This procedure is straightforward for the Euler equations. However, for the viscous terms of the Navier–Stokes equations (or even for Laplace's equation), rediscretization on complex polygonal control

volumes is non-trivial. Although rediscretization is possible for such terms (using for example least-squares gradient construction), an alternative is to construct the coarse grid equations algebraically from the fine grid discrete equations. This technique, which forms the basis for algebraic multigrid methods, states that, given a restriction operator **R**, a prolongation operator **P** and the fine grid operator **A**, the sequence of operators

$$\mathbf{A}_{\text{coarse}} = \mathbf{RAP} \tag{1}$$

yields a valid coarse grid operator $\mathbf{A}_{\text{coarse}}$. This is often referred to as a Galerkin coarse grid operator construction, since it can be shown that if **A** minimizes a functional over a set of functions spanned by the fine grid, then **RAP** minimizes the same functional over the smaller set of functions spanned on the coarser level.[18]

In the case of the Euler operations, if the restriction operator **R** and the prolongation operator **P** are both taken as piecewise constant (sometimes referred to as injection), then the Galerkin coarse grid operator construction and the finite volume rediscretization coarse grid operator construction become equivalent.[17,19,20] Therefore a unifying approach for the Navier–Stokes equations consists of employing the Galerkin construction of equation (1) for both the inviscid and viscous terms. The difficulty with this approach is that when simple piecewise constant operators are employed for restriction and prolongation, the resulting coarse grid discrete equations are not consistent with the governing equations (i.e. they do not approximate the original Navier–Stokes equations in the limit of small mesh size, since the viscous part of the operator is not scaled appropriately). This inconsistency arises only for the viscous terms and is due to the simple form of the restriction and prolongation operators.

There are two possibilities for improving the accuracy of the coarse grid operator. The first is to replace equation (1) by

$$\begin{aligned} \mathbf{A}_{\text{coarse}} &= \mathbf{A}_{\text{coarse}_{\text{inviscid}}} + \mathbf{A}_{\text{coarse}_{\text{viscous}}}, \\ \mathbf{A}_{\text{coarse}} &= \mathbf{R}\,\mathbf{A}_{\text{fine}_{\text{inviscid}}}\mathbf{P} + \frac{\mathbf{R}\,\mathbf{A}_{\text{coarse}_{\text{viscous}}}\mathbf{P}}{2^n}, \end{aligned} \tag{2}$$

where $n = 0$ represents the fine grid level, with n increasing for successively coarser grids. The subscripts 'fine' and 'coarse' in the above equation refer to any two successive mesh levels in the multigrid sequence. This heuristic fix applied to the viscous terms was derived in References 19 and 20 by examining a simple one-dimensional diffusion problem. It restores the consistency of the equations

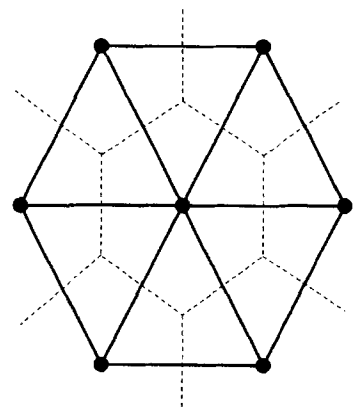

Figure 1. Dual mesh and resulting control volume at a mesh vertex

for the one-dimensional case without changing the operator stencil. The other approach is to employ more accurate forms for the restriction and prolongation operators in equation (1). In Reference 20, various forms of the prolongation, restriction and coarse grid operators were examined for a simple two-dimensional Laplace equation. While the more accurate forms resulted in increased speed of convergence of the multigrid scheme, the complexity of the operators was greatly increased, reducing the overall benefit. A desirable feature of the Galerkin construction using piecewise constant restriction and prolongation operators is that for fine grid operators based on nearest-neighbour stencils the resulting coarse grid operator also results in a nearest-neighbour stencil, thus limiting the complexity of the coarse grid operator and preserving properties of the original operator such as diagonal dominance and symmetry. Hence in this work, the form of equation (2) is employed for constructing the coarse grid equations.

As mentioned in Section 1, agglomeration multigrid can be interpreted as either a geometric or an algebraic technique. Our reliance on equations (1) and (2) for the construction of the coarse grid discrete equations indicate our preference for the algebraic interpretation. In fact, when simple piecewise constant restriction and prolongation operators are employed, the Galerkin coarse grid operator construction can be interpreted as an equation-summing technique. For each agglomerated cell the associated coarse grid equation can be obtained simply by summing the constituent fine grid equations, rescaling the viscous terms as per equation (2) and replacing the fine grid variables with the corresponding coarse grid variables. Thus the solution of the coarse grid equations reduces to solving a smaller set of summed fine grid equations. Since the fine grid equations have originally been assembled in a symmetric edge-based format, the construction of the coarse grid discrete equations is particularly simple. When a new agglomerated cell is formed, all edge coefficients associated with the edges interior to the cell cancel out owing to symmetry and those associated with the outer edges common to a neighbouring cell are simply summed, as depicted in Figure 2 for the two-dimensional case.

The coarse grid equations are thus directly inferred from the fine grid equations with no dependence on the geometry of the coarse grids. This general philosophy is applied to all details of the multigrid implementation. For example, the boundary conditions on the coarse grids are not derived from the coarse grid geometry but are inferred by the constituent fine grid equations of each coarse grid cell. This results in a natural treatment of coarse grid cells which overlap various boundary condition types. The removal of the influence of geometry from the multigrid formulation results in a more robust algorithm which is unaffected by geometry resolution (and even changes in geometry topology) on coarse grids.

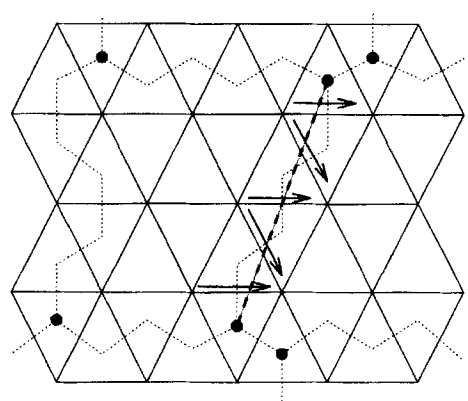

Figure 2. Combination of multiple face segments with similar neighbouring cells into a single face

4. AGGLOMERATION STRATEGY

The algorithm used to construct the coarse agglomerated levels is a graph-based technique which seeks to remove or agglomerate a subset of the fine grid points, thus resulting in a smaller set of points which constitute the coarse grid. There is a duality here between agglomeration and point removal. If each agglomerated cell is considered to be composed of its seed point (i.e. the point at which the agglomeration process was initiated) and its agglomerated points, then the seed point corresponds to a preserved coarse grid point and the agglomerated points correspond to the deleted fine grid points. The main difference is that whereas the point removal process simply results in a new set of points, with no implied connectivity, the agglomeration process also results in a new coarse grid graph, i.e. that obtained by drawing an edge between every pair of neighbouring coarse grid cells.

Our original algorithm was based on an unweighted graph where all edges were treated equally. The principle is to construct coarse grid graphs which are maximal independent sets of the previous finer grid graph. A subset of the vertices of a graph is termed an *independent set* if no vertices in the set are adjacent. An independent set is *maximal* if any vertex is not in the set is dominated by (adjacent to) at least one vertex in it. The algorithm is detailed below.

1. Pick a starting vertex on a surface element.
2. Agglomerate control volumes associated with its neighbouring vertices which are not already agglomerated.
3. Define a front as comprised of the exterior faces of the agglomerated control volumes. Place the exposed edges (duals to the exterior faces) in a queue.
4. Pick the new starting vertex as the unprocessed vertex incident to a new starting edge which is chosen from the following choices given by order of priority:
   (a) an edge on the front that is on the solid wall
   (b) an edge on the solid wall
   (c) an edge on the front that is on the far-field boundary
   (d) an edge on the far-field boundary
   (e) the first edge in the queue.
5. Go to step 2 until the control volumes for all vertices have been agglomerated.

For anisotropic problems such as high-Reynolds-number viscous flows, directional coarsening strategies are known to be beneficial.[23] Initial attempts at modifying the coarsening strategy based on the coarse grid cell aspect ratios has met with little success.[17] The present approach is to modify the algorithm detailed above based on a weighted graph. Each edge of the graph is associated with a weight and coarsening is performed preferentially in the direction of the strongest graph weights. The graph weights should be based on a quantity which reflects the strength of coupling between neighbouring discrete equations, such as the edge coefficient of the discrete equations associated with either vertex on both ends of the edge. For highly directional problems such techniques can result in semicoarsening strategies and have been shown to result in vary favourable convergence rates.[18,24] The use of weighted graphs as opposed to cell aspect ratios to guide the coarsening process is another example of the algebraic rather than geometric philosophy adopted throughout the multigrid implementation.

There are several problems which arise when applying these techniques to three-dimensional Navier–Stokes problems. The first reflects the fact that the governing equations consist of a system of equations rather than a scalar equation and thus the use of a simple discrete edge coefficient as a graph weight may not be appropriate. The second problem relates to the complexity of the coarse grids. In highly anisotropic regions a 2:1 coarsening is generally produced by the weighted graph techniques, while in the isotropic regions an 8:1 coarsening results. When the process is applied recursively, the isotropic regions of the mesh are coarsened much faster than the non-isotropic regions, thus resulting in large

diparities in neighbouring cell sizes, which can reduce the effectiveness of the coarse grid operator. A secondary effect is to increase the complexity of the coarse grids (as compared with an unweighted graph algorithm), thus increasing the cost of a multigrid cycle and obviating the possibility of using W-cycles.

The approach taken in this work is to apply the weighted graph techniques in a weak manner. The weight associated with each edge is presently based on the inverse of the edge length. This provides a crude but consistent measure of the degree of couping between neighbouring grid points. The unweighted graph algorithm described above is modified by only agglomerating neighbouring vertices for which the graph weight is greater than α times the average weight at that point. A value of $\alpha = 0$ reproduces the unweighted algorithm, while a value of $\alpha = 1$ corresponds to the fully weighted algorithm. In this work the value of $\alpha = 0{\cdot}001$ has been used exclusively. This has the effect of modifying the original unweighted algorithm only in regions of extreme grid stretching, such as near the aerofoil surfaces. This results in more uniform coarse grids while limiting their complexity. The problems described in the previous paragraph must be overcome before the beneficial effect of directional coarsening can be fully exploited.

The agglomeration process is very efficient and runs in time linearly proportional to the number of edges in the mesh. Given a list of edges in the mesh, the programme recursively constructs the specified number of coarser levels. For example, the construction of five coarse levels for the grid employed in the last example of Section 5 (2·3 million points and 16 million edges) required 1600 CPU seconds on a single CRAY C90 processor. By comparison, the extraction of the edges from the original description of the mesh requires 1800 CPU seconds and a multigrid time step requires 215 s. Neither the edge extraction nor the agglomeration procedure make use of vectorization and their performance relative to the flow solver on a scalar machine can be expected to be much faster. The main reason for running the agglomeration procedure on the CRAY C90 is the amount of memory required. The agglomeration procedure currently requires approximately the same amount of memory as the flow solver. This can be substantially reduced in the future simply by better streamlining the code and by deferring the computation of the coarse grid edge coefficients, given the agglomeration graph, to the flow solver module.

5. RESULTS

The present algorithm was used to solve several turbulent flow test-cases of aerodynamic interest which were chosen to:

1. demonstrate the relative efficiency of the current agglomeration strategy compared to a previously developed overset-mesh multigrid strategy;
2. illustrate the flexibility of the current methodology in dealing with meshes of arbitrary construction over complex geometries;
3. demonstrate the capability of solving very large problems with acceptable overheads and minimal degradation in convergence efficiency.

5.1. Single-segment wing

The first test-case involves the transonic flow over a wing of aspect ratio 2 with no sweep or spanwise variation. The wing section (independent of span location) is an RAE 2822 aerofoil. The three-dimensional grid employed for computing the flow over this wing geometry contained 1·04 million points and 6 million tetrahedra and is displayed in Figure 3. The mesh is formed by first constructing a two-dimensional unstructured grid about an RAE 2822 aerofoil using the method described in

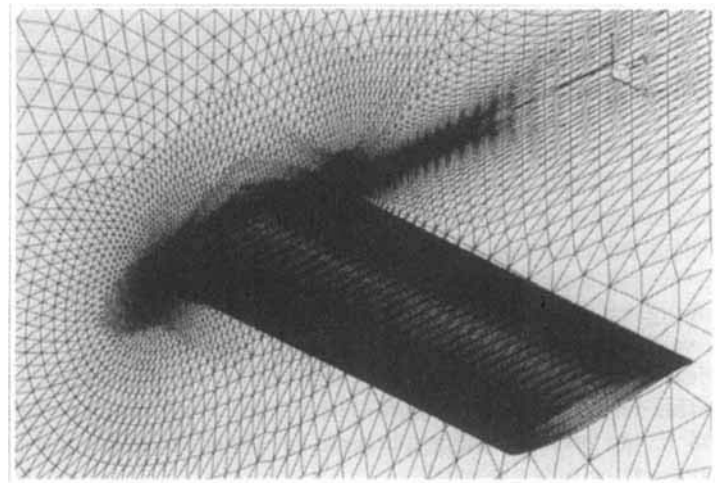

Figure 3. Unstructured grid employed for viscous flow over RAE 2822 wing (1·04 million points, 6 million tetrahedra, wall spacing 10^{-5} chords)

Reference 25. The two-dimensional mesh is then stacked in the spanwise direction, thus forming a mesh of spanwise prisms. This prismatic mesh is then converted into a tetrahedral mesh by dividing each prism into three tetrahedra. The resulting geometry consists of a wing with a symmetry plane at both ends of the wing. There is thus no wing tip present and no spanwise variation whatsoever. This can be thought of as a typical wing-in-wind-tunnel two-dimensional test. The normal mesh spacing at the wing surface is 10^{-5} chords. A total of five coarse agglomerated levels were used in the multigrid sequence.

The freestream Mach number for this case is 0·73, the incidence is 2·79° and the Reynolds number is 6·5 million. The solution is depicted qualitatively in Figure 4 as a plot of density contours on the surface of the wing and symmetry walls. The lack of any spanwise variation of the contours on the wing indicates the presence of purely two-dimensional flow. The flow is transonic and a normal shock is observed slightly aft of the mid-chord location. Figure 5 provides a more quantitative picture of this solution. The computed surface pressure at the mid-span location is compared with experimental data as well as with the computed results of the two-dimensional code on an equivalent station grid of 31,571 points. Both the two- and three-dimensional codes tend to underpredict the lift compared with the experimental data, a fact which is attributed to the turbulence model employed. For example, the same two-dimensional code achieves a lift value some 10% higher using the Baldwin–Lomax model. However, the two- and three-dimensional flow solutions agree very well with each other. The three-dimensional solution is slightly more diffusive than the two-dimensional solution, which is attributed to the presence of extra spanwise dissipation, which is non-zero even in a two-dimensional flow owing to the presence of diagonal edges in between neighbouring spanwise stations.

The convergence of the agglomeration multigrid algorithm for this case is shown in Figure 6, where it is compared with the convergence rate obtained on the same problem by the overset-mesh multigrid algorithm (developed previously in Reference 13) and with a well-validated two-dimensional multigrid code using the overset-mesh strategy[26] on the equivalent two-dimensional section grid. For this particular case the agglomeration multigrid algorithm could not be initiated with a constant initial flow field on the finest grid. Since grid sequencing using the agglomerated multigrid levels has not been implemented, a small number of single-grid cycles (10 cycles in this example) were employed to precondition the initial flow field prior to multigrid time stepping. The convergence rate of the

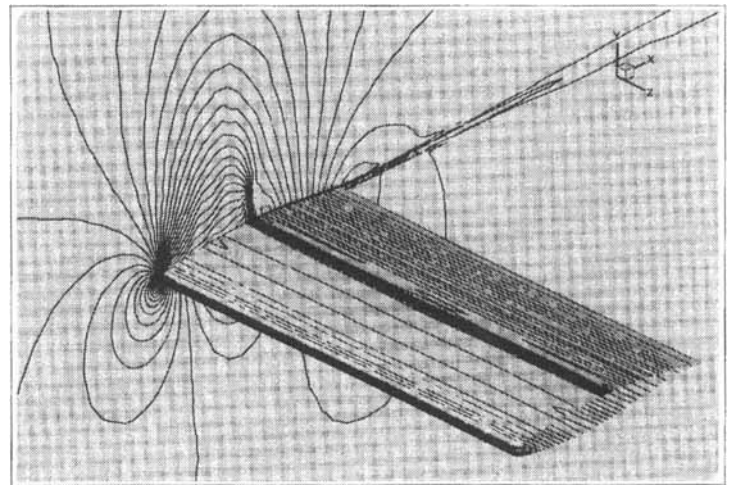

Figure 4. Computed density contours for flow over RAE 2822 wing (Mach 0·73, *Re* 6·5 million, incidence 2·79°)

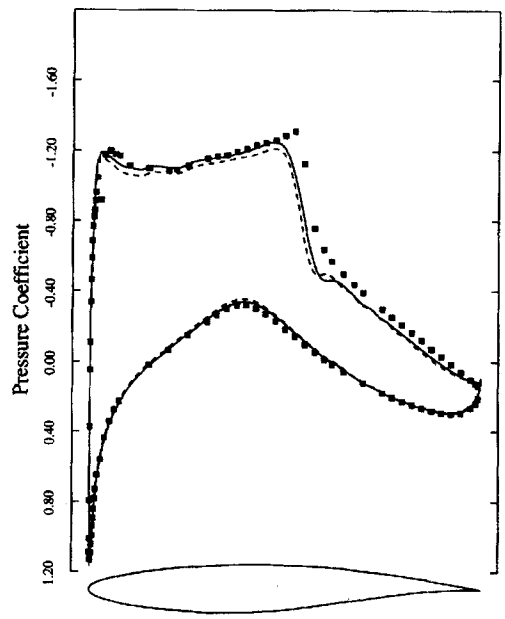

Figure 5. Comparison of computed (———, 2D; – – –, 3D) and experimental (■) surface pressure for RAE 2822 wing (Mach 0·73, *Re* 6·5 million, incidence 2·79°)

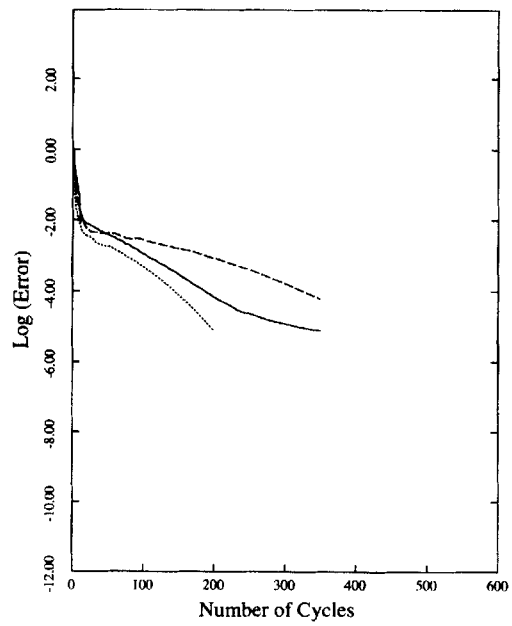

Figure 6. Agglomeration multigrid convergence rate (———) for RAE 2822 wing compared with 3D overset-mesh multigrid convergence rate (– – –) and 2D overset-mesh multigrid code run on equivalent 2D problem (· · · · · ·)

agglomeration code is seen to be faster than the equivalent overset-mesh multigrid convergence rate but slightly slower than that produced by the two-dimensional code. A slight slowdown on the asymptotic rate is observed around 300 cycles. In Figure 7 the present multigrid convergence rate is compared with the convergence rate of the same code without multigrid acceleration. The multigrid run reduces the average flow field density residuals by 5·5 orders of magnitude in 350 cycles, while the single-grid run barely achieves a reduction of two orders of magnitude in 600 cycles. This is also reflected in the values of the lift coefficient, which approaches its final value much more rapidly in the multigrid case.

The agglomeratin multigrid code required 82 s of CPU time per multigrid cycle on a single CRAY C90 processor, while the overset-mesh multigrid code required 75 s per multigrid cycle and the single-grid code 38 s per cycle. Thus the two multigrid methods are nearly equivalent in terms of overall solution efficiency and both provide an order-of-magnitude increase in efficiency over the single-grid approach. This run required a total of 185 Mwords of memory (as opposed to 177 Mwords for the overset-mesh code), which includes all required coarse grid storage and translates to approximately 180 words per fine grid vertex. This case (350 multigrid cycles) required a total of 8 h of CPU time on the CRAY C90 using a single processor.

5.2. Three-element high-lift wing

The next test-case consists of flow over a high-lift wing configuration with a slat and a single slotted flap. The wing has an aspect ratio of 2 with no sweep or spanwise variation. The wing section (independent of span location) is a Douglas three-element aerofoil which has been extensively tested both numerically and experimentally.[26] The three-dimensional fine grid employed for computing the flow over this wing geometry is displayed in Figure 8. This grid is formed by stacking a set of two-dimensional grids in the spanwise direction as described in the previous subsection. The final grid contains 1·84 million points and 10·6 million tetrahedra. The normal spacing at the wall is 10^{-6} aerofoil

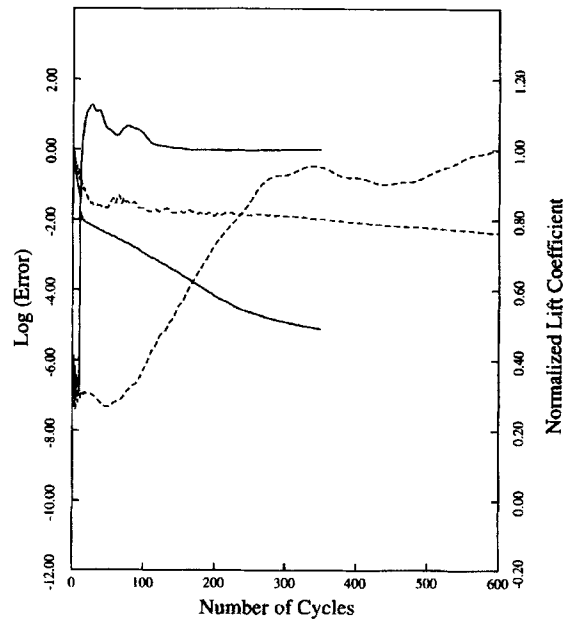

Figure 7. Comparison of agglomeration multigrid convergence rate (———) and single-grid convergence rate (– – –) for flow over RAE 2822 wing

chords for each aerofoil element. A total of six mesh levels were employed in the multigrid procedure. The freestream Mach number is 0·2, the incidence is 16·21° and the Reynolds number is 9 million for this case. The flow is assumed to be fully turbulent, so no transition points are specified. As in the previous case, the flow is entirely two-dimensional, enabling the comparison of the three-dimensional solution with a well-validated two-dimensional solver.

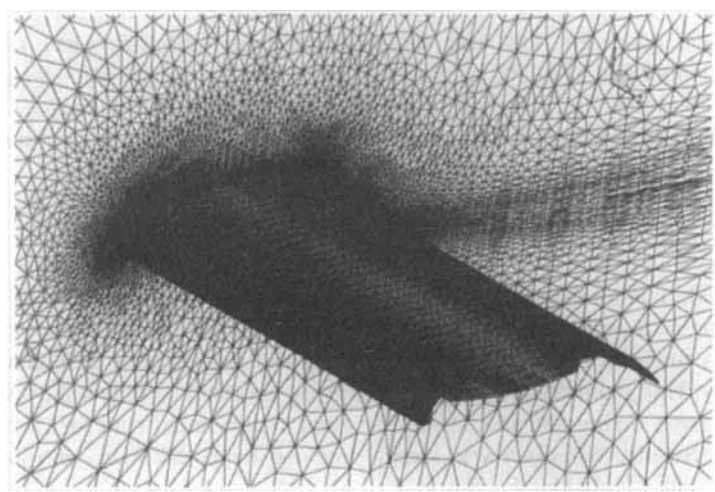

Figure 8. Fine unstructured grid for viscous flow over wing–slat–flap geometry (1·84 million points, 10·6 million tetrahedra, wall spacing 10^{-6} chords)

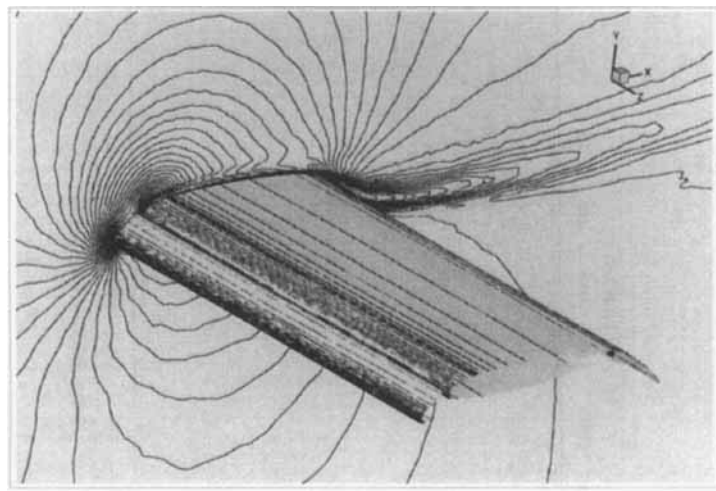

Figure 9. Computed solution for three-element wing test-case. Mach contours are displayed on the symmetry plane and density contours on the wing surface (Mach 0·2, *Re* 9 million, incidence 16·21°)

The computed Mach number contours on the wall and density contours on the aerofoil surfaces are depicted in Figure 9. A more quantitative description of the solution is given in Figure 10, where the three-dimensional computed surface pressure coefficients are compared with experimental values and with the computed values obtained using the two-dimensional code.[26] Good agreement is observed between both computations and the experimental values. The convergence rate of the agglomeration multigrid method is compared with that of the overset-mesh multigrid approach of Reference 13 and that of the two-dimensional code using the overset-mesh multigrid technique in Figure 11. The agglomeration multigrid approach is seen to converge at almost the same rate as the overset-mesh multigrid method. Both achieve identical asymptotic rates, although the non-nested multigrid approach achieves a slightly more rapid initial residual reduction. The agglomeration multigrid method achieves a residual reduction of five orders of magnitude over 400 multigrid cycles, for an average residual reduction rate of 0·973. Surprisingly, both three-dimensional codes converge slightly more rapidly than the two-dimensional code. For this case the agglomeration multigrid solver required a total of 340 Mwords of memory and 210 CPU seconds per cycle. The entire calculation was performed in four restart phases of 100 cycles. Each batch job of 100 multigrid cycles could be executed in approximately 40 min of wall clock time using all 16 processors of the CRAY C90 in a time-sharing environment during which 60% of the machine was allocated to this specific job.

5.3. Partial-span flap

The final test-case involves the solution of a truly three-dimensional flow field. The geometry consists of an unswept wing of aspect ratio 2 with a partial-span single-slotted flap which extends up to mid-span. The flap defection for this case is 30° and the gap and overlap are 0·023 chords and − 0·0039 chords respectively. This geometry is currently undergoing extensive testing in the 7X10 wind-tunnel at NASA Ames Research Center and should provide good experimental data for CFD validation in the near future. The geometry definition was kindly provided by D. Mathias, who has also performed overset-structured grid calculations for this case.[27] An unstructured grid with high stretching near the wing surfaces was generated for this configuration by S. Pirzadeh using his advancing-layers method.[28] The initial mesh contained a total of 300,000 points and 1·7 million cells. The normal spacing at the

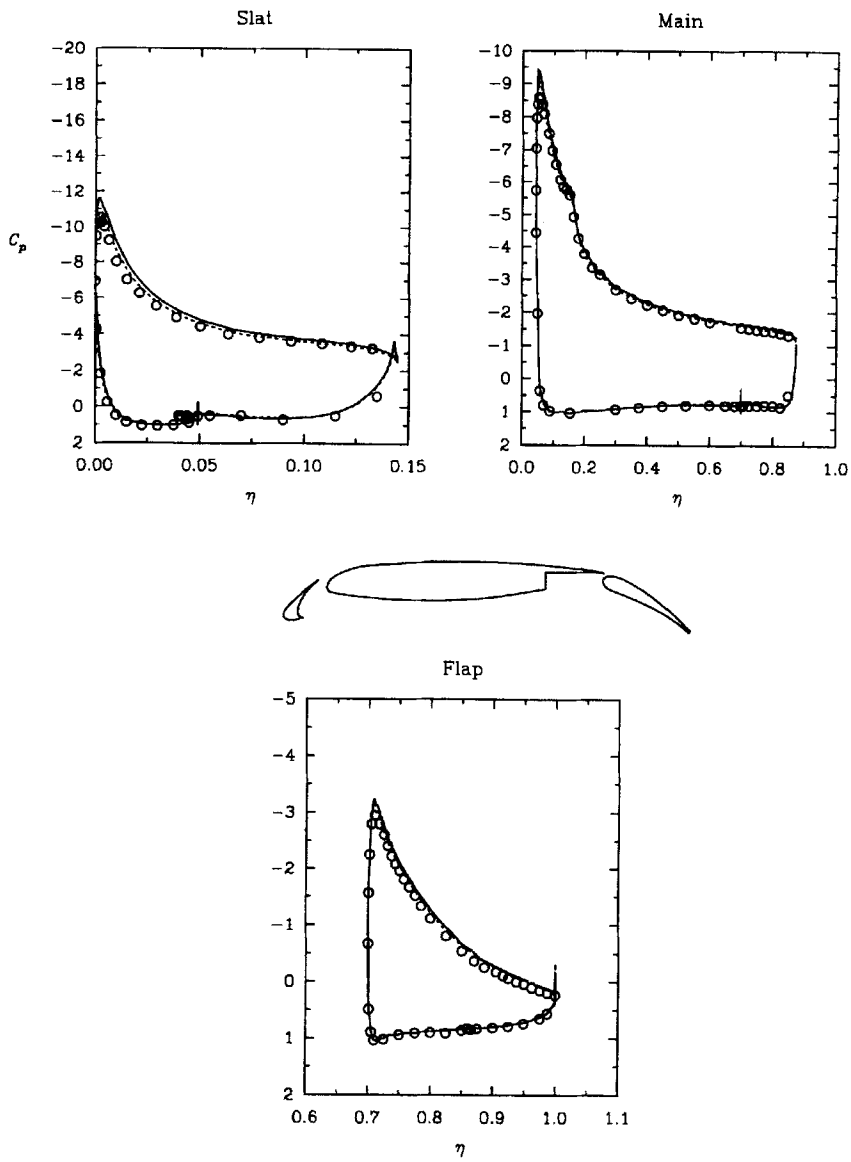

Figure 10. Comparison of computed (———) and experimental (○) surface pressure for three-element wing case (Mach 0·2, *Re* 9 million, incidence 16·21°

aerofoil surfaces is 10^{-5} chords and the wind-tunnel walls are modelled inviscidly. This mesh is depicted in Figure 12 along with the coarse agglomerated meshes employed in the multigrid procedure. While the flow was computed on this initial mesh using these agglomerated meshes, a finer mesh was also constructed and a new set of agglomerated meshes was generated for the fine grid computation. This finer mesh, which is depicted in Figure 13, contains 2·3 million points and 13·6 million cells and was derived from the initial mesh by a single pass of a simple global h-refinement technique. The

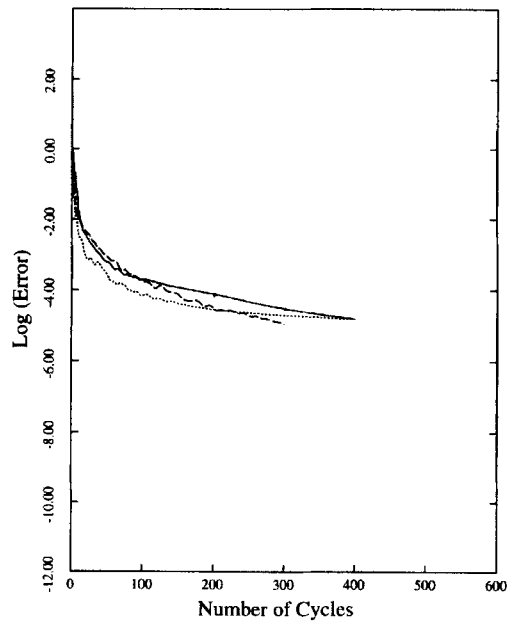

Figure 11. Comparison of agglomeration multigrid convergence rate (———) with overset-mesh multigrid convergence rate (– – –) and 2D overset-mesh multigrid code run on equivalent 2D problem (· · · · · ·)

resulting fine mesh contains a normal wall spacing of 5×10^{-6} chords over the entire wing surface. The spanwise resolution of this mesh is somewhat larger than required but is a result of current unstructured mesh strategies which favour isotropic element generation outside of the boundary layer regions.

The freestream Mach number for this cae is 0·2, the incidence is 10° and the Reynolds number is 2 million, which for this particular flap rigging corresponds to an approach condition. The fine grid solution is illustrated qualitatively in Figure 14 as a set of computed Mach contours on the wind-tunnel wall and density contours on the wing surface. At the time of writing, no experimental data were available for comparison and a full accuracy validation must be deferred to future work. Nevertheless, this case can be used to demonstrate the effectiveness of the multigrid approach in solving fully three-dimensional flows on very large grids.

The convergence rate of the agglomeration multigrid algorithm operating on the fine mesh using six mesh levels is shown in Figure 15, where it is compared with the convergence rate obtained on the coarser 300,000-point mesh using five mesh levels. The fine grid case achieved a residual reduction of 4·5 orders of magnitude over 300 multigrid cycles, which results in an average residual reduction rate of 0·967. Furthermore, the convergence rate achieved by the fine grid is almost identical with that achieved on the coarser grid. This provides a good indication that the agglomeration multigrid strategy has achieved a nearly grid-independent convergence rate for this case, especially given that the fine grid contains eight times as many points as the coarse grid. The fine grid case required a total of 425 Mwords of memory and 18 h of total CPU time for 300 multigrid cycles on a CRAY C90. This was performed in three restart runs of 100 multigrid cycles, each requiring approximately 6 h of CPU time, but less than 1 h of wall clock time using all 16 processors in a time-sharing environment where 40% of the machine was available for this specific job. An average computational rate of 1·6 Gflops was reported during these runs by the CRAY hardware performance monitor. This suggests that the entire 300 cycle run could be performed in 1·2 h at a speed of 4 Gflops in a dedicated environment on the CRAY C90 using all 16 processors.

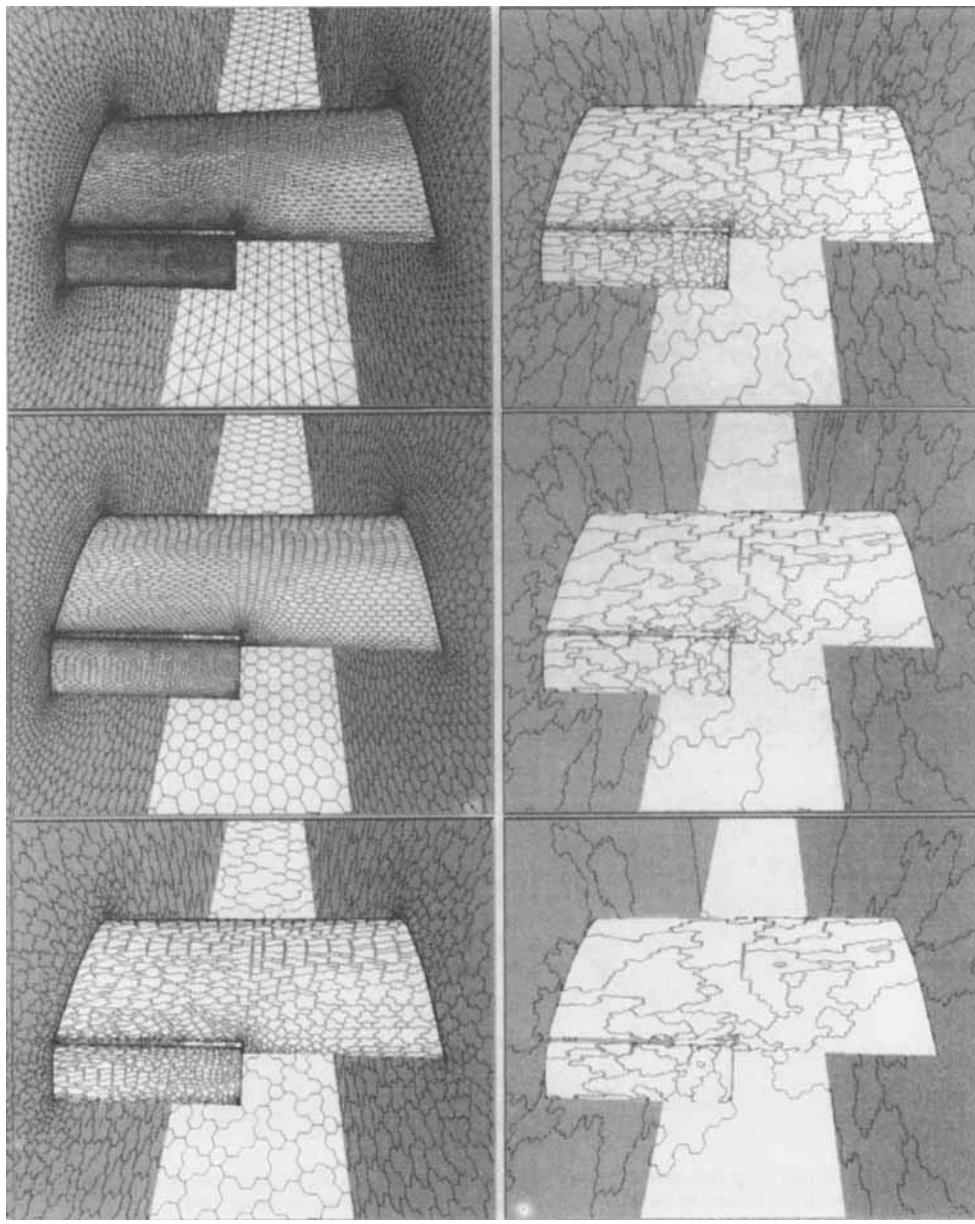

Figure 12. Initial unstructured grid, its dual mesh and four agglomerated coarse mesh levels employed for multigrid algorithm for partial-span-flap wing in wind-tunnel geometry (300,000 points, 1·7 million tetrahedra, wall spacing 10^{-5})

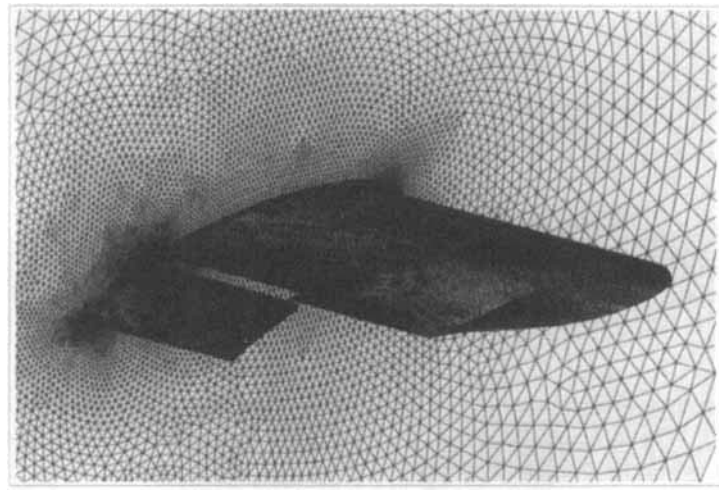

Figure 13. Fine unstructured mesh employed for viscous flow computation over partial-span-flap wing in wind-tunnel geometry (2·3 million points, 13·6 million tetrahedra, wall spacing 5×10^{-6} chords)

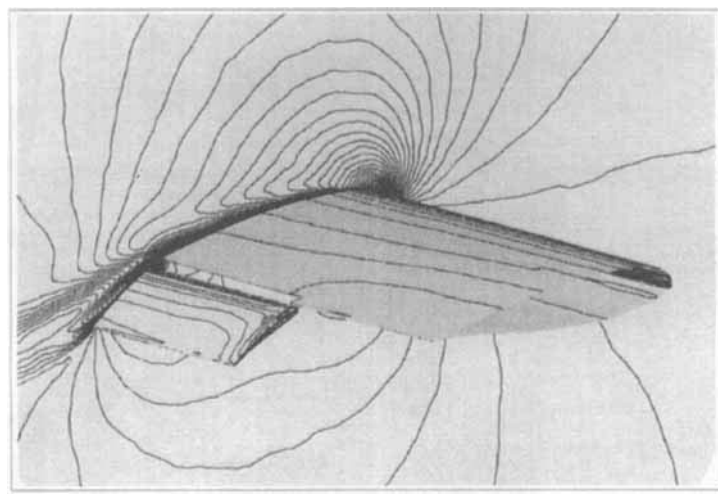

Figure 14. Computed solution for partial-span-flap wing test-case. Mach contours are displayed on the wind-tunnel wall and density contours on the wing surface (Mach 0·2, *Re* 2 million, incidence 10·0°)

6. CONCLUSIONS

The agglomeration multigrid strategy has been shown to be an effective means for accelerating convergence and reducing the CPU time required for large three-dimensional unstructured grid Navier–Stokes calculations while incurring minimal additional memory overheads. The agglomeration strategy delivers convergence rates similar to those obtained using the previously developed overset-mesh multigrid approach,[13] but it is fully automatic and can deal with meshes of arbitrary construction.

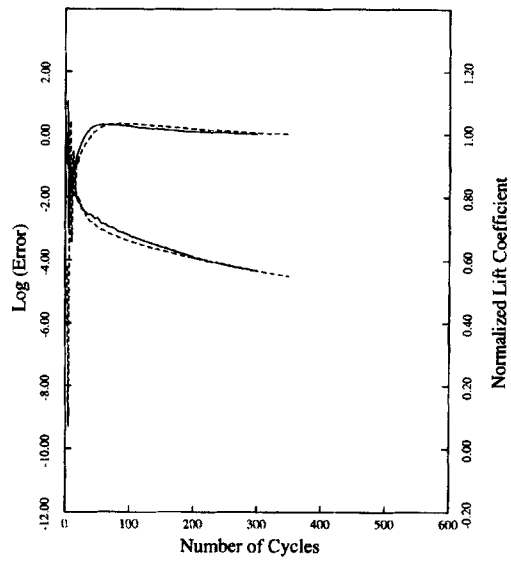

Figure 15. Agglomeration multigrid convergence rates on coarse (————, 300,000 points) and fine (- - - -, 2·4 million points) grid for flow over partial-span-flap wing in wind-tunnel

The fact that the present method is comparable with the overset-mesh multigrid method in terms of convergence rates lends support to the claim that the heuristics involved in deriving the coarse grid operator for the viscous terms are justified. Although better coarse grid operators may be devised, we believe that larger gains can be made by developing more sophisticated coarsening strategies. The degradation of multigrid convergence rates with increasing grid stretching for high-Reynolds-number viscous flows is well known. The use of weighted graph techniques to produce directional and adaptive coarsening strategies, which has only been invoked weakly in the present work, is currently under way[24] and will be investigated more thoroughly in future work.

ACKNOWLEDGEMENTS

The authors would like to thank D. Mathias for providing the partial-span flap geometry and S. Pirzadeh for generating the unstructured grid on this configuration using his advancing-layers method. This work was made possible in large part thanks to the computer time provided by the National Aerodynamic Simulation (NAS) facility.

This work was supported by NASA contract NAS1-19480.

REFERENCES

1. V. Vatsa and B. W. Wedan, 'Development of a multigrid code for 3D Navier–Stokes equations and its application to a grid refinement study', *Comput. Fluids*, **18**, 391–403 (1990).
2. R. J. Gomez and E. C. Ma, 'Validation of a large scale chimera grid system for the Space Shuttle launch vehicle', *AIAA Paper 94-1859*, 1994.
3. V. Venkatakrishnan and D. Mavriplis, 'Implicit solvers for unstructured meshes', *J. Comput. Phys.*, **105**, 83–91 (1993).
4. W. K. Anderson and D. L. Bonhaus, 'An implicit upwind algorithm for computing turbulent flows on unstructured grids', *Comput. Fluids*, **23**, 1–24 (1994).
5. L. Fezoui and B. Stoufflet, 'A class of implicit upwind schemes for Euler simulations with unstructured meshes', *J. Comput. Phys.*, **84**, 174–206 (1989).

6. D. L. Whitaker, 'Three-dimensional unstructured grid Euler computations using a fully implicit upwind method', *AIAA Paper 93-3337*, 1993.
7. Z. Johan, T. J. R. Hughes, K. K. Mathur and S. L. Johnsson, 'A data parallel finite element method for computational fluid dynamics on the Connection Machine system', *Comput. Methods Appl. Mech. Eng.*, **99**, 113–124 (1992).
8. S. D. Connell and D. G. Holmes, 'A 3D unstructured adaptive multigrid scheme for the Euler equations', *AIAA Paper 93-3339*, 1993.
9. V. Parthasarathy and Y. Kallinderis, 'New multigrid approach for three-dimensional unstructured adaptive grids', *AIAA J.*, **32**, 956–963 (1994).
10. D. J. Mavriplis, 'Three dimensional unstructured multigrid for the Euler equations', *AIAA J.*, **30**, 1753–1761 (1992).
11. M. P. Leclercq, 'Resolution des equations d'Euler par des methodes multigrilles conditions aux limites en regime hypersonique', *Ph.D. Thesis*, Applied Math, Université de Saint-Etienne, 1990.
12. J. Peraire, J. Peiro and K. Morgan, 'A 3D finite-element multigrid solver for the Euler equations', *AIAA Paper 92-0449*, 1992.
13. D. J. Mavriplis, 'A three dimensional multigrid Reynolds averaged Navier–Stokes solver for unstructured meshes', *AIAA Paper 94-1878*, 1994.
14. H. Guillard, 'Node nested multigrid with Delaunay coarsening', *INRIA Rep. 1898*, 1993.
15. M. H. Lallemand, H. Steve and A. Dervieux, 'Unstructured multigridding by volume agglomeration: current status', *Comput. Fluids*, **21**, 397–433 (1992).
16. W. A. Smith, 'Multigrid solution of transonic flow on unstructured grids', in O. Baysal (ed.), *Recent Advances and Applications in Computational Fluid Dynamics*, pp. 140–147, ASME, New York, 1990.
17. V. Venkatakrishnan and D. Mavriplis. 'Agglomeration multigrid for the 3D Euler equations', *AIAA J.*, **23**, 633–640 (1995).
18. J. W. Ruge and K. Stuben, 'Algebraic multigrid', in S. F. McCormick (ed.), *Multigrid Methods*, SIAM, Philadelphia, PA, 1987, pp. 73–131.
19. B. Koobus, M. H. Lallemand and A. Dervieux, 'Unstructured volume–agglomeration MG: solution of the Poisson equation, *INRIA Rep. 1946*, 1993.
20. D. J. Mavriplis and V. Venkatakrishnan, 'Agglomeration multigrid for two-dimensional viscous flows', *Computers Fluids*, **24**, 553–570 (1995).
21. T. J. Barth 'Numerical aspects of computing viscous high-Reynolds number flows on unstructured meshes', *AIAA Paper 91-0721*, 1991.
22. P. R. Spalart and S. R. Allmaras, 'A one-equation turbulence model for aerodynamic flows', *AIAA Paper 92-0439*, 1992.
23. W. A. Mulder, 'A new multigrid approach to convection problems', *J. Comput. Phys.*, **83**, 303–323 (1989).
24. E. Morano, D. J. Mavriplis and V. Venkatakrishnan, 'Coarsening strategies for unstructured multigrid techniques with application to anisotropic problems', *Proc. Copper Mountain Conf. on Multigrid Methods*, April 1995, eds. N. D. Melson, T. A. Manteuffel, S. F. McCormick and C. C. Douglas, NASA CP, 1996 (to appear).
25. D. J. Mavriplis, 'Unstructured and adaptive mesh generation for high-Reynolds number viscous flows', in A. S. Arcilla, J. Hauser, P. R. Eiseman and J. F. Thompson (eds), *Proc. Third Int. Conf. on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, pp. 79–92, North-Holland, Amsterdam, 1991.
26. W. O. Valarezo and D. J. Mavriplis, 'Navier–Stokes applications to high-lift airfoil analysis', *AIAA Paper 93-3534*, 1993.
27. D. L. Mathias, K. R. Roth, J. C. Ross, S. E. Rogers and R. M. Cummings, 'Navier–Stokes analysis of the flow about a flap edge', *AIAA Paper 95-0185*, 1995.
28. S. Pirzadeh, 'Viscous unstructured three-dimensional grids by the advancing-layers methods', *AIAA Paper 94-0417*, 1994.